



Peter Loos, Markus Nüttgens,
Klaus Turowski, Dirk Werth (Hrsg.)

MobIS 2008

Modellierung betrieblicher Informationssysteme

**– Modellierung zwischen SOA und
Compliance Management –**

**27.–28. November 2008
Saarbrücken, Germany**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-141

ISBN 978-3-88579-235-2

ISSN 1617-5468

Volume Editors

Prof. Dr. Peter Loos

Institut für Wirtschaftsinformatik (IWi) im DFKI
Universität des Saarlandes
Campus D 3 2, 66123 Saarbrücken, Germany
Email: peter.loos@iwi.dfki.de

Prof. Dr. Markus Nütgens

Universität Hamburg
WISO Fakultät, Wirtschaftsinformatik
Von-Melle-Park 9, 20146 Hamburg, Germany
Email: markus.nuetgens@wiso.uni-hamburg.de

Prof. Dr. Klaus Turowski

Universität Augsburg
Wirtschaftsinformatik und Systems Engineering
Universitätsstr. 16, 86159 Augsburg, Germany
Email: klaus.turowski@wiwi.uni-augsburg.de

Dr. Dirk Werth

Institut für Wirtschaftsinformatik (IWi) im DFKI
Universität des Saarlandes
Campus D 3 2, 66123 Saarbrücken, Germany
Email: dirk.werth@iwi.dfki.de

Series Editorial Board

Heinrich C. Mayr, Universität Klagenfurt, Austria (Chairman, mayr@ifit.uni-klu.ac.at)

Jörg Becker, Universität Münster, Germany

Hinrich Bonin, Leuphana-Universität Lüneburg, Germany

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, SAP Research, Germany

Johann-Christoph Freytag, Humboldt-Universität Berlin, Germany

Ulrich Furbach, Universität Koblenz, Germany

Michael Koch, Universität der Bundeswehr, München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Liggesmeyer, TU Kaiserslautern und Fraunhofer IESE, Germany

Ernst W. Mayr, Technische Universität München, Germany

Heinrich Müller, Universität Dortmund, Germany

Sigrid Schubert, Universität Siegen, Germany

Martin Warnke, Leuphana-Universität Lüneburg, Germany

Dissertations

Dorothea Wagner, Universität Karlsruhe, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics

Andreas Oberweis, Universität Karlsruhe (TH)

© Gesellschaft für Informatik, Bonn 2008

printed by Köllen Druck+Verlag GmbH, Bonn

Veranstalter

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)
Universität des Saarlandes
Campus D32, Stuhlsatzenhausweg 3, D-66123 Saarbrücken

Prof. Dr. Peter Loos (Tagungsleitung und Vorsitzender des Programmkomitees)
Dipl.-Kfm., Dipl.-Inform. Thorsten Dollmann (Organisation)

Die Tagung wurde von der Gesellschaft für Informatik, Fachgruppe Informationssystem-Architekturen: Modellierung betrieblicher Informationssysteme (WI-MobIS), gemeinsam mit den Arbeitskreisen Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK) und Komponentenorientierte betriebliche Anwendungssysteme (WI-KobAS) sowie dem Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) organisiert.

Programmkomitee

Prof. Dr. Thomas Allweyer, FH Kaiserslautern
Prof. Dr. Jan vom Brocke, Hochschule Liechtenstein
Prof. Dr. Jörg Becker, Universität Münster
Prof. Dr. Jörg Desel, KU Eichstätt
Prof. Dr. Werner Esswein, TU Dresden
Dr. Peter Fettke, Universität des Saarlandes
Prof. Dr. Ulrich Frank, Universität Duisburg-Essen
Prof. Dr. Norbert Gronau, Universität Potsdam
Prof. Dr. Andreas Gadatsch, Fachhochschule Bonn-Rhein-Sieg
Prof. Dr. Martin Gaedke, Technische Universität Chemnitz
Dr. Helge Hess, IDS Scheer AG
Prof. Dr. Axel Hahn, Universität Oldenburg
Prof. Dr. Ekkart Kindler, DTU Copenhagen
Dr. Christine Legner, European Business School, Oestrich-Winkel
Prof. Dr. Peter Loos, Universität des Saarlandes (Tagungsleiter und Vorsitz MobIS2008)
Dr. Jan Mendling, Queensland University of Technology, Brisbane,
Prof. Dr. Markus Nüttgens, Universität Hamburg (Vorsitz WI-EPK-Track)
Prof. Dr. Volker Nissen, TU Ilmenau
Prof. Dr. Andreas Oberweis, Universität Karlsruhe (TH)
Prof. Dr. Frank Rump, FH Oldenburg/Ostfriesland/Wilhelmshaven
Prof. Dr. Michael Rebstock, Fachhochschule Darmstadt
Prof. Dr. Michael Rosemann, Queensland University of Technology, Brisbane
Prof. Dr. Peter Rittgen, Uni Boras
Prof. Dr. Carlo Simon, Provadis Hochschule Frankfurt
Prof. Dr. Elmar J. Sinz, Universität Bamberg
Dr. Oliver Thomas, Universität des Saarlandes
Prof. Dr. Klaus Turowski, Universität Augsburg (Vorsitz WI-KobAS-Track)
Dr. Dirk Werth, Universität des Saarlandes (Vorsitz ModCollGP-Track)
Prof. Dr. Mathias Weske, Universität Potsdam
Prof. Dr. Robert Winter, Universität St. Gallen

Vorwort

Die in diesem Sammelband zusammengefassten Beiträge wurden anlässlich der Tagung „MobIS2008 – Modellierung betrieblicher Informationssysteme“ am 27. und 28. November 2008 verfasst und im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) in Saarbrücken präsentiert. Zur Tagung wurden 48 Beiträge eingereicht. Insgesamt wurden für dieses Buch 19 Beiträge angenommen, was zu einer Annahmequote von knapp 39 % führte.

Die Ordnung der Beiträge in diesem Band richtet sich nach den Themenschwerpunkten der Haupttagung MobIS2008 sowie der beiden durch die beteiligten GI-Arbeitskreise ausgerichteten Tracks. Im ersten Teil des Buches steht mit *Modellierung zwischen SOA und Compliance Management* das Leithema der Haupttagung im Vordergrund. Der zweite Teil des Buches beinhaltet eine Selektion der Beiträge des Tracks *Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*. Der dritte Teil dieses Konferenzbandes präsentiert ausgewählte Beiträge des Tracks *Komponentenorientierte betriebliche Anwendungssysteme*.

Die Herausgeber danken an dieser Stelle den Mitgliedern des Programmkomitees, die alle eingegangenen Beiträge in kürzester Zeit begutachteten. Dank schulden wir auch Prof. Dr. Heinrich C. Mayr als Herausgeber der LNI-Reihe für die Aufnahme des Tagungsbandes in diese Reihe, Cornelia Winter seitens der Geschäftsstelle der Gesellschaft für Informatik sowie dem Team vom Köllen-Verlag für die professionelle Zusammenarbeit.

Unser herzlicher Dank gebührt schließlich insbesondere den Personen, die an der Organisation der Tagung und an der Erstellung dieses Buches unmittelbar mitgewirkt haben. Zu erwähnen sind hier alle beteiligten studentischen und wissenschaftlichen Mitarbeiter am Institut für Wirtschaftsinformatik (IWi) im DFKI. Hervorzuheben ist dabei Herr Thorsten Dollmann, dessen unermüdlicher Einsatz diese Konferenz überhaupt erst möglich gemacht hat.

Die Herausgeber

Saarbrücken/Hamburg/Augsburg, im Oktober 2008

Inhaltsverzeichnis

Vorwort.....	1
Modellierung zwischen SOA und Compliance Management	
Compliance-Prüfung bei Anwendung dynamischer Bindung in serviceorientierten Architekturen.....	7
<i>Gabriela Loosli</i>	
Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle	23
<i>Patrick Delfmann, Sebastian Herwig, Łukasz Lis, Armin Stein</i>	
Model-based Competency-oriented Business Process Analysis	39
<i>Katrina Leyking, Ralf Angeli</i>	
On the Choice Between Graph-Based and Block-Structured Business Process Modeling Languages.....	59
<i>Oliver Kopp, Daniel Martin, Daniel Wutke, Frank Leymann</i>	
3D Representation of Business Process Models	73
<i>Stefanie Betz, Eichhorn Daniel, Susan Hickl, Stefan Klink, Agnes Koschmider, Yu Li, Andreas Oberweis, Ralf Trunko</i>	
Designing and Utilising Business Indicator Systems within Enterprise Models – Outline of a Method	89
<i>Ulrich Frank, David Heise, Heiko Kattenstroth, Hanno Schauer</i>	
Business Process Compliance Checking: Current State and Future Challenges.....	107
<i>Marwane El Kharbili, Ana Karla Alves de Medeiros, Sebastian Stein, Wil M. P. van der Aalst</i>	
An Engineering Approach to Enterprise Architecture Design and its Application at a Financial Service Provider.....	115
<i>Stephan Aier, Stephan Kurpjuweit, Otto Schmitz, Jörg Schulz, André Thomas, Robert Winter</i>	
Towards simulation-supported enterprise architecture management.....	131
<i>Sabine Buckl, Florian Matthes, Wolfgang Renz, Christian M. Schweda, Jan Sudeikat</i>	

Prozessmusterbezogene Kommunikationsmaßnahmen im Requirements
Engineering – Eine Fallstudie 147
Markus Schäfermeyer, Marianne Corvera

Using Fuzzy Process Models to Improve Technical Customer Services:
A Case Study for Heating Facilities 165
Oliver Thomas, Philipp Walter, Thorsten Dollmann, Peter Loos, Michael Schlicker

Geschäftsprozessmanagement mit EPK

EPK-Validierung zur Modellierungszeit in der bflow* Toolbox 181
Volker Gruhn, Ralf Laue, Heiko Kern, Stefan Kühne

Reducing Complexity of Large EPCs 195
Artem Polyvyanyy, Sergey Smirnov, Mathias Weske

Ein Transformationsmodell zur Überführung von Prozessmodellen
in eine Simulationsumgebung 209
Mathias Petsch, Hagen Schorcht, Volker Nissen, Katja Himmelreich

Kennzahlenbasierte Analyse von Geschäftsprozessmodellen als Beitrag
zur Identifikation von SOA Services 221
Werner Esswein, Jens Weller, Jeannette Stark, Martin Juhrisch

Integrierte Produkt- und Prozessmodellierung: Rahmenkonzept und
Anwendungsfall zur EU-Dienstleistungsrichtlinie 239
Frank Hogreve and Markus Nüttgens

Komponentenorientierte betriebliche Anwendungssysteme

Zur systematischen Identifikation von Services: Kriterien, aktuelle Ansätze
und Klassifikation 255
Dominik Birkmeier, Sebastian Klöckner, Sven Overhage

Components for Growing the RESTful Enterprise 273
Andreas Heil, Johannes Meinecke, Martin Gaedke

Text Driven Modeling Architecture – Modellierung im Zeitalter
Serviceorientierter Architekturen 285
André Mai

Modellierung zwischen SOA und Compliance Management

Compliance-Prüfung bei Anwendung dynamischer Bindung in serviceorientierten Architekturen

Gabriela Loosli

Institut für Wirtschaftsinformatik, Abteilung Information Engineering
Universität Bern
Engehaldenstrasse 8
CH-3012 Bern
gabriela.loosli@iwi.unibe.ch

Abstract: Eine wesentliche Eigenschaft der serviceorientierten Architektur (SOA) ist die Bindung zur Laufzeit. Diese dynamische Bindung von Services kann verschiedene Ausprägungsformen aufweisen. Bei Anwendung der weitreichendsten Form der dynamischen Bindung besteht die Gefahr, dass aufgrund von Änderungen des Service-Bestands im Repository zur Laufzeit automatisch nicht konforme Services gewählt werden. Nicht konform bedeutet in diesem Zusammenhang, dass nicht alle erforderlichen Regulierungen erkannt und demnach nicht alle erfüllt werden. Bei Wiederverwendung von Services in einem anderen Kontext müssen u.U. zusätzliche Regulierungen erfüllt werden. Bisherige Change-Management-Ansätze, welche auf der Überprüfung aller Services und Applikationen vor Produktivsetzung basieren, lösen dieses Problem nicht. Demnach kann Compliance nicht immer gewährleistet werden. Dieser Beitrag stellt einen Ansatz vor, der mit Hilfe von semantischen Konzepten die Auswahl nicht konformer Services verhindern soll.

1 Einleitung

Ein wichtiges Ziel einer serviceorientierten Architektur (SOA) ist die Unterstützung der Agilität eines Unternehmens im Sinne der Flexibilität, Änderungen in den Geschäftsprozessen zeitnah in IT-Systeme umsetzen zu können [Er04, 297; KBS04, 1ff; Ab07, 1; JG07, 189; WS07, 44ff]. Eine SOA ist eine Software-Architektur, in der Services die fundamentalen Elemente darstellen. Services sind Software-Einheiten verschiedener Granularität, welche zu komplexeren Services bis hin zu Prozessen bzw. Applikationen aggregiert werden können (vgl. z.B. [PG03; Er04, 33f; KBS04, 58ff; Do05, 7]). Die Flexibilität wird erreicht durch lose Kopplung der Services. Indem eine Beziehung zwischen ihnen erst im Prozessablauf hergestellt wird, können sie innerhalb eines Prozesses besser ausgetauscht oder erst bei Bedarf eingebunden werden. Im SOA-Konzept ist dies sogar zur Laufzeit möglich, da die Services nicht im Quellcode fest verlinkt, sondern über ihre Angaben im Repository gebunden werden. Die Bindung zur Laufzeit wird auch als dynamische Bindung bezeichnet und trägt wesentlich zur losen Kopplung bei [Er04, 297; KBS04, 46ff; Do05, 9; JG07, 190f].

Neben diesem Vorteil ist jedoch zu beachten, dass bei einer dynamischen Auswahl von Services ein besonderes Augenmerk auf die rechtlichen Aspekte gelegt werden muss. Beispielsweise bedeutet eine "zufällige Einbindung eines Dienstes" eine "nicht gesteuerte Veränderung des Gesamtsystems. Somit wäre das System in einem undefinierten Zustand und damit nicht mehr betriebsbereit" [Do05, 261].

Obwohl auch bei statischer Bindung noch ungelöste Probleme in Bezug auf Compliance vorhanden sind (u.a. da Prozesse aus Services bestehen, welche aus unterschiedlichen Organisation(seinheit)en stammen; vgl. z.B. [Pa06, 24; JG07, 191]), konzentriert sich dieser Beitrag auf die dynamische Bindung. Es wird gezeigt, welche Auswirkungen die verschiedenen Ausprägungsformen der dynamischen Bindung von Services auf die Compliance zu Vorschriften haben und ein Ansatz vorgestellt, wie mithilfe von semantischen Konzepten der Kontext der Servicenutzung berücksichtigt und daraus folgend die anzuwendenden Regulierungen bestimmt werden können. Damit soll die Auswahl nicht konformer Services verhindert werden.

Der Beitrag ist wie folgt gegliedert: In Abschnitt 2 wird auf verwandte Arbeiten sowie die Abgrenzung zu ihnen eingegangen. Abschnitt 3 definiert den Begriff Compliance. Danach werden in Abschnitt 4 anhand einzelner Beispiele die Problematik der Erfüllung der Compliance in einer SOA bei Anwendung von verschiedenen Ausprägungsformen der dynamischen Bindung erläutert und diesbezügliche Lösungsansätze vorgestellt. Abschnitt 5 gibt eine Zusammenfassung und einen Ausblick.

2 Verwandte Arbeiten

Obwohl Compliance ein sehr aktuelles Thema ist, wird es in der SOA-Literatur wenig beachtet. [Do05, 252ff] widmen den rechtlichen Rahmenbedingungen ein eigenes Teilkapitel und erwähnen auch die Problematik der dynamischen Bindung diesbezüglich. Jedoch geben sie an, das Thema nicht erschöpfend zu behandeln. So sollen zwar "alle zur Auswahl stehenden Dienste den rechtlichen Anforderungen genügen und ein Wechsel der Dienste im Voraus einkalkuliert und dokumentiert" werden [Do05, 261]; die Prüfung der Prozesse und die Wiederverwendung in einem anderen Kontext werden ebenso nicht betrachtet wie die verschiedenen Formen der dynamischen Bindung, diesen wird in der SOA-Literatur allgemein wenig Beachtung geschenkt.

In neueren Veröffentlichungen wird der Begriff *SOA-Governance* erwähnt (vgl. z.B. [Ab07; JG07; Ke07; Si07; SS07; Jo08]). Jedoch wird dieser Begriff oft umfassender verwendet als nur im Hinblick auf Compliance-Nachweise und umfasst auch Aspekte des klassischen IT-Managements (zur Abgrenzung der Begriffe vgl. [KL06, 451, 454]). Die dynamische Bindung wird in diesen Beiträgen nicht berücksichtigt. [JG07, 191] erwähnen die Wiederverwendung in anderen Kontexten, damit möglicherweise verbundene Probleme werden jedoch nicht betrachtet.

Im Gegensatz dazu berücksichtigt [Fo06] die zusätzlich zu erfüllenden Regulierungen und demnach die Auswirkungen der Wiederverwendung von *bestehenden Services* in *neuen Prozessen*. Was fehlt, ist die umgekehrte Betrachtungsweise, die Auswirkungen von *neu* im Repository verfügbaren *Services* auf *bestehende Prozesse* untersucht (vgl. Abschnitt 4.1).

Für die Erbringung der Compliance-Nachweise existieren interessante *Lösungsansätze*, welche die Prozesse, teilweise sogar zur Laufzeit, überprüfen. Jedoch weisen alle Unvollständigkeiten auf: Einige sind nicht spezifisch auf eine SOA ausgerichtet [Gi05; Ag06; NS07]. Andere prüfen Prozessänderungen zur Laufzeit nicht [LMX07; NS07]. [Ag06] konzentrieren sich auf eine einzige Regulierung. In [Og04] werden generische, allen Regulierungen zugrunde liegende, Compliance-Services (z.B. Zugriffskontrollen) vorgeschlagen. Jedoch ist dabei zu beachten, dass gerade die teilweise unterschiedlichen Details der verschiedenen Regulierungen für ihre Erfüllung von Bedeutung sind. Eine Übersicht über Arbeiten in den Bereichen semantische Konzepte und dynamische Bindung gibt [Ku08]. Die Arbeiten beziehen sich u.a. auf fehlerhafte und nicht passende Dienste, jedoch nicht spezifisch auf Compliance zu Regulierungen.

3 Compliance

Es ist nicht neu, dass Unternehmen Vorschriften erfüllen müssen. Neu ist die starke Zunahme, die weit reichenden Anforderungen sowie die Internationalität der Vorschriften (vgl. z.B. [Do05, 252f; Kn07, S98f]). Das meist erwähnte Beispiel in diesem Zusammenhang ist der Sarbanes-Oxley Act (SOX). Einen Überblick über die Auswirkungen dieser Regulierung auf Informationssysteme geben [KW06]. Neben den Vorschriften zur Rechnungslegung existieren weitere gesetzliche Vorgaben wie Datenschutzgesetze, branchenspezifische Vorschriften (etwa Basel II oder REACH), oder interne Regeln, welche beachtet werden müssen. Zu all diesen unterschiedlichen Regulierungen, welche oft wenig präzis formulierte Bestimmungen enthalten und sich häufig ändern, müssen Unternehmen Compliance nachweisen.

Compliance kann definiert werden als Einhalten aller für das entsprechende Unternehmen relevanten gesetzlichen, behördlichen oder aufsichtsrechtlichen und internen Vorschriften sowie die Beachtung von marktüblichen Standards und Standesregeln zur Schaffung höherer Transparenz und Kontrollierbarkeit der Verhaltensweisen eines Unternehmens und seiner Mitarbeiter (vgl. [Th01, 447; Eb06, 10; JG07, 15]).

Die IT ist davon (neben den Regelungen, die sie direkt erfüllen muss) einerseits durch die Abbildung der Geschäftsprozesse in ihren Systemen und andererseits durch die automatisierte Unterstützung der Prüfung mittels Kontrollen und Berichten (vgl. u.a. [KW06; Kn07]) betroffen. Compliance wird oft in Verbindung mit Corporate Governance gebracht. Aus der Corporate Governance leitet sich die IT-Governance für den IT-Bereich [KL06, 451; Kn07, S98; SS07, 69] und daraus wiederum die SOA-Governance für serviceorientierte Architekturen ab [Ke07, 289; SS07, 69].

4 Compliance-Nachweise in einer SOA

4.1 Mögliche Problemfälle

In einer SOA werden Services in Repositories verzeichnet und gesucht. Es ist demnach wichtig, dass die Services im Repository zu relevanten Regulierungen compliant sind. Ein Service kann von mehreren Prozessen in jeweils verschiedenen Kontexten genutzt werden, indem er *wieder verwendet* wird. Neben internen können auch Services von *externen* Anbietern eingesetzt werden. Zunächst stellt sich die Frage, was in einer SOA unter einem compliant Service zu verstehen ist. Dies verdeutlicht folgendes Beispiel eines Gefahrenpotenzials für den SOX:

"The company has a standard sales contract, but sales personnel frequently modify the terms of the contract. Sales personnel frequently grant unauthorized and unrecorded sales discounts to customers without the knowledge of the accounting department. These amounts are deducted by customers in paying their invoices and are recorded as outstanding balances on the accounts receivable aging. Although these amounts are individually insignificant, they are material in the aggregate and have occurred consistently over the past few years" [Pc04, 261].

Bezogen auf eine SOA bedeutet dies beispielsweise, dass es einen separaten "Rabatt-Service" gibt, der nur von autorisierten Mitarbeitern aufgerufen werden kann. Dieser nimmt eine korrekte Verbuchung des Rabatts im System vor und reduziert damit den Rechnungsbetrag. Der nachfolgende "Rechnungsstellungs-Service" stellt die Rechnung auf den (Rest-) Betrag aus, der im System vorhanden ist. Die Compliance beim "Rabatt-Service" wird sichergestellt, indem die Zugriffsauthorisierungen, die Übereinstimmung mit dem Standard-Verkaufsvertrag sowie die korrekte Verbuchung im System überprüft werden. Beim "Rechnungsstellungs-Service" geschieht dies, indem sichergestellt wird, dass der Rechnungsbetrag aus dem System geholt und nicht manuell eingegeben oder verändert werden kann. Diese Prüfungen können zur Entwicklungszeit stattfinden, so dass die Services erst nach erfolgreicher Prüfung im Repository registriert werden. Das bedeutet, dass dem mit einer organisatorischen Maßnahme, einem Genehmigungsprozess für die Produktivsetzung der Services, begegnet werden kann.

Einen Sonderfall stellt der Bezug von *externen* Services dar. Wird beispielsweise zur Laufzeit der Service-Anbieter gewechselt, muss dieser ebenfalls den Anforderungen der Regulierungen genügen und der Service-Nutzer muss dies sicherstellen. Er muss sich also für compliance-relevante Aufgaben darum kümmern, wie die Leistung erbracht wird [Do05, 261; Kn07, S101]. Das widerspricht einem wichtigen Merkmal der SOA: Die Trennung der Schnittstelle (*was*, für den Service-Nutzer relevant) von der Implementierung (*wie*, erledigt durch den Service-Anbieter) (vgl. z.B. [Er04, 37]). Auch dieses Problem kann mit einer organisatorischen Maßnahme gelöst werden: Durch Einschränkung der zur Auswahl stehenden Services. Beispielsweise dürfen nur Services aus dem unternehmenseigenen Repository verwendet werden. Externe Services werden wie die internen geprüft, bevor sie in dieses Repository aufgenommen werden.

Selbst wenn alle Services im Repository compliant sind, gilt das nicht automatisch für alle Prozesse, welche diese Services verwenden. Nämlich dann, wenn Services in einem zuvor nicht berücksichtigten *Kontext* in einem anderen Prozess *wieder verwendet* werden. Auch diese Situation wird mit einem Beispiel verdeutlicht:

Ein Service, welcher Log-Daten speichert, wird auf Compliance geprüft, bevor er in Produktion geht. Später wird der Service in einem Bestellprozess wieder verwendet, in dem er neben den allgemeinen Bestellinformationen auch Kreditkartendaten speichert. Nun unterliegt derselbe Service beispielsweise zusätzlich den Regeln des Payment Card Industry Data Security Standard (PCI DSS), ohne dass er selber verändert und neu in Produktion gebracht wurde. Das Einzige, was sich geändert hat, ist die Art der Verwendung, also der Kontext [Fo06]. Die PCI-DSS-Regeln untersagen beispielsweise das Speichern bzw. Aufbewahren von Daten wie vollständige Magnetstreifen-Information oder PIN [Pc08].

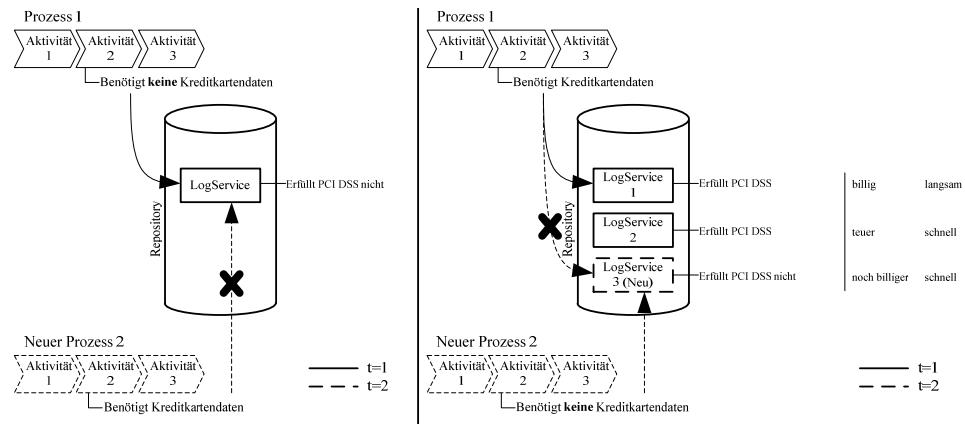


Abbildung 1: Auswirkungen von bestehenden Services auf neue Prozesse und umgekehrt

Die linke Seite der Abbildung 1 zeigt, dass es nicht genügt, das Angebot (die Services bzw. die Applikationen, welche die Services implementiert haben) alleine zu überprüfen, sondern dass zusätzlich der Kontext (Prozesse/Applikationen), in dem die Services genutzt werden, zu berücksichtigen ist. Eine logische Folgerung wäre, analog zu den Services die (neu erstellten oder geänderten) Prozesse zu überprüfen, bevor sie in das produktive System gelangen. Also wiederum eine organisatorische Maßnahme, ein Genehmigungsprozess für die Produktivsetzung der Prozesse. Jedoch genügt dies alleine nicht, wie der umgekehrte Fall (veranschaulicht in der rechten Seite der Abbildung 1) zeigt: Ein bestehender Prozess 1, welcher Kreditkartendaten benötigt, hat zwei bestehende Log-Services, einen billigeren, langsameren und einen teureren, schnelleren zur Auswahl. Beide erfüllen die geforderte PCI-DSS-Regulierung. Der Prozess sucht den

billigsten Service (zu Wirtschaftlichkeitsbewertungen alternativer Geschäftsprozesskonfigurationen vgl. [Br08]). Der Vorteil dieser dynamischen Wahl ist, dass neue oder verbesserte Services ohne Code-Änderung im aufrufenden Prozess genutzt werden können. Wird beispielsweise Log-Service 2 verbessert und neu am billigsten angeboten, wird dieser bei der nächsten Ausführung des Prozesses 1 automatisch berücksichtigt. In der Ausgangssituation fällt die Wahl auf Log-Service 1. Später wird ein weiterer Prozess 2 erstellt, der einen eigenen Log-Service 3 einbringt. Da dieser unterschiedliche nicht-funktionale Eigenschaften (Preis, Geschwindigkeit: Er ist noch billiger und trotzdem schnell, da er die Aussortierung kritischer Daten unterlässt) hat, wird er zusätzlich bewilligt. Er erfüllt die PCI-DSS-Regulierung nicht, was für den Prozess 2 kein Problem darstellt, da er keine Kreditkartendaten benötigt. Jedoch greift Prozess 1 beim nächsten Aufruf automatisch auf den nun billigsten, jedoch für ihn nicht konformen, Log-Service 3 zu: Durch Änderung des Service-Bestands im Repository wird zur Laufzeit automatisch ein nicht konformer Service gewählt.

4.2 Auswirkungen der dynamischen Bindung auf die Compliance

Wie nachfolgend ausgeführt wird, kann die dynamische Bindung verschiedene Ausprägungsformen aufweisen. Je nach eingesetzter Form entsteht das Problem der Wiederverwendung in einem anderen Kontext gar nicht bzw. kann mit organisatorischen Maßnahmen gelöst werden. Bevor auf diese Formen eingegangen wird, wird der Begriff der dynamischen Bindung definiert.

In einer SOA wird bei einer *Bindung* von Services zu Prozessen bzw. höherwertigen Services von Implementierungsdetails (wie beispielsweise der physischen Adresse) abstrahiert, indem diese nicht fest im Quellcode des implementierten Prozesses sondern in einem separaten Dokument, der Service-Beschreibung, im Repository enthalten sind. Im Quellcode wird nur angegeben, auf welche Service-Beschreibung zugegriffen werden soll [PA05, 152ff; TS07, 18]. Diese Abstrahierung ermöglicht erst die Bindung zur Laufzeit (*dynamische Bindung*).

Analog zum oben stehenden Beispiel, der Abstrahierung von der physischen Adresse, liegt für [TS07, 18f] eine dynamische Bindung dann vor, wenn ein Service-Consumer erst zur Laufzeit die Adresse des Service-Providers ermittelt. [Do05, 9] gehen einen Schritt weiter und verstehen unter diesem Begriff, dass zum Zeitpunkt der Code-Generierung meist nicht bekannt ist, welche Services überhaupt, beispielsweise aufgrund externer Einflüsse oder Benutzerpräferenzen, zur Laufzeit aufgerufen werden.

[PA05, 158f] untergliedern statische (Bindung zur Entwicklungszeit) und dynamische Bindung detaillierter: Von der Bindung zur "Registration time" (die Art, wie ein Service resp. seine Beschreibung im Repository erfasst wird, beeinflusst seine Auffindbarkeit) bis zur Bindung zur "Invocation time" (die Entscheidung, welcher Service benutzt wird, wird zum letztmöglichen Zeitpunkt, beim Serviceaufruf, gefällt). Sie vermerken, dass in den meisten Fällen die Bindung zur "Invocation time" als dynamische Bindung bezeichnet wird. In diesem Beitrag wird unter dynamischer Bindung ebenfalls die Bindung zum spätestmöglichen Zeitpunkt verstanden.

Neben dem *Zeitpunkt* der Servicebindung (hier: dynamische Bindung) kann weiter unterschieden werden, wie viel Information im Quellcode enthalten ist, um den entsprechenden Service bzw. dessen Beschreibung im Repository zu identifizieren. Das bedeutet, dass verschiedene *Formen* der dynamischen Bindung unterschieden werden können.

4.2.1 Bindung anhand des Namens

Im einfachen Fall wird genau *ein* Service eindeutig anhand seines *Namens* im Repository identifiziert. Dies wird als "Binding by reference" [PA05, 155] bzw. "Runtime service lookup by name" [KBS04, 63] bezeichnet. Gemäß [KBS04, 63] ist die Service-Definition (wie der Service anzusprechen ist und was er als Resultat liefert) zur Entwicklungszeit dem Service-Nutzer bekannt und wird entsprechend berücksichtigt. In diesem Fall besteht das dynamische Element lediglich darin, dass der physische Ausführungsort (die Adresse) des Services nicht im vornherein bestimmt ist. So kann sich dieser ändern, ohne dass der Quellcode des implementierten Prozesses geändert werden muss. Ebenso besteht die Möglichkeit, denselben Service auf verschiedenen Maschinen bereitzustellen (in der Service-Beschreibung können mehrere physische Adressen, sogenannte "Endpoints", angegeben werden) und entsprechend der Auslastung den einen oder anderen Ausführungsort zu wählen (vgl. z.B. [Si07, S113]). Des Weiteren kann der Anbieter gewechselt werden, indem das Angebot im Repository entsprechend angepasst wird. Wird der Service bzw. dessen Beschreibung des neuen Anbieters unter demselben Namen abgelegt (und der "alte" Service entfernt), so greift der Prozess beim nächsten Aufruf automatisch auf den neuen Service zu.

Compliance-Nachweise: Wird als organisatorische Maßnahme die Compliance der Services geprüft, bevor diese im Repository registriert werden, ist diese Form der dynamischen Bindung eher unproblematisch: Der Service ist eindeutig bestimmt. Die dynamische Auswahl betrifft lediglich unterschiedliche Ausführungsorte. Das Problem der Wiederverwendung in einem anderen Kontext kann nicht entstehen, da zur Laufzeit kein anderer Service gewählt werden kann. Im Beispiel mit den Log-Services ist Log-Service 1 anhand des Namens eingebunden, egal mit welchen Services das Repository erweitert wird, es wird immer auf diesen zugegriffen (rechte Seite der Abbildung 1). Da davon ausgegangen werden kann, dass nicht nur die Services, sondern auch die Prozesse selbst vor der produktiven Einführung auf die Erfüllung der vorgeschriebenen Regulierungen geprüft werden, kann im umgekehrten Fall (linke Seite der Abbildung 1) das Problem mit dieser organisatorischen Maßnahme gelöst werden.

4.2.2 Bindung anhand Bedingungen oder Eigenschaften

Im komplexen Fall werden die Services nach *Bedingungen* oder *Eigenschaften* im Repository gesucht. Dementsprechend wird diese Art als "Binding by constraint" [PA05, 156] bzw. als "Runtime service discovery based on reflection" [KBS04, 63f] bezeichnet. Krafczig et al. [KBS04] fügen eine weitere Zwischenform ein, die sie als "Runtime service lookup by properties" bezeichnen. Dabei wird zwar bereits nach Eigenschaften gesucht, aber nur innerhalb einer im Quellcode festgelegten Vorauswahl von Services und nicht innerhalb des gesamten aktuellen Bestands im Repository. Da bereits eine Vorauswahl getroffen wurde, ist es, im Gegensatz zur offenen Suche, nicht zwingend nötig, die Semantik bei der Suche zu berücksichtigen [KBS04, 63f; PA05, 156]. Ein Beispiel für diesen Typ liefern [Sp07] und [Br08, 104ff]: Um eine Aktivität in einem Geschäftsprozess abzuwickeln, stehen mehrere Services mit identischer Funktionalität, jedoch unterschiedlichen nicht-funktionalen Eigenschaften, wie Ausführungszeit oder Kosten, zur Verfügung (Vorauswahl). Wie im Beispiel mit den Log-Services besteht die Möglichkeit, zwischen einem langsameren, billigeren oder einem schnelleren, teureren Service zu wählen. Anhand der Präferenzen und Restriktionen wird die passende Service-Komposition automatisch ausgewählt. Die Festlegung, welche Services von der Funktionalität her identisch sind, wird manuell hinterlegt [Sp06, 5; Sp07, 316f; Br08, 110f]. Werden *mehrere* möglich anzuwendende Services als Resultat der Suche zurückgegeben, müssen diese evaluiert und ein Service ausgewählt werden. Beispielsweise kann diese Auswahl auf die Dienstgüte der Services, die Reputation der Anbieter oder die Kosten Bezug nehmen [Be05, 269ff; PA05, 156f; Br08, 104f].

Compliance-Nachweise: Bei der Zwischenform ist die Situation ähnlich wie bei der Bindung anhand des Namens. Anstelle eines einzelnen Services ist hier eine Menge von Services vorgegeben. Die Auswahl sollte mithilfe von Dokumentation begründbar sein (Angabe der Kriterien oder Regeln). Bei der weitreichendsten Form jedoch, der offenen Suche, kann das Problem der Wiederverwendung in einem anderen Kontext nicht alleine mit der Überprüfung des Prozesses vor Produktivsetzung gelöst werden, wie das Beispiel der Log-Services zeigt: Da anhand der Eigenschaft Kosten bei jedem Aufruf von neuem der billigste Service innerhalb des gesamten aktuellen Bestands des Repositorys gesucht wird, kann aufgrund einer Änderung des Bestands beim nächsten Aufruf ein nicht konformer Service gewählt werden, wie im Beispiel der Log-Service 3 für den Prozess 1 (vgl. Abbildung 1, rechte Seite).

Zwar wird diese Form der dynamischen Bindung in der Praxis kaum angewandt [KBS04, 64], u.a. weil semantische Konzepte benötigt werden, von denen noch nicht viele Implementierungen existieren (vgl. z.B. [Be07, 70; HKZ08]). Jedoch ist wünschenswert, dass zukünftig anhand von Zielvorgaben möglichst automatisiert Prozesse erstellt, geändert oder gelöscht werden können (vgl. z.B. [PLL06, 6]). Dafür wird eine Suche (und Einbindung) von Services anhand Bedingungen oder Eigenschaften benötigt. Aus diesem Grund sollten die daraus entstehende Compliance-Problematik frühzeitig erkannt und diesbezügliche Lösungsansätze erstellt werden.

4.3 Lösungsansätze

Eine Lösungsmöglichkeit sind die in Abschnitt 4.1 erwähnten *organisatorischen* Maßnahmen, Genehmigungsprozesse für Services und Prozesse sowie Einschränkung der zur Auswahl stehenden Services. Jedoch genügen diese je nach angewandter Form der dynamischen Bindung nicht, wie im vorhergehenden Abschnitt erläutert wurde. Zudem sind sie aus Gründen der Wirtschaftlichkeit und der Fehleranfälligkeit, falls möglich, um *technische* Hilfsmittel zu ergänzen oder gar zu ersetzen.

Es gibt einige *Tools* auf dem Markt, welche die Compliance unterstützen sollen. Einen Überblick über SOX-Software-Produkte geben [Ag06]. Jedoch ist zu beachten, dass die wenigsten Produkte Funktionalitäten wie die Überprüfung eines Geschäftsprozesses zur Laufzeit beinhalten [Fo06]. Zudem müssen Kontrollaktivitäten manuell implementiert werden [Ag06]. Ob es möglich ist, *alle* Regulierungen in *allen* Aspekten detailliert zu modellieren und zur Laufzeit zu überprüfen, ist fraglich.

Bereits in Abschnitt 2 wurden Lösungsansätze und ihre Grenzen für dieses Problem betrachtet. Bei allen Ansätzen müssen zudem die zu unterstützenden Regulierungen manuell bestimmt werden. Wird zur Laufzeit ein Service ausgewählt und soll gleichzeitig die Auswahl eines nicht konformen Service verhindert werden, wird eine technische Unterstützung für dieses Problem benötigt, denn, wie das Beispiel der Log-Services zeigt, sind die zu erfüllenden Regulierungen kontextabhängig. Das bedeutet, dass zuerst der Kontext bestimmt werden muss, um in einem weiteren Schritt daraus die zu erfüllenden Regulierungen abzuleiten, welche dann durch den Service erfüllt werden müssen.

Da es sich um die Nutzung des Services handelt, hängt der *Kontext* von den Eingabedaten ab, also dem Wert der Inputparameter. Eine Möglichkeit wäre, ihn anhand des Datentyps der zu liefernden Daten zu bestimmen; dies setzt jedoch voraus, dass die Datentypen genau festgelegt werden. Eine weitere Möglichkeit wäre, die Daten selbst auf Muster zu überprüfen, jedoch ist fraglich, ob daraus ein Kontext eindeutig bestimmbar ist. Die wohl beste Variante ist, nicht nur das Angebot, sondern auch die *Anfrage*, den aufzurufenden Prozess, der wiederum als Service betrachtet werden kann, semantisch zu beschreiben. Dabei muss eine Beschreibungssprache gewählt werden, welche die Anfragen getrennt behandelt.

Web Services, die zurzeit aktuelle Implementierungsform von Services in einer SOA, werden größtenteils in der Sprache WSDL beschrieben. Aus einer WSDL-Datei können jedoch nur syntaktische Informationen, nicht jedoch ihre Bedeutung (Semantik) gelesen werden. Um Web Services semantisch zu beschreiben, gibt es standardisierte Beschreibungssprachen. Die wichtigsten sind OWL-S, WSMO und WSDL-S. Sie werden in [Kl06] und [PLL06] erläutert und verglichen. Einen neueren Ansatz stellt SAWSDL dar.

Als nächster Schritt muss die Beziehung zu den zu erfüllenden *Regulierungen* hergestellt werden. Dafür eignet sich eine Ontologie; dies ist eine formale Beschreibung von Begriffen (Konzepten) und ihren Beziehungen (Relationen), welche für eine Gruppe von Personen (Domäne) Gültigkeit besitzt [Gr93; MSS01; He02]. In den Semantic-Web-Services-Beschreibungssprachen können Referenzen zu solchen Domänen-Ontologien hergestellt werden. In einer Ontologie können auch Synonyme berücksichtigt werden: So können Benutzer anstelle des allgemeinen "Kreditkartendaten"-Begriffs auch Kreditkartenbezeichnungen wie "Visa" oder "MasterCard" angeben. Für dieses Anwendungsbereich (Domäne) sind die beiden Begriffe gleichbedeutend. Vorzugsweise sind bestehende Ontologien anzuwenden oder als Basis zu nehmen, um den Entwicklungs- und Wartungsaufwand zu verringern. In diesem Fall bieten sich zunächst Rechtsontologien an (vgl. z.B. [GST05]); diese sind jedoch stark auf juristische Fälle oder Situationen wie das Handeln im Affekt ausgerichtet. Demnach ist eine Ontologie vorzuziehen, welche sich auf Regulierungen konzentriert, zu denen Unternehmen ihre Compliance nachweisen müssen. Zu prüfen ist beispielsweise die Eignung der in [Gi05] beschriebenen Ontologie.

Eine der vorgeschlagenen organisatorischen Maßnahmen lautete, dass alle Services im Repository compliant sein müssen. Jedoch ist mit einer solchen Forderung weiterhin unklar, zu welchen Regulierungen ein Service compliant ist. Um diese Situation zu klären, müsste die Beschreibung der *angebotenen* Services entsprechend erweitert werden.

Damit ein automatisierter Abgleich zwischen Anfrage und Angebot gemacht werden kann, sollte dieselbe Beschreibungssprache verwendet werden. Ebenso sollte auf dieselbe Ontologie Bezug genommen werden. Für extern bezogene Services könnte die Angabe des Partnerunternehmens, zu welchen Regulierungen ihr Service compliant ist, durch eine Prüfungs- oder Zertifizierungsgesellschaft bestätigt werden. In Abbildung 2 ist dies dargestellt: Sowohl der anfragende Prozess (Eingabedaten in einem spezifischen Kontext) wie auch der angebotene Web Service (zu welchen Regulierungen compliant) werden semantisch beschrieben und beziehen sich auf dieselbe Ontologie, in der die Zuordnung vom Kontext zu den benötigten Regulierungen vorgenommen wird.

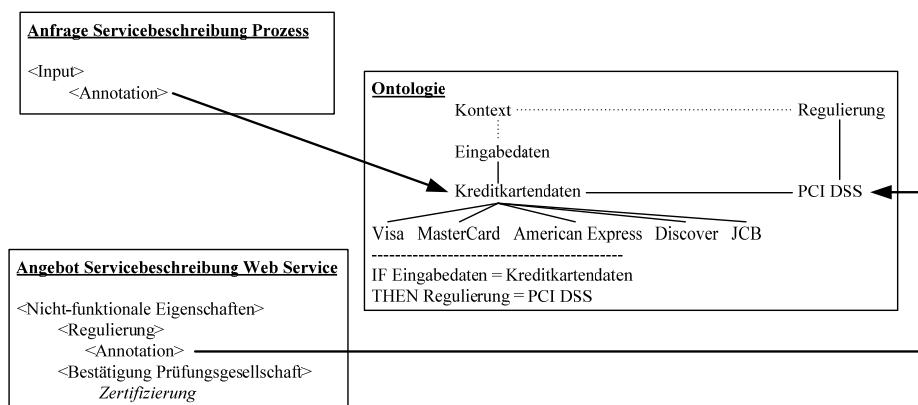


Abbildung 2: Lösungsskizze

Eine Ontologie sollte möglichst breit einsetzbar sein (abgebildete Regulierungen, nutzende Unternehmen). Da es Regulierungen gibt, welche abhängig von der Branche oder anderen Kriterien erfüllt werden müssen, wird in einem weiteren Schritt der Kontext um die Unternehmensart erweitert. Das bedeutet, dass, je nachdem in *welchem Unternehmen* und mit *welchen Daten* der Service eingesetzt wird, u.U. andere Regulierungen relevant sind. Zur Prüfung einer Regulierung (und allfälliger damit einhergehender Zertifizierung) werden oft Prüfstandards angewandt, in denen genaue Testprozeduren angegeben werden. Prüfstandards können sich auf Frameworks beziehen.

In Abbildung 3 wird dies veranschaulicht, wobei (oberhalb der Rechtecke) das Beispiel mit den Kreditkartendaten abgebildet ist. PCI DSS muss von allen Unternehmen (z.B. Händler) erfüllt werden, welche Kreditkartendaten einsetzen. Die Anforderungen basieren auf dem ISO/IEC-17799-Standard (welcher in der Zwischenzeit weiterentwickelt und als ISO/IEC 27002 veröffentlicht wurde). Die Applikationsanbieter (also auch Service-Anbieter) werden nach dem Payment Application Data Security Standard (PA-DSS) geprüft und zertifiziert, sofern die Software extern angeboten wird [Pc08]. Auch bei Eigenentwicklungen muss der PCI DSS eingehalten werden, jedoch wird für den unternehmensinternen Gebrauch keine Service-Zertifizierung (Bestätigung) benötigt.

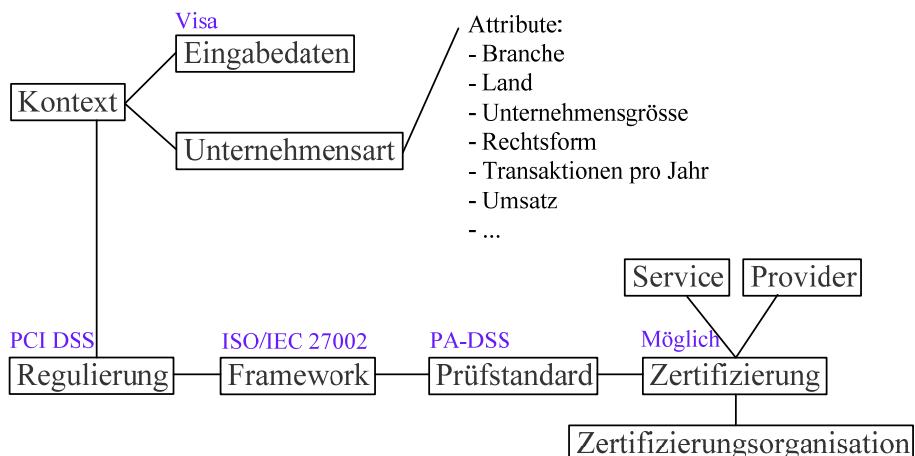


Abbildung 3: Ontologieskizze

5 Zusammenfassung und Ausblick

Die dynamische Bindung, die semantischen Konzepte, und die Governance [Ab07, 7f] gehören zu den "Grand Challenges" der Forschung in Bezug auf serviceorientierte Architekturen [Pa06]. Dementsprechend fehlen Anwendungen der Semantik und damit verbunden der dynamischen Bindung in der Praxis zurzeit weitgehend [Pa06; Ku08, 1]. Jedoch sollten mögliche Probleme schon heute beachtet werden.

In diesem Beitrag wurde gezeigt, dass die verschiedenen Ausprägungsformen der dynamischen Bindung unterschiedliche Auswirkungen auf Compliance-Nachweise besitzen. Bei Anwendung der weitreichendsten Form der dynamischen Bindung besteht die Gefahr, dass bei Wiederverwendung von Services in einem anderen Kontext nicht konforme Services gewählt und daher nicht alle erforderlichen Regulierungen erfüllt werden. Da bisherige Ansätze dieses Problem nicht lösen, wurde ein Ansatz vorgestellt, wie mithilfe von semantischen Konzepten der Kontext der Servicenutzung und, daraus folgend, die anzuwendenden Regulierungen bestimmt werden können. Dabei zeigt sich, dass semantische Konzepte nicht nur bei Auswahl der Services, sondern auch für Compliance-Nachweise hilfreich sein können.

Als nächster wichtiger Schritt muss die Ontologie, welche in Abbildung 3 grob skizziert wurde, erstellt oder auf Basis einer anderen Ontologie aufgebaut werden. Rechtsontologien erscheinen als Basis wenig geeignet; zu prüfen sind Regulierungsontologien. Dabei müssen auch die Ontologiebeschreibungssprache sowie ihre Ausdrucksmächtigkeit berücksichtigt werden. Weiter müssen eine geeignete semantische Service-Beschreibungssprache sowie die relevanten Sprachelemente gewählt werden. Nach einem "proof of concept" könnte in einem zusätzlichen Schritt geprüft werden, ob die Ontologie (semi-)automatisch aus Regulierungs-Texten extrahiert werden kann.

Literaturverzeichnis

- [Ab07] Aberdeen Group: Management and Governance: Planning for an Optimized SOA Application Lifecycle, Report, 2007.
- [Ag06] Agrawal, R.; Johnson, C.; Kiernan, J.; Leymann, F.: Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In (Liu, L.; Reuter, A.; Whang, K.-Y.; Zhang, J. Hrsg.): Proc. 22nd International Conference on Data Engineering (ICDE 2006), Atlanta, 2006. IEEE Computer Society, 2006; S. 92.
- [Be05] Berbner, R.; Heckmann, O.; Mauthe, A.; Steinmetz, R.: Eine Dienstgüte unterstützende Webservice-Architektur für flexible Geschäftsprozesse. In: WIRTSCHAFTSINFORMATIK 47 (2005) 4; S. 268-277.
- [Be07] Bell, D.; De Cesare, S.; Iacovelli, N.; Lycett, M.; Merico, A.: A framework for deriving semantic web services. In: Information Systems Frontiers 9 (2007) 1; S. 69-84.
- [Br08] vom Brocke, J.: Serviceorientierte Architekturen – SOA – Management und Controlling von Geschäftsprozessen. Vahlen, München, 2008.
- [Do05] Dostal, W.; Jeckle, M.; Melzer, I.; Zengler, B.: Service-orientierte Architekturen mit Web Services – Konzepte – Standards – Praxis. Spektrum – Akademischer Verlag, München, 2005.
- [Eb06] Eidg. Bankenkommission (EBK): Rundschreiben 06/6 – Überwachung und interne Kontrolle, http://www.ebk.admin.ch/d/regulier/rundsch/2006/rs_0606_d.pdf, 2006.
- [Er04] Erl, T.: Service-Oriented Architecture – Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River et al., 2004.
- [Fo06] Foody, D.: The Challenges of SOA – Which rules are necessary and which are just nice to have. In: SOA Web Services Journal 6 (2006) 9; <http://webservices.sys-con.com/read/284550.htm>.
- [Gi05] Giblin, C.; Liu, A.Y.; Müller, S.; Pfitzmann, B.; Zhou, X.: Regulations Expressed As Logical Models (REALM), Research Report, IBM Research, <http://www.zurich.ibm.com/security/publications/2005/GiLiMuPfZh05REALM-Report.pdf>, 2005.
- [Gr93] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition 5 (1993) 2; S. 199-220.
- [GST05] Gangemi, A.; Sagri, M.; Tiscornia, D.: A constructive framework for legal ontologies. In (Benjamins, V.; Casanovas, P.; Breuker, J.; Gangemi, A. Hrsg.): Law and the Semantic Web. Springer, Berlin, Heidelberg, 2005; S. 97-124.
- [He02] Hesse, W.: Ontologie(n) – Aktuelles Schlagwort. In: Informatik Spektrum 25 (2002) 6; S. 477-480.
- [HKZ08] Haniewicz, K.; Kaczmarek, M.; Zyskowski, D.: Semantic Web Services Applications – a Reality Check. In: WIRTSCHAFTSINFORMATIK 50 (2008) 1; S. 39-46.
- [JG07] Johannsen, W.; Goeken, M.: Referenzmodelle für IT-Governance – Strategische Effektivität und Effizienz mit COBIT, ITIL & Co. dpunkt.verlag, Heidelberg, 2007.
- [Jo08] Josuttis, N.: SOA in der Praxis – System-Design für verteilte Geschäftsprozesse. dpunkt.verlag, Heidelberg, 2008.
- [KBS04] Krafzig, D.; Banke, K.; Slama, D.: Enterprise SOA – Service-Oriented Achitecture Best Practices. Prentice Hall PTR, Upper Saddle River et al., 2004.
- [Ke07] Keller, W.: SOA-Governance – SOA langfristig durchsetzen und managen. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis service-orientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 289-307.
- [Kl06] Klein, M.: Automatisierung dienstorientierten Rechnens durch semantische Dienstbeschreibungen, Dissertation, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena. Universitätsverlag Karlsruhe, Karlsruhe, 2006.

- [KL06] Knolmayer, G.; Loosli, G.: IT Governance. In (Zaugg, R.J. Hrsg.): Handbuch Kompetenzmanagement. Durch Kompetenz nachhaltig Werte schaffen. Haupt Verlag, Bern, Stuttgart, Wien, 2006; S. 449-457.
- [Kn07] Knolmayer, G.: Compliance-Nachweise bei Outsourcing von IT-Aufgaben. In: WIRTSCHAFTSINFORMATIK 49 (2007) Sonderheft; S. S98-S106.
- [Ku08] Kuropka, D.; Tröger, P.; Staab, S.; Weske, M. (Hrsg.): Semantic Service Provisioning. Springer, Berlin, Heidelberg, 2008.
- [KW06] Knolmayer, G.; Wermelinger, T.: Der Sarbanes-Oxley Act und seine Auswirkungen auf die Gestaltung von Informationssystemen. In (Siegel, T.; Klein, A.; Schneider, D.; Schwintowski, H.-P. Hrsg.): Unternehmungen, Versicherungen und Rechnungswesen. Duncker&Humblot, Berlin, 2006; S. 513–536.
- [LMX07] Liu, Y.; Müller, S.; Xu, K.: A static compliance-checking framework for business process models. In: IBM SYSTEMS JOURNAL 46 (2007) 2; S. 335-361.
- [MSS01] Mädche, A.; Staab, S.; Studer, R.: Ontologien. In: WIRTSCHAFTSINFORMATIK 43 (2001) 4; S. 393-395.
- [NS07] Namiri, K.; Stojanovic, N.: A Model-driven Approach for Internal Controls Compliance in Business Processes. In (Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. Hrsg.): Proc. of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), Innsbruck, 2007. CEUR Workshop Proceedings, Vol. 251, 2007; S. 40-43.
- [Og04] O'Grady, S.: SOA Meets Compliance: Compliance Oriented Architecture, Study, Red-Monk, http://redmonk.com/public/COA_Final.pdf, 2004.
- [PA05] Pautasso, C.; Alonso, G.: Flexible Binding for Reusable Composition of Web Services. In (Gschwind, T.; Assmann, U.; Nierstrasz, O. Hrsg.): Proc. 4th International Workshop on Software Composition (SC 2005), Edinburgh, 2005. Springer, Berlin, Heidelberg, 2005; S. 151-166.
- [Pa06] Papazoglou, M.P.; Traverso, P.; Dustdar, S.; Leymann, F.; Krämer, B.J.: Service-Oriented Computing Research Roadmap. In (Cubera, F.; Krämer, B.J.; Papazoglou, M.P. Hrsg.): Proc. Dagstuhl Seminar 05462 - Service Oriented Computing (SOC), Dagstuhl, 2006. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Dagstuhl, <http://drops.dagstuhl.de/opus/volltexte/2006/524/pdf/05462.SWM.Paper.524.pdf>, 2006.
- [Pc04] Public Company Accounting Oversight Board (PCAOB): AUDITING STANDARD No. 2 – An Audit of Internal Control Over Financial Reporting Performed in Conjunction with An Audit of Financial Statements, http://www.pcaobus.org/Rules/Rules_of_the_Board/Auditing_Standard_2.pdf, 2004.
- [Pc08] Payment Card Industry (PCI) Security Standards Council: Payment Application Data Security Standard (PA-DSS), http://www.pcisecuritystandards.org/security_standards/pa_dss.shtml, 2008.
- [PG03] Papazoglou, M.P.; Georgakopoulos, D.: Service-Oriented Computing. In: Communications of the ACM 46 (2003) 10; S. 25-28.
- [PLL06] Polleres, A.; Lausen, H.; Lara, R.: Semantische Beschreibung von Web Services. In (Pellegrini, T.; Blumauer, A. Hrsg.): Semantic Web – Wege zur vernetzten Wissensgesellschaft. Springer, Berlin, Heidelberg, 2006; S. 505-524.
- [Si07] Siedersleben, J.: SOA revisited: Komponentenorientierung bei Systemlandschaften. In: WIRTSCHAFTSINFORMATIK 49 (2007) Sonderheft; S. S110-S117.
- [Sp06] Spahn, M.; Berbner, R.; Heckmann, O.; Steinmetz, R.: Ein heuristisches Optimierungsverfahren zur dienstgütebasierten Komposition von Web-Service-Workflows, Technical Report, Technische Universität Darmstadt, <ftp://ftp.kom.e-technik.tu-darmstadt.de/pub/papers/SBHS06-1-paper.pdf>, 2006.

- [Sp07] Spahn, M.: Ein heuristisches Verfahren zur dienstgütebasierten Optimierung flexibler Geschäftsprozesse. In (Braun, T.; Carle, G.; Stiller, B. Hrsg.): Proc. 15. Fachtagung Kommunikation in Verteilten Systemen (KiVS 2007), Bern, 2007. Springer, Berlin, Heidelberg, 2007; S. 315-322.
- [SS07] Schelp, J.; Stutz, M.: SOA Governance. In: HMD - Praxis der Wirtschaftsinformatik 43 (2007) 253; S. 66-73.
- [Th01] Thelesklaf, D.: Outsourcing von Compliance-Dienstleistungen – Compliance als Teil des Risk Managements. In: Der Schweizer Treuhänder 75 (2001) 5; S. 447-452.
- [TS07] Tilkov, S.; Starke, G.: Einmaleins der serviceorientierten Architekturen. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 9-36.
- [WS07] Winter, R.; Schelp, J.: Agilität und SOA. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 41-47.

Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle

Patrick Delfmann, Sebastian Herwig, Lukasz Lis, Armin Stein

European Research Center for Information Systems
Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3
48149 Münster

[vorname.nachname]@ercis.uni-muenster.de

Abstract: Bei der verteilten Erstellung von fachkonzeptionellen Informationsmodellen können, insbesondere aufgrund der Subjektivierung, beim Vergleich und der anschließenden Zusammenführung von Teilmodellen eine Reihe von Vergleichskonflikten (z. B. Namens-, Detaillierungs- oder Typkonflikte) auftreten, deren Auflösung Gegenstand zahlreicher methodischer Arbeiten ist. Diese betrachten jedoch ausschließlich die Problembehandlung zur Zeit der Konsolidierung. In diesem Artikel wird eine Methode vorgestellt, die Probleme, welche bei der nachträglichen Auflösung von Namenskonflikten aufgrund semantischer Diversitäten auftreten, bereits während der Modellierung verhindert. Die Methode ist generisch und somit für jede Modellierungstechnik und jede natürliche Sprache anwendbar.

1 Motivation

Für die erfolgreiche und effiziente Entwicklung von Informationssystemen sind im Vorfeld erstellte fachkonzeptionelle Modelle eine allgemein anerkannte Grundlage [KK84, Ka88]. Durch den gewachsenen Umfang der Projekte werden diese Modelle jedoch in der Regel nicht mehr von einzelnen Personen an einem Ort, sondern sowohl zeitlich als auch räumlich verteilt entwickelt, was eine Effizienzsteigerung gegenüber der Einzelentwicklung mit sich führen soll [Br03]. Außerdem werden zur optimalen Erfassung der fachlichen Anforderungen zunehmend auch Fachvertreter an der Erstellung der Modelle beteiligt, so dass Ergebnisse unterschiedlicher Modellierungsparadigmen und -kenntnisse in die Modelle einfließen [Fi04, WJ04]. Hierdurch entstehen Teilmodelle, die anschließend zu einem Gesamtmodell integriert werden müssen. Diesem Umstand trägt beispielsweise die Entstehung von Entwicklungscommunities wie die Open Model Initiative Rechnung, welche darum bemüht ist, Modelle nicht nur unternehmensintern, sondern auch gemeinschaftlich verteilt über die Unternehmensgrenzen hinweg zu erstellen [FSK07].

Durch die Diversität der Modellierer unterscheiden sich die Modelle häufig in Bezug auf die verwendeten Begriffe sowie in Bezug auf den Abstraktions- und Detaillierungsgrad, wie empirische Studien belegen [HS06]. Diese Missstände führen dazu, dass ein erschwerender Vergleich der Modelle vor einer Zusammenführung notwendig ist, da ansonsten bei der Integration der Modelle im Zuge der Erstellung einer Gesamtbetrachtung des Modellierungsgegenstandes u. a. *Namenskonflikte* (*Homonym-, Synonym- und Abstraktionskonflikte*), *Typkonflikte* und *Detaillierungskonflikte* auftreten können [Pf08]. Um diese Konflikte zu lösen, muss eine aufwändige manuelle Anpassung unter Einbeziehung sämtlicher an dem Projekt beteiligter Modellierer und Fachvertreter vorgenommen werden, was den Gedanken der Effizienzsteigerung ad absurdum führt. Eine Vielzahl von wissenschaftlichen Beiträgen beschäftigt sich mit Vorschlägen zur Durchführung dieser Aufgabe (vgl. Abschnitt 2).

Dieser Beitrag betrachtet die Vermeidung von Namenskonflikten schon vor der Integration der Teilmodelle. Es wird eine Methode vorgeschlagen, mit der schon während der Modellierung verhindert werden kann, dass es bei einer an die Modellierung anschließenden Zusammenführung der Teilmodelle zu Namenskonflikten kommt. Dies geschieht durch ein automatisiertes Anleiten des Modellierers bei der Bezeichnung verwendeter Modellelemente mittels Bezeichnungskonventionen. Diese basieren auf bereits im Vorfeld des Modellierungsprojektes festgelegten Begriffs- und Phrasenstrukturkonventionen. Während Begriffskonventionen durch Begriffsmodelle abgebildet werden können, die neben Konventionen zur Verwendung von Substantiven auch solche zur Verwendung weiterer Wortarten spezifizieren, bietet sich für die Definition von Phrasenstrukturkonventionen die Verwendung der Head-driven Phrase Structure Grammar (HPSG) an [PS94]. Hierbei handelt es sich um eine formale Grammatik zur Spezifikation der Syntax natürlicher Sprachen. Somit ist die Methode einerseits von der verwendeten Modellierungssprache, andererseits aber auch von der verwendeten Landessprache unabhängig.

Der Artikel ist folgendermaßen strukturiert: Zunächst werden in Abschnitt 2 verwandte Arbeiten sowie insbesondere der durch diesen Beitrag zu deckende methodische Entwicklungsbedarf diskutiert. Des Weiteren wird die im Rahmen des Beitrags angewendete Forschungsmethode vorgestellt. In Abschnitt 3 erfolgt die konzeptionelle Spezifikation der Methode zur Vermeidung von Namenskonflikten, deren Anwendung in Abschnitt 4 exemplarisch anhand eines Demonstrators gezeigt wird. Abschnitt 5 vermittelt einen Ausblick auf notwendigen, zukünftigen Forschungsbedarf, der sich u. a. den hier noch nicht behandelten Konfliktarten widmen muss.

2 Verwandte und zu leistende Arbeiten; Forschungsmethode

2.1 Verwandte Arbeiten und methodischer Entwicklungsbedarf

Die Ursprünge der Betrachtung von Namenskonflikten sind in Arbeiten der 1980er und 1990er Jahren zu suchen. Als Motivation wurde damals die Notwendigkeit der Integration von Unternehmensdatenbanken gesehen, wobei in einem ersten Schritt zunächst die

vorhandenen Datenbankschemata in ein konsolidiertes Schema überführt werden müssen [BL84, BLN86, BKK91, LB01]. Bei den in diesem Zusammenhang betrachteten Modellierungssprachen handelt es sich häufig um Dialekte des Entity-Relationship-Modells (ERM) [Ch76]. Die Autoren begegnen der Problematik zumeist über einen Vergleich der Bezeichner der in den Datenbankschemata verwendeten Schemaelemente. Eine Automatisierung des semantischen Vergleichs der Elemente wird durch die Verfasser grundsätzlich ausgeschlossen, so dass immer eine manuelle Anpassung durchgeführt werden muss. Semiautomatisierte Lösungen machen es sich zunutze, dass bei der Modellierung mittels ERM in der Regel ausschließlich einzelne Substantive verwendet werden, was die Komplexität des Vergleichs auf einem relativ geringen Niveau hält. Wesentlich komplexer stellt sich die Aufgabe bei der Verwendung gängiger Prozessmodellierungssprachen wie bspw. Ereignisgesteuerte Prozessketten (EPK) [KNS92] dar, bei denen ganze Satzphrasen üblich sind, die zusätzlich zu Substantiven sämtliche möglichen Wortarten enthalten können. Die oben angeführten Beiträge betrachten diese Problematiken in ihren Ansätzen nicht.

Weitere Ansätze zur Behandlung von Namenskonflikten betrachten die Einbindung von Ontologien [Gr93, Gu98] zum semantischen Vergleich der in den Modellen verwendeten Bezeichner. So können bspw. Domänenontologien an die in den Teilmodellen vorkommenen Begriffe gekoppelt werden, um Beziehungen und Ähnlichkeiten zwischen Elementen zu identifizieren und zu explizieren [Ho07]. Ein Begriffsvergleich auf syntaktischer Ebene erfolgt in [EKO07], wobei mittels eines kombinierten Ähnlichkeitsmaßes automatisiert entschieden wird, ob zwei betrachtete Modellelemente identisch sind. Dieses Vorgehen ist jedoch nicht fehlerfrei, da Homonym- und Synonymkonflikte nur durch eine semantische Betrachtung identifiziert werden können, was eine nachträgliche Überarbeitung notwendig macht. Dieses Problem beruht auch darauf, dass es bei der Kopplung mit einer „allgemein akzeptierten“ Ontologie vorkommen kann, dass syntaktisch identische Begriffe der Teilmodelle während der Modellierung mit einer unterschiedlichen Semantik versehen werden. Ein Algorithmus erkennt dann fälschlicherweise Elemente mit gleichem Bezeichner als identisch und somit integrierbar an [KO05, Sa07]. Es reicht demnach nicht aus, wenn die Modellierer sich in der gleichen Domäne bewegen.

Vielmehr ist es für diesen Fall notwendig, dass sich die Modellierer im Vorfeld des Modellierungsprojekts auf eine gültige Ontologie einigen und die hinterlegten Begriffe während der Modellierung konsequent verwendet werden. Um dies zu gewährleisten, ist eine methodische Unterstützung erforderlich, die dem Modellierer jederzeit den Zugriff auf und den Abgleich der Modelle mit der Ontologie erlauben.

Eine solche Lösung liegt bis dato aber nicht vor. Sollte eine solche Lösung existieren, besteht weiterhin das Problem, dass Begriffe falsch verwendet werden, wenn die durch die Ontologie ursprünglich vorgegebene Semantik eine andere ist, als die vom Modellierer intendierte.

Ein restriktiverer Ansatz von [Pf08] sieht vor, zur Vermeidung der Verwendung falscher Begriffe, diese bereits in die Modellierungssprache zu integrieren und anhand von Prozessbausteinen dem Modellierer den kleinstmöglichen Spielraum für Fehler einzuräumen. Die Semantik der Modellelemente ist diesen dann also inhärent. Dieser Vorschlag

wird vom Autor exemplarisch für die Domäne der öffentlichen Verwaltungen in Form der Prozessmodellierungssprache PICTURE konkretisiert. Namenskonflikte sind in dieser Sprache aufgrund der Existenz von exklusiv 26 spezifischen Prozessmodellbausteinen mit definierter Semantik von vornherein ausgeschlossen. Die Eignung solcher Sprachen ist auf überwiegend lineare Geschäftsprozesse mit klaren Strukturen beschränkt. Als kritisch anzumerken ist, dass eine Anwendung solcher vordefinierter Prozessbausteine in unstrukturierten Bereichen, die eine größere Anzahl von Bausteinen erforderlich machen würden, nicht möglich ist, da der Modellierer unter Umständen eine zu geringe Freiheit hat, den Sachverhalt adäquat darzustellen.

Einen freieren Ansatz verfolgen [Ro96, Ku00], die Satzphrasen als Bezeichner vorschlagen. Die eher praxisorientierten Arbeiten stehen im Zusammenhang mit der Modellierung von Geschäftsprozessen, deren Klarheit eine immanente Anforderung im Rahmen von Reorganisationsprojekten ist. Als Beispiele für solche Phrasenstrukturen führen die Autoren bspw. *<Substantiv> + <Verb im Infinitiv>* – also z. B. „Dienstreiseantrag prüfen“ – an. Zur Verbreitung solcher Modellierungskonventionen werden z. B. Methodenhandbücher vorgeschlagen [RSD08]. Die Sicherstellung der *Umsetzung* der in den Handbüchern vermerkten Konventionen wird durch solche Ansätze allerdings nicht adressiert. Folglich besteht die Gefahr, dass solche Konventionen zwar bestehen, aber nicht konsequent eingehalten werden.

Zusätzlich fordern [KR98] Fachbegriffsmodelle – ähnlich wie [KO05, Sa07] Lexika –, die im Vorfeld des Modellierungsprojekts Unternehmensbegriffe eindeutig definieren. Diese sind jedoch, wie bereits bei den Ansätzen der Schemaintegration, lediglich durch Substantive belegt. Einen Schritt weiter gehen die Ansätze von [Pa03, RM05, BKS08]. Diese Autoren schlagen – unabhängig von der eigenen Erstellung von Fachbegriffsmodellen bzw. Lexika – die Anbindung an vorhandene Wörterbücher wie bspw. WordNet [<http://wordnet.princeton.edu>] vor. Hierbei handelt es sich um eine umfangreiche Sammlungen englischer Substantive, Verben und Adjektive sowie deren semantischer Beziehungen. Hierbei geht wiederum der Domänenbezug verloren und es entstehen ähnliche Problematiken wie bereits oben für die Nutzung von Ontologien. Weiterhin werden die Überlegungen lediglich als erste Idee angeführt und nicht hinreichend spezifiziert. Bis-her sind weder eine methodische Umsetzung noch eine Implementierung dieser Ansätze bekannt.

Zusammenfassend lässt sich der folgende methodische Entwicklungsbedarf ableiten, den es zur Vermeidung von Namenskonflikten beim Vergleich von konzeptionellen Informationsmodellen wie folgt zu begegnen gilt:

- *Spezifikation von Begriffskonventionen bezogen auf sämtliche Wortarten (über Substantive hinaus):* Für die Realisierung wird ein Fachkonzept der Begriffsmodellspezifikation vorgestellt.
- *Spezifikation von Phrasenstrukturkonventionen:* Hierfür wird eine Grammatik vorgestellt, die es erlaubt, Phrasenstrukturen zu spezifizieren und mit gültigen Begriffen zu verwenden.

- *Kopplung der Konventionen mit fachkonzeptionellen Modellierungssprachen:* Eine fachkonzeptionelle Spezifikation zur Integration der Bezeichnungskonventionen mit Modellierungssprachen soll diesem Mangel begegnen.
- *Überwachung der Einhaltung dieser Konventionen:* Hierfür wird ein Verfahren vorgestellt, das diese methodische Unterstützung ermöglicht.

2.2 Forschungsmethode

Der vorliegende Beitrag folgt dem Design-Science-Forschungsansatz [He04], der im Kern die Konstruktion von *wissenschaftlichen Artefakten* (1) wie z. B. Methoden, Sprachen und Softwarewerkzeugen zum Inhalt hat. Im Rahmen einer solchen Forschungsmethode ist sicherzustellen, dass die durchgeföhrten Arbeiten ein *relevantes Thema* (2) behandeln. Diese Relevanz ist zu belegen. Darüber hinaus ist sicherzustellen, dass die konstruierten Forschungsauftekte einen *innovativen Beitrag* (3) zur bereits in der Disziplin vorhandenen Wissensbasis darstellen, d. h., dass vergleichbare Lösungen nicht bereits im Vorfeld existieren. Die konstruierten Artefakte sind im Rahmen eines Design Science-Ansatzes zu *evaluieren* (4), um sicherzustellen, dass die gesetzten Forschungsziele tatsächlich erreicht werden. Der Charakter eines Design Science-Ansatzes ergibt sich für diesen Beitrag aus der Konstruktion des *wissenschaftlichen Artefakts* (1) in Form einer Methode zur Vermeidung von Namenskonflikten in der konzeptionellen Informationsmodellierung. Dieses Thema stellt ein *relevantes (2) Forschungsfeld* dar, wie sich aus der Konsolidierungsproblematik ergibt, die durch die verteilte Modellierung induziert wird (vgl. Abschnitt 1). Die *Innovation* (3) zeigt sich darin, dass Lösungen für die diskutierten Probleme beim Vergleich von Informationsmodellen bisher nicht vorliegen und hier entsprechende Lösungsansätze entwickelt werden (vgl. Abschnitt 3). Die *Evaluation* (4) der entwickelten Methode erfolgt anhand eines Demonstrators, der die grundätzliche Anwendbarkeit illustriert (vgl. Abschnitt 4).

3 Fachkonzept

Zur Unterstützung der formalen Spezifikation und automatisierten Durchsetzung von Bezeichnungskonventionen wird im Folgenden das Vorgehen der konventionsunterstützten verteilten Modellierung beschrieben und die notwendigen fachkonzeptionellen Spezifikationen vorgenommen.

3.1 Vorgehensmodell

Die Syntax der in einem bestimmten Modellierungskontext (z. B. Domäne, Projekt, Unternehmen) verwendeten Bezeichnungen leitet sich im Allgemeinen von zwei Komponenten ab. Zum einen findet eine Grammatik einer konkreten natürlichen Sprache Anwendung, so dass generell nur grammatisch korrekte Phrasen gebildet werden dürfen. Zum anderen wird diese natürlichsprachliche Grammatik weiter an die Belange des konkreten Anwendungskontexts angepasst, indem domänenspezifische Begriffs- und

Phrasenstrukturkonventionen in die Syntax einfließen (vgl. Abbildung 1). Bei der Bezeichnung „Prüfe eingehenden Dienstreiseantrag“ sorgt bspw. die natürlichsprachliche – in diesem Fall deutsche – Grammatik für die Kongruenz (Übereinstimmung) des Numerus, Genus und Kasus zwischen den Wortformen „prüfe“ und „Dienstreiseantrag“. Gleichzeitig tragen die domänenpezifischen Bezeichnungskonventionen dazu bei, dass nur die zugelassenen Begriffe verwendet werden (nur „Dienstreiseantrag“, nicht „Dienstreiseformular“) und dass die Struktur der Bezeichnung der erlaubten Phrasenstrukturkonvention entspricht (<Verb im Imperativ> + <Substantiv im Singular>).

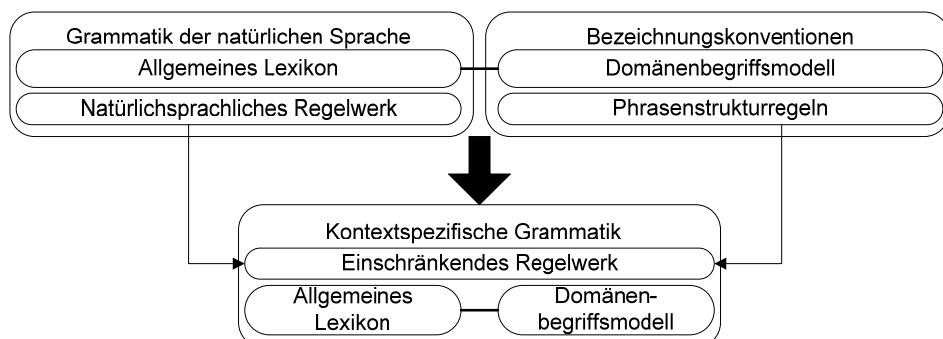


Abbildung 1: Anpassung der natürlichen Sprache durch anwendungsbezogene Bezeichnungskonventionen

Die Grammatik der für Bezeichnungen verwendeten natürlichen Sprache ist durch einen Formalismus beschrieben, der in der Regel aus einem Lexikon und einem sprachlichen Regelwerk besteht [Ca01]. Diese natürlichsprachliche Grammatik ist durch domänenpezifische Bezeichnungskonventionen, die sich wiederum aus Begriffs- und Phrasenstrukturkonventionen zusammensetzen, zu ergänzen. Begriffskonventionen werden mit Hilfe eines Domänenbegriffsmodells formalisiert, welches an das Lexikon der natürlichen Sprache anknüpft. Des Weiteren werden die gültigen Phrasenstrukturregeln mittels sogenannter Phrasentypen definiert. Somit wird die natürliche Sprache durch eine kontextspezifische Grammatik gezielt auf die im Modellierungsprojekt gültige Menge von Bezeichnungen angepasst. Dies erlaubt die spätere Überprüfbarkeit der Bezeichnungen und die Durchsetzung von Strukturkonventionen. Der fachkonzeptionelle Aufbau der Spezifikation von Bezeichnungskonventionen wird im Abschnitt 3.2 vorgestellt.

Das Domänenbegriffsmodell legt die für den konkreten Anwendungszweck erlaubten Begriffe fest und kann entweder von Grund auf neu aufgebaut bzw. aus ggf. vorhandenen Begriffsmodellen generiert werden. Im Gegensatz zu herkömmlichen Begriffsmodellen, die ausschließlich Substantive beinhalten, ist das Begriffsmodell zusätzlich auf die Wortarten der Verben und Adjektive auszudehnen. Alle übrigen Wortarten sind weitgehend domänenunabhängig, so dass deren Aufnahme in das Begriffsmodell nicht notwendig ist. Diese sind aber weiterhin im Lexikon der natürlichen Sprache enthalten. Das Domänenbegriffsmodell ist über Synonym-, Homonym- und Wortbildungsbeziehungen mit dem Lexikon und ggf. weiteren computerlinguistischen Diensten zu verknüpfen. Da solche Dienste für verschiedene natürliche Sprachen bereits vorliegen und wie z. B. Canoo [<http://canoo.net>], GermaNet [<http://www.sfs.uni-tuebingen.de/lst>] und

Wortschatz-Lexikon [<http://wortschatz.uni-leipzig.de>] teilweise Online-Zugriff bieten, ist deren technische Anbindung grundsätzlich möglich. Im Falle einer späteren Verletzung der Begriffskonventionen durch den Modellierer können somit bspw. synonyme, erlaubte Begriffe ermittelt und vorgeschlagen werden. Zum mindest das Domänenbegriffsmodell ist mit textuellen Beschreibungen der Begriffe zu hinterlegen, so dass die Bedeutung eines jeden Begriffs durch die Modellierer nachgeschlagen werden kann. Es ist darauf zu achten, dass sich ein Begriffsmodell während seines Einsatzes nicht verändern darf, um die Konsistenz bisheriger Anwendungen zu erhalten. Sollen Online-Dienste verwendet werden, ist hier bspw. die Möglichkeit einer lokalen Kopie bzw. eines versionierten Dienstes zu prüfen.

Die Spezifikation der Bezeichnungskonventionen muss für jeden Anwendungskontext (z. B. Domäne, Modellierungsprojekt, Unternehmen) einmalig durchgeführt werden, wobei ggf. bestehende Konventionen wiederverwendet werden können. Dies geschieht in Abhängigkeit von Modellierungssprachen, da für diese jeweils unterschiedliche Konventionen denkbar sind. Beispielsweise werden Funktionen einer EPK mit *Verrichtungen* bezeichnet, während Bezeichnungen von Ereignissen *Zustände* angeben [KNS92]. Für jeden Modellelementtyp sind im Folgenden die Bezeichnungskonventionen in Form von Phrasentypen anzugeben (z. B. für Funktionen in EPK <Substantiv> + <Verb im Infinitiv> sowie für Ereignisse <Substantiv> + <Hilfsverb 3. Person> + <Verb im Partizip Perfekt> [Ro96, Ku00]), wobei Alternativen je Elementtyp möglich sind. Um eine praktische Umsetzbarkeit zu gewährleisten, müssen die Phrasenstrukturen in einer solchen Form definiert werden, die sich dem Formalismus der natürlichsprachlichen Grammatik vereinbaren lässt.

Die während der Modellierung eingegebenen Bezeichnungen werden durch einen Automatismus direkt gegen die kontextspezifische Grammatik geprüft, wobei zwei Aspekte berücksichtigt werden. Zum einen ist zu untersuchen, ob die Struktur der vom Modellierer gewählten Bezeichnung den einschränkenden Regeln der zweckbezogenen Grammatik entspricht. Zum anderen ist zu prüfen, ob die gewählten Begriffe durch die Grammatik zugelassen sind. Handelt es sich bei einem Begriff um ein Substantiv, Verb oder Adjektiv – d. h. um Wortarten, die im Begriffsmodell vertreten sind – wird überprüft, ob das entsprechende Wort im Begriffsmodell erlaubt wurde. Vertreter sonstiger Wortarten werden gegen das natürlichsprachliche Lexikon geprüft. Im gesamten Prüfprozess finden computerlinguistische Methoden und Werkzeuge (u. a. Grammatikformalismen und darauf basierende Parser) Anwendung.

Sind die Begriffe und ihre Struktur gültig, so gilt die gewählte Modellelementbezeichnung als konform mit der einschränkenden kontextspezifischen Grammatik. Im Falle der Verletzung eines oder beider Aspekte werden ausgehend von der grammatischen Struktur der Bezeichnung und den vom Modellierer gewählten Wörtern alternative Phrasenstrukturen oder Begriffe angeboten, die mit den Bezeichnungskonventionen konform sind. Die Entscheidung, ob die angebotenen Phrasen als Bezeichnung ausgewählt werden, liegt beim Modellierer. Dabei kann dieser die Bedeutungen der einzelnen Begriffe aus den jeweiligen Beschreibungen der Begriffsmodelle entnehmen und daraufhin entscheiden, ob diese zu seiner intendierten Bezeichnung passen. Alternativ hat der er die

Möglichkeit, einen der vorgeschlagenen Phrasentypen als Schablone zu wählen und die darin enthaltenen Platzhalter mit erlaubten Wörtern zu füllen.

3.2 Modellierungstechnische Spezifikation

Die modellierungstechnische Spezifikation liefert die Grundlage für die Deklaration und Durchsetzung von Bezeichnungskonventionen und wird mit Hilfe von ERM in (min,max)-Notation [SS83] modelliert. Jedem *Elementtyp* einer Modellierungssprache muss mindestens eine *Phrasenstrukturkonvention* zugeordnet werden, welche die Form eines *Phrasentyps* hat. Ein Phrasentyp beschreibt den strukturellen Aufbau einer Phrase, die als Bezeichnung verwendet werden kann. Er setzt sich rekursiv zusammen aus weiteren Phrasentypen oder *Worttypen*, die atomare Elemente einer Phrase sind und als Platzhalter für die Konkretisierung durch Wörter fungieren. Ein Beispiel für einen Worttyp ist *<Substantiv im Nominativ Singular>*, ein Beispiel für einen Phrasentyp ist *<Substantiv im Nominativ Singular>+<Verb im Infinitiv>*. Die Zusammensetzung von Phrasentypen wird durch den *Phrasenstrukturkonventions-Aufbau* spezifiziert, der einem Phrasentyp entweder Subphrasentypen oder Worttypen mit ihrer *Position* im übergeordneten Phrasentyp zuordnet (vgl. Abbildung 2).

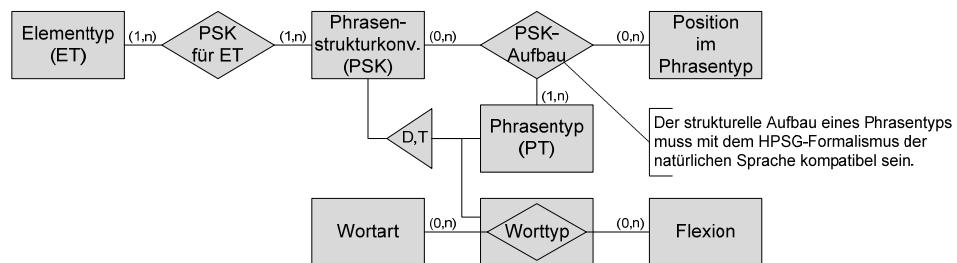


Abbildung 2: Spezifikation von Phrasenstrukturkonventionen

Ein Worttyp spezifiziert die gewünschte *Wortart* (*Substantiv*, *Verb*, *Adjektiv*, *Adverb*, *Artikel*, *Pronomen*, *Präposition*, *Konjunktion* oder *Numerale*) sowie die erwartete *Flexion*, d. h. die Beugungsform. Als Beugungsformen kommen u. a. *Kasus*, *Numerus*, *Tempus*, *Genus*, *Modus*, *Person* und *Komparation* in Frage, die i. d. R. kombiniert werden. Zu beachten ist, dass nicht jede Beugungsform auf jede Wortart anwendbar ist. Mit der in einem Phrasentyp enthaltenen sachlogischen Folge von Worttypen, deren Wortart und Flexion bestimmt ist, lassen sich beliebige Phrasenstrukturkonventionen definieren. Hierbei ist zu beachten, dass die deutsche Sprache oder auch jede andere beliebige natürliche Sprache einer Syntax unterworfen ist, die der Spezifikation von Bezeichnungskonventionen als Grundlage dient.

Deklarierte Strukturregeln sind dabei als Ergänzung der zu Grunde liegenden deutschen Sprachsyntax anzusehen und schränken demgemäß die Bezeichnungsfreiheit des Modellierers, die durch die natürliche Sprache gegeben ist, zusätzlich ein. Um den Abgleich zwischen der natürlichen Sprache und den anwendungsbezogenen Phrasenstrukturregeln zu ermöglichen, wird gefordert, dass beide Syntaxspezifikationen in kompatiblen Formalismen vorliegen (vgl. Annotation in Abbildung 2). Somit ist es möglich, direkt bei

der Spezifikation der Strukturregeln deren Konformität mit der natürlichen Sprache zu überprüfen und eventuelle Widersprüche zu signalisieren.

Zur formalen Beschreibung der Syntax natürlicher Sprachen sind aus dem Bereich der Computerlinguistik unterschiedliche Ansätze bekannt. Die Gruppe der Unifikationsgrammatiken bezeichnet eine Klasse derartiger Ansätze, welche derzeit eine weite Verbreitung und Akzeptanz erfahren [Ca01]. Diesen Ansätzen ist gemein, dass die Deklaration grammatischer Regeln stets in Bezug auf Phrasenstrukturen sowie deren Bestandteile – Phrasentypen als komplexe Einheiten oder Wörter als atomare Einheiten – geschieht. Die Formalisierung von Grammatikregeln erfolgt hierbei unter Verwendung von Merkmalsstrukturen. Eine Merkmalsstruktur repräsentiert eine Menge hierarchisch strukturierter Merkmal-Wert-Kombinationen, die die grammatischen Eigenschaften einer Phrasenstruktur sowie deren Bestandteile beschreiben. Die sich hierbei ergebenden Phrasenstrukturregeln geben an, welche grammatischen Eigenschaften und welche Position Wörter oder Phrasentypen als einzelne Bestandteile einer Phrasenstruktur einnehmen müssen, um den syntaktischen Sprachregeln zu entsprechen, denen diese Phrasenstruktur unterliegt. Exemplarisch kann eine Phrasenstruktur aus einem *Phrasentyp* bestehen, welcher sich aus zwei Worttypen zusammensetzt, die über die Merkmale *Wortart* und *Flexion* beschrieben werden (vgl. Abbildung 3). Das Merkmal *Flexion* setzt sich abhängig vom Worttyp aus weiteren Merkmalen zusammen. Für die Wortart *Verb* kann es sich um die Merkmale *Person*, *Modus* und *Tempus*, für *Substantiv* um *Numerus*, *Genus* und *Kasus* handeln. Die hierbei angegebene Folge der Worttypen markiert deren Position innerhalb der Phrasenstruktur.

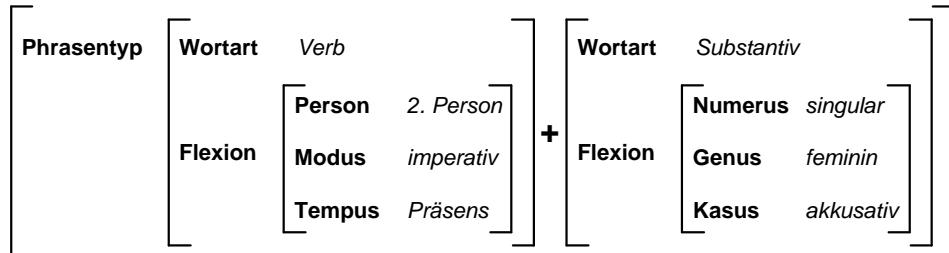


Abbildung 3: Exemplarische Darstellung einer Phrasenstruktur

Mit der *Head-driven Phrase Structure Grammar (HPSG)* [PS94] ist ein weithin etablierter Vertreter der Unifikationsgrammatiken gegeben, der bei der formalen Spezifikation von Sprachregeln dem grundlegenden Prinzip dieser Klasse von Grammatiken folgt. Hierzu bedient sich die HPSG neben einem grammatischen Regelwerk insbesondere einem Lexikon. Auf Grundlage von HPSG ist bereits eine Anzahl formaler Sprachspezifikationen verfügbar. Die formale Spezifikation der deutschen Sprachsyntax in HPSG liefert bspw. [Mu99].

Zur Sicherung der Syntaxkonformität wird diese Spezifikation bei der Definition von Phrasenstrukturkonventionen zu Grunde gelegt. Die Formalisierung definierter Konventionen erfolgt zu diesem Zweck ebenfalls in HPSG. Folglich werden die mit den Phrasenstrukturkonventionen verbundenen Grammatikregeln sowie zugehörige Lexikoneinträge mithilfe von Merkmalsstrukturen formalisiert. Die zugehörigen Lexikoneinträge

umfassen hierbei die zur Verfügung stehenden Wörter mit ihren grammatischen Eigenschaften. Die erlaubte Form eines Wortes sowie die Position, die es innerhalb einer Phrasenstruktur einnehmen darf, werden dabei durch die entsprechenden Merkmal-Wert-Kombinationen der Phrasenstrukturkonvention vorgegeben. In Bezug sowohl auf die Konformitätsprüfung im Rahmen der Spezifikation von Phrasenstrukturkonventionen als auch auf die spätere Überprüfung der bei der Modellierung angegebenen Bezeichnungen sind somit für HPSG verfügbare Parsing-Methoden wie bspw. das *Babel-System* [Mu96] anwendbar.

Unflektierte Wörter werden unabhängig von ihrer Wortart als *Lexeme* bezeichnet (bspw. das Verb „prüfen“), flektierte Wörter als *Wortform* (z. B. die 3. Person Singular „prüft“ des Lexems „prüfen“). Die bei der Modellierung verwendeten Wortformen (auf Instanzebene) sind ihren Wortarten und Flexionsformen, d. h. ihren Worttypen (auf Typebene) zuzuordnen (vgl. Abbildung 4).

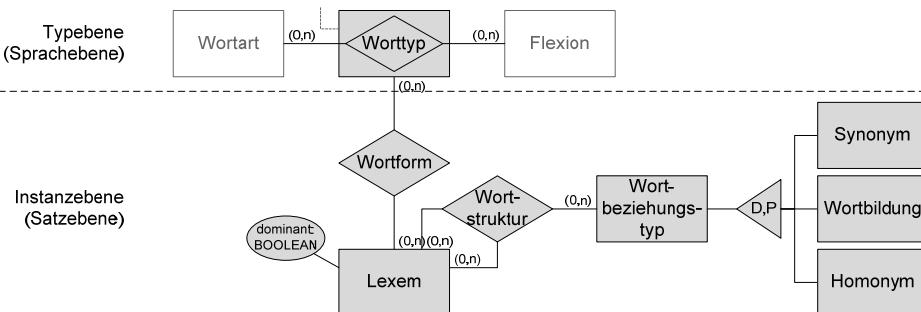


Abbildung 4: Spezifikation von Begriffskonventionen

Die Kombination aus Lexem und Worttyp ergibt damit die *Wortform*. Für die Spezifikation von Begriffsmodellen werden erlaubte Wörter in Form von Lexemen abgelegt und über diverse Beziehungstypen verknüpft. So lässt sich definieren, welche Lexeme sich zueinander *synonym* oder *homonym* verhalten bzw. durch *Wortbildung* aus anderen Lexemen gebildet werden (bspw. wird das Substantiv *Rechnungsprüfung* aus den Substantiven *Rechnung* und *Prüfung* komponiert, welche ihrerseits durch Substantivierung aus *rechnen* bzw. *prüfen* entstehen). In der sich hieraus ergebenden *Wortstruktur* werden die Begriffe der Modellierungsdomäne abgelegt und als *dominant* gekennzeichnet. Dies bewirkt, dass beim Auftreten synonymer Begriffe derjenige präferiert wird, der als dominant gekennzeichnet wurde.

Über die Beziehungstypen werden allgemein verfügbare Lexika und Dienste (vgl. Abschnitt 3.1) an das Domänenbegriffsmodell angekoppelt, um Modellierern bei Verwendung eines für die Domäne ungültigen Wortes eine gültige Alternative aus dem Domänenbegriffsmodell anbieten zu können.

4 Praktische Umsetzung

Zur Überprüfung der Umsetzbarkeit der fachkonzeptionellen Methode sowie zu Präsentationszwecken wurde eine prototypische Implementierung in Form eines Demonstrators realisiert. Der Funktionsumfang des Demonstrators orientiert sich an dem oben angeführten Vorgehensmodell.

Vor der eigentlichen Modellierung müssen zunächst das Domänenbegriffsmodell sowie das verwendete kontextspezifische Regelwerk erzeugt werden. Abbildung 5 visualisiert das Anlegen von Verben, wobei lediglich die Flexionsform des Infinitivs angegeben werden muss.

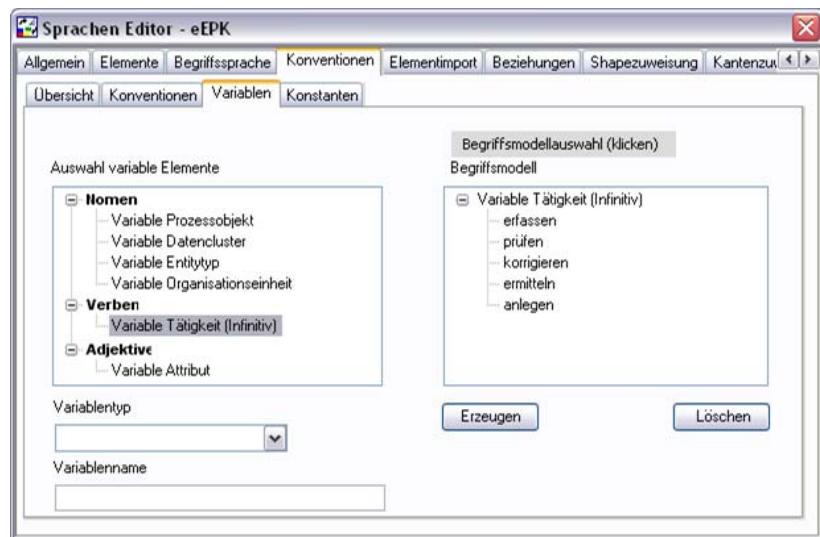


Abbildung 5: Spezifikation von Begriffskonventionen (Verben)

Für die spätere Verwendung weiterer Flexionsformen wird entweder auf externe Lexika mit hinterlegten Flexionsformen zurückgegriffen oder automatisiert eine entsprechende Form gemäß der Grammatik gebildet. Neben dem manuellen Anlegen der Begriffe ist es auch möglich, diese aus entsprechend formatierten Dateien – bspw. aus frei verfügbaren Lexika – zu importieren.

Die Oberfläche zum Anlegen von Substantiven (vgl. Abbildung 6) entspricht in ihrem Aufbau derjenigen für das Anlegen von Verben. Hierbei ist es jedoch zusätzlich möglich, durch Drag & Drop Beziehungen wie „ist Synonym für“ oder „ist ein(e)“ zwischen Begriffen zu erstellen.

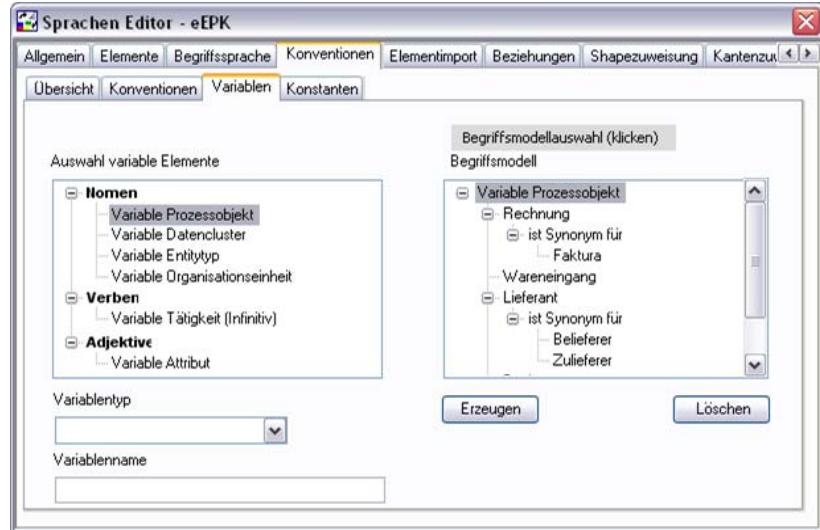


Abbildung 6: Spezifikation von Begriffskonventionen (Substantiv)

Im nächsten Schritt werden Phrasenstrukturkonventionen festgelegt und den Sprachelementen, für die sie gültig sind, zugeordnet. Abbildung 7 zeigt exemplarische Phrasenstrukturkonventionen für Ereignisse der EPK. In dem hier präsentierten Beispiel wurde für Bereitstellungereignisse die Struktur <Substantiv> + <„ist“> (+ <„nicht“>) + <Verb im Partizip Perfekt> und für Auslöseereignisse die Form <Substantiv> + <„ist“> (+ <„nicht“>) + <Verb im erweiterten Infinitiv> gewählt.

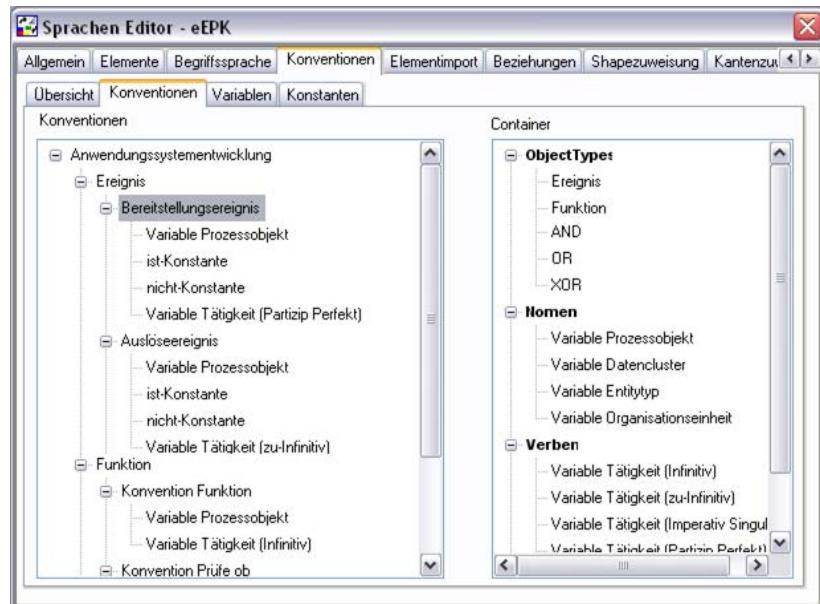


Abbildung 7: Spezifikation von Phrasenstrukturkonventionen

Ausgehend von den spezifizierten Bezeichnungskonventionen wird nun während der Modellierung geprüft, ob die vom Modellierer gewählten Bezeichnungen in ihrer Struktur und in der Begriffsauswahl den Vorgaben entsprechen (vgl. im Folgenden Abbildung 8). Eine korrekte Bezeichnungsvergabe bleibt für den Modellierer unbemerkt. Erkennt der Algorithmus jedoch eine Konventionsverletzung, wird die eingegebene Bezeichnung analysiert. Identifizierte Wörter werden ausgehend von ihrer Flexionsform in ihre Grundform überführt und ggf. durch ihr konventionstreues Synonym, das im Domänenbegriffsmodell als dominant markiert wurde, ersetzt. Ein nicht verzeichnetes Wort wird derzeit nicht akzeptiert.

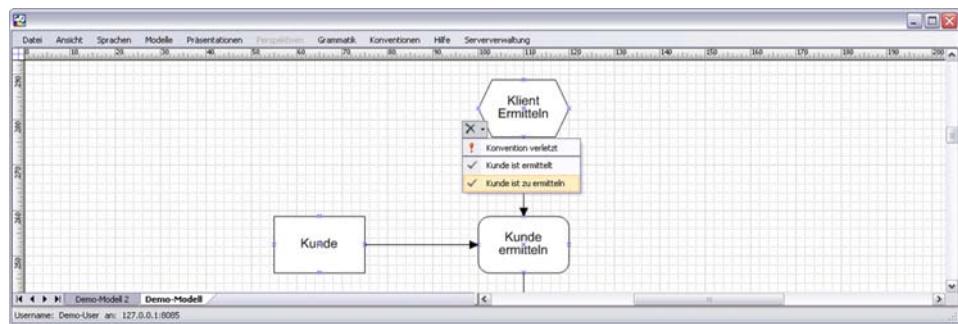


Abbildung 8: Anleitung des Modellierers zur korrekten Bezeichnung

Die Menge der Lexeme, die durch dieses Vorgehen identifiziert werden konnten, werden dem Modellierer – ggf. nach wiederholter Flexion – als Alternativen angeboten. Im Beispiel werden statt der Bezeichnung „Klient ermitteln“ die Vorschläge „Kunde ist zu ermitteln“ und „Kunde ist ermittelt“ generiert. Der Modellierer entscheidet, welche der angebotenen Alternativen er wählt bzw., ob er die angebotenen Alternativen verwirft und eine neue Bezeichnung wählt.

5 Fazit und Forschungsperspektiven

Im vorliegenden Beitrag wurde die methodische Verankerung von Bezeichnungskonventionen im Rahmen der fachkonzeptionellen Modellierung vorgestellt. Diese Methode dient als Ansatz für eine Verbesserung der Integrationsmöglichkeiten verteilt erstellter Informationsmodelle. Zwei Bestandteile ermöglichen den in Wissenschaft und Praxis bekannten und in Abschnitt 2 vorgestellten Integrationsproblemen zu begegnen. Zum einen wird durch die bereits im Vorfeld stattfindende Erstellung von Bezeichnungsrestriktionen die während des Modellierungsprojekts verfügbaren Bezeichnungsmöglichkeiten reduziert, sodass bei einer späteren Konsolidierung der TeilmODELLE bereits Informationen über den Inhalt der Modellelemente verfügbar sind. Zum anderen wird durch die stete Überprüfung der Eingaben des Modellierers sichergestellt, dass die erstellten Modelle diesen Anforderungen entsprechen. Ein erster Demonstrator belegt die Durchführbarkeit der Methode.

Grenzen der Methode liegen in dem nicht zu vernachlässigenden Aufwand, der mit der Spezifikation der notwendigen Begriffsmodelle, deren Verknüpfung mit allgemeinen Lexika sowie der Festlegung der Phrasenstrukturkonventionen einhergeht. Daher wird die Methode trotz der Wiederverwendbarkeit dieser Artefakte wohl vorwiegend Anwendung in großen bis sehr großen Modellierungsprojekten und Begriffsgemeinschaften finden.

Als zukünftiger Forschungsbedarf lässt sich zunächst die Weiterentwicklung der Methode in Hinsicht auf die Toleranz im Umgang mit fehlerhaften oder fehlenden Begriffen prognostizieren. Es wird bspw. zu klären sein, ob das Domänenbegriffsmodell zur Laufzeit erweitert werden darf oder nicht, und wenn ja, welche Konsequenzen dies für evtl. bereits bestehende Teilmodelle mit sich bringt. Der Demonstrator soll in weiteren Arbeiten verbessert werden, so dass auch die Anbindung an Online-Dienste ermöglicht werden kann. In diesem Rahmen ist auch die technisch fehlerfreie Integrationsfähigkeit bestehender Dienste zu prüfen.

Die Ergebnisse aus quantitativen empirischen Evaluationen in Forschung und Lehre werden im Sinne der Design Science ebenfalls in die Verfeinerung der Methode einfließen. Insbesondere ist zu belegen, dass die verfolgte Zielsetzung der Vereinfachung der Modellkonsolidierung im Rahmen der verteilten Modellierung auch erreicht werden kann. Weiterhin ist im Rahmen von qualitativen Untersuchungen festzustellen, ob zwei Phrasen, die strukturell wie inhaltlich identisch aufgebaut und deren Wortbedeutungen ebenfalls bekannt sind (d. h. es handelt jeweils sich um die gleiche Phrase), von verschiedenen Akteuren im Rahmen der verteilten Modellierung auch als gleichbedeutend empfunden werden.

Literaturverzeichnis

- [BL84] Batini, C., Lenzerini, M.: A Methodology for Data Schema Integration in the Entity Relationship Model, in: IEEE Transactions on Software Engineering, 10 (1984) 6, S. 650–663.
- [BLN86] Batini, C., Lenzerini, M., Navathe, S. B.: A Comparative Analysis of Methodologies for Database Schema Integration, in: ACM Computing Surveys, 18 (1986) 4, S. 323–364.
- [BKK91] Bhargava, H. K., Kimbrough, S. O., Krishnan, R.: Unique Name Violations, a Problem for Model Integration or You Say Tomato, I Say Tomahto, in: ORSA Journal on Computing, 3 (1991) 2, S. 107-120.
- [BKS08] Bögl A., Kobler M., Schrefl M.: Knowledge Acquisition from EPC Models for Extraction of Process Patterns in Engineering Domains, in: Proceedings der Multikonferenz Wirtschaftsinformatik 2008 (MKWI 2008), München 2008.
- [Br03] vom Brocke, J.: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen, Berlin 2003.
- [Ca01] Carstensen, K. U. et al.: Computerlinguistik und Sprachtheorie – Eine Einführung, Heidelberg, Berlin 2001.
- [Ch76] Chen, P. P.-S.: The Entity-Relationship Model – Toward a Unified View of Data, in: ACM Transactions on Database Systems, 1 (1976) 1, S. S. 9-36.

- [EKO07] Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models, in: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM) 2007, Ballarat 2007.
- [Fi04] Fischer, G. et al.: Meta-design: a manifesto for end-user development, in: Communications of the ACM, 47 (2004) 9, S. 33-37.
- [FSK07] Frank, U., Strecker, S., Koch, S.: „Open Model“ – ein Vorschlag für ein Forschungsprogramm der Wirtschaftsinformatik, in: A. Oberweis, C. Weinhardt, H. Gimpel, A. Koschmider, V. Pankratius, B. Schnizler (Hrsg.), eOrganisation: Service-, Prozess-, Market-Engineering, 8. Internationale Tagung Wirtschaftsinformatik, Band 2, Karlsruhe 2007, S. 217-234.
- [Gr93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications, in: Knowledge Acquisition, 5 (1993) 2, S. 199-220.
- [Gu98] Guarino, N.: Formal Ontology and Information Systems, in: N. Guarino (Hrsg.), Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, Trento, 1998, S. 3-15.
- [HS06] Hadar, I., Soffer, P.: Variations in conceptual modeling: classification and ontological analysis, in: JAIS, 7 (2006) 8, S. 568-592.
- [He04] Hevner, A. R. et al.: Design Science in Information Systems Research, in: MIS Quarterly, 28 (2004) 1, S. 75-105.
- [Ho07] Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies, in: H. Österle, J. Schelp, R. Winter (Hrsg.): Proceedings of the 15th European Conference on Information Systems (ECIS 2007), St. Gallen 2007, S. 1620-1631.
- [Ka88] Karimi, J.: Strategic Planning for Information Systems: Requirements and Information Engineering Methods, in: Journal of Management Information Systems, 4 (1988) 4, S. 5-24.
- [KNS92] Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“, in: A.-W. Scheer (Hrsg.), Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992.
- [KO05] Koschmider, A., Oberweis, A.: Ontology Based Business Process Description. In: Enterprise Modelling and Ontologies for Interoperability, Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Co-located with CAiSE'05 Conference, Porto 2005.
- [KK84] Kottemann, J. E., Konsynski, B. R.: Information Systems Planning and Development: Strategic Postures and Methodologies, in: Journal of Management Information Systems, 1 (1984) 2, S. 45-63.
- [Ku00] Kugeler, M.: Informationsmodellbasierte Organisationsgestaltung. Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation, Berlin 2000.
- [KR98] Kugeler, M., Rosemann, M.: Fachbegriffsmodellierung für betriebliche Informationssysteme und zur Unterstützung der Unternehmenskommunikation. In: Informationssystem Architekturen. Fachausschuss 5.2 der Gesellschaft für Informatik e. V. (GI), 5 (1998) 2, S. 8-15.
- [LB01] Lawrence, R., Barker, K.: Integrating Relational Database Schemas using a Standardized Dictionary, in: Proceedings of the 2001 ACM symposium on Applied computing (SAC), Las Vegas 2001.
- [Mu96] Müller, S.: The Babel-System – An HPSG-Fragment for German, a Parser and a Dialog Component, in: Proceedings of the 4th International Conference on the Practical Application of Prolog, London 1996.
- [Mu99] Müller, S.: Deutsche Syntax deklarativ – Head-Driven Phrase Structure Grammar für das Deutsche, Tübingen 1999.

- [Pa03] Palapoli, L. et al.: DIKE: a system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases, in: Software – Practice & Experience, 33 (2003) 9, New York 2003.
- [Pf08] Pfeiffer, D.: Semantic Business Process Analysis – Building Block-based Construction of Automatically Analyzable Business Process Models, Münster 2008.
- [PS00] Phalp, K., Shepperd, M.: Quantitative analysis of static models of processes, in: Journal of Systems and Software, 52 (2000) 2-3, S. 105-112.
- [PS94] Pollard, C. J., Sag, I. A.: Head Driven Phrase Structure Grammar, in: University of Chicago Press, Studies in Contemporary Linguistics, London, Chicago 1994.
- [RM05] Rizopolous, N., McBrien, P.: A General Approach to the Generation of Conceptual Model Transformations, in: Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), Porto 2005.
- [Ro96] Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden 1996.
- [RSD08] Rosemann, M.; Schwegmann, A., Delfmann, P.: Vorbereitung der Prozessmodellierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.): Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 6. Auflage, Berlin et al. 2008, S. 45-103.
- [Sa07] Sabetzadeh, M. et al.: A Relationship-Driven Framework for Model Merging, Workshop on Modeling in Software Engineering (MiSE'07) at the 29th International Conference on Software Engineering, Minneapolis 2007.
- [SS83] Schlageter G., Stucky, W.: Datenbanksysteme – Konzepte und Modelle, 2. Auflage, Stuttgart 1983.
- [VTM08] Vergidis, K., Tiwari, A., Majeed, B.: Business process analysis and optimization: beyond reengineering, in: IEEE Transactions on Systems, Man, and Cybernetics, 38 (2008) 1, 2008, S. 69-82.
- [WJ04] Wulf, V., Jarke, M.: The economics of end-user development, in: Communications of the ACM, 47 (2004) 9, S. 41-42.

Model-based Competency-Oriented Business Process Analysis

Katrina Leyking

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI GmbH)
Stuhlsatzenshausweg 3
66123 Saarbrücken
katrina.leyking@iwi.dfki.de

Ralf Angeli

IDS Scheer AG
Altenkesseler Str. 17
66115 Saarbrücken
ralf.angeli@ids-scheer.com

Abstract: Traditionally, business process analysis is primarily concerned with the identification of structural weaknesses, inefficient capacity utilization or IT deficiencies. Despite this well-founded knowledge about process performance, and despite the increasingly recognized relevance of human capital for business performance, business process analysis efforts rarely touch human resources – except for downsizing or layoffs. The objective of this paper is to extend traditional business process modelling and business process simulation by competency requirements and their impact on process performance. By doing so, business process performance can be improved through the integration of competency requirements into process design. The approach addresses the need to derive competency requirements from business processes in order to facilitate the alignment of employee competency profiles with business roles (staffing) as well as the effective closure of business-relevant competency gap through tailored learning processes (business-driven personnel development).

1 Introduction

The concept of business process analysis has been used for almost three centuries in almost all types of industries. Initially brought up by Hammer and Champy as revolutionary business process reengineering (BPR) in the 1980s [HC03], it has reached acceptance as a continuous management concept throughout industries. According to the philosophy of continuous process improvement (CPI) business processes shall be constantly analysed for potential improvements [Sc99]. Traditionally, business process analysis is primarily concerned with the identification of structural weaknesses, inefficient capacity utilization or IT deficiencies. Business processes can be streamlined by remediating or reducing bottlenecks, redundancies, or downtimes. Integrated enterprise systems such as Enterprise resource planning (ERP) systems provide a means to mitigate media or system disruptions and rationalize processes.

Business process models support process optimization efforts by graphically mapping their logical and timely flow of control and information. They also serve as a basis for simulating business process performance. Process simulation software provides the possibility to experiment with alternative business process designs and project their impact on process performance.

Despite this well-founded knowledge about process performance, and despite the increasingly recognized relevance of the human factor for business performance, business process analysis efforts rarely touch human resources – except for downsizing or layoffs. At the same time, training and learning management functions are increasingly required to justify their activities and costs in a business context. It is obvious that the key to strategic personnel development is the alignment of employee competencies with business objectives. Given the central role of business processes for enterprise strategy as well as the existing organizational aspects in business process modelling methods, the gap between business process management and learning management becomes apparent.

This paper presents an approach to synchronize learning processes with business processes. The objective is to extend traditional business process modelling with competency requirements and to use business process simulation to analyse their impact on process performance. Thus, business process performance shall be improved by integrating competency requirements into process design. Many of the common modelling methodologies are primarily concerned with the flow of activities and required resources for their execution but less concerned with knowledge and competencies as an influencing factor of process performance. The approach presented in this paper extends the traditional view on business process analysis by taking competencies and their impact on the effectiveness and efficiency of process execution into account.

The paper is structured into five chapters. After this introduction, chapter 2 sets the approach in its broader research context of process-oriented learning, as it is tackled by the research project PROLIX funded by the European Commission [PRO08]. Chapter 3 focusses upon the modelling aspects of competency-oriented business process analysis. It prepares the ground by discussing related work on business process modelling (3.1) and reasons on the selection of ARIS as core process modelling notation. Competency objects and corresponding model types are introduced (3.2), before they are integrated into business process models as required competencies (3.3). Chapter 4 builds upon those competency-enhanced business process models, which serve as a basis for process simulation. After giving the reader a short overview on the state-of-the-art of process simulation (4.1), possible use cases of competency-oriented process simulation are outlined (4.2). The process of complementing the business process model with competency-relevant performance figures (simulation configuration) is conceptualized in (4.3) and illustrated by a prototypical implementation in (4.4). Chapter five summarizes the approach and provides an outlook on future research.

2 Process-oriented Learning Lifecycle

Competency-oriented Business Process Analysis is a major component of the complete process-oriented learning management cycle (Figure 1) as it is conceptualized and implemented by the European Research project PROLIX [PRO08]. The cycle is based on the notion that learning process design must follow competency requirements identified in business processes. The underlying assumption is that only learning processes that convey those competencies to employees that are required by their employee-specific role in business processes will have a positive impact on business process performance. The cycle is structured into six major phases, starting with business process modelling and competency gap analysis, which feed into learning scenario modelling and learning process design. The execution of learning processes by the individual is followed by performance monitoring activities that evaluate the impact of learning on business processes. The cycle closes with the business analyst comparing the business outcome of the learning process (as it is monitored by the previous phase) against the initial business process need. Therefore, the business analyst visualizes and analyzes the performance results from the monitoring phase. Unless the results are satisfactory, business processes and / or learning processes are adapted and optimized according to the analysis before restarting the cycle.

In this cycle business process analysis comprises the modelling of business processes including the identification of competencies or roles required to carry out the functions of a business process and the subsequent (optional) simulation of this business process.

Integrating all phases into a closed cycle is not only a methodological challenge but also a technological one. Whereas this paper focuses rather on methods of competency-oriented business process analysis than on technical integration, the implementation of competency-oriented analysis is prototypically demonstrated in an existing business process environment, complemented by a simulation configurator.

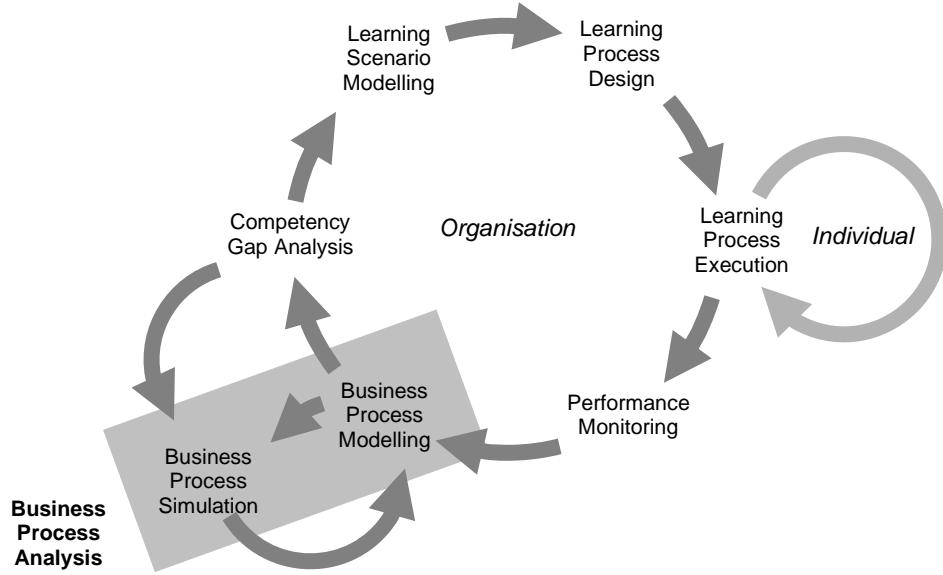


Figure 1: Process-oriented Learning Lifecycle

3 Competency-oriented Business Process Modelling

3.1 Methodological Basis

Given the widespread acceptance of existing process modelling methods, the objective was not to create a novel competency-oriented modelling method but extend an existing standard. Amongst others [PRO07A], the ARIS methodology qualified best for supporting the competency-enriched process modelling approach [Sc99].

- ARIS provides knowledge-modelling facilities, which are close to what is required for competency modelling in the context of business processes.
- The semi-formal approach of ARIS suits the requirement of easily understandable modelling for non-technicians such as HR analysts while being precise at the same time.
- ARIS models and especially its core process modelling method Event-Driven-Process-Chains (EPC) are well accepted and adopted in enterprises to the extent of a de-facto standard [Sc99].
- Extensive support for modelling organisational structures is provided by the ARIS perspectives.
- The ARIS meta-model allows for easy extension of the methodology with new models and objects through the view-oriented framework.

- EPCs can be converted to machine-interpretable process models (e.g. XPDL or BPEL).
- With the ARIS process simulator, a proven simulation tool for EPC process models is available and core simulation functionalities can be leveraged.

3.2 Competency Modelling

Competency modelling is concerned with the documentation and presentation of competencies, competency sources and competency sinks within an organisation [PRO07B]. The creation of such models is an important precondition for understanding an organisation's competency situation and needs. For competency modelling the *Competency* object type and the model types *Competency Structure Diagram* and *Competency Map* are introduced which are derived from the existing ARIS model types *Knowledge Structure Diagram* and *Knowledge Map* and integrate well with the ARIS perspective of organizational modelling consisting of modelling objects such as person types (corresponds with roles), persons, organizational units..

A competency is regarded as a skill or knowledge which can be defined independently of an individual or group. A *Competency profile* consists of a structured collection of competencies. Each Competency can occur in multiple *proficiency levels*. Thus, a competency profile varies in terms of competency levels assigned to the subordinated competencies. A competency profile instantiation specifies the levels of all subordinated competencies. In its maximum form a competency profile instantiation includes only competencies of the maximum level. The number of possible instantiations of a competency profile (in terms of competency levels) is exponential to the the number of proficiency levels and the number of competencies.

The *Competency* object type, illustrated by a light brown thought bubble, can represent both a competency and a competency profile. The *Competency* object type provides two attribute types for maintaining the proficiency levels: *Degree of coverage* and *Desired degree of coverage* that allows to model competency levels.

Competency Structure Diagram models are used to describe the structure of a competency, i.e. its composition and internal relationships. In EPC diagrams only single competencies without detail will be shown; competency structure diagrams allow their detailed structure as a hierarchy of subcompetencies to be described. The main object type used in competency structure diagrams is the *Competency* which can describe both a composite and an atomic competency. *Competency* objects can be brought into a nesting or specialising order by connecting them with connections of type *encompasses* or *is generalization of* respectively.

Figure 2 shows a competency structure diagram with nested competencies.

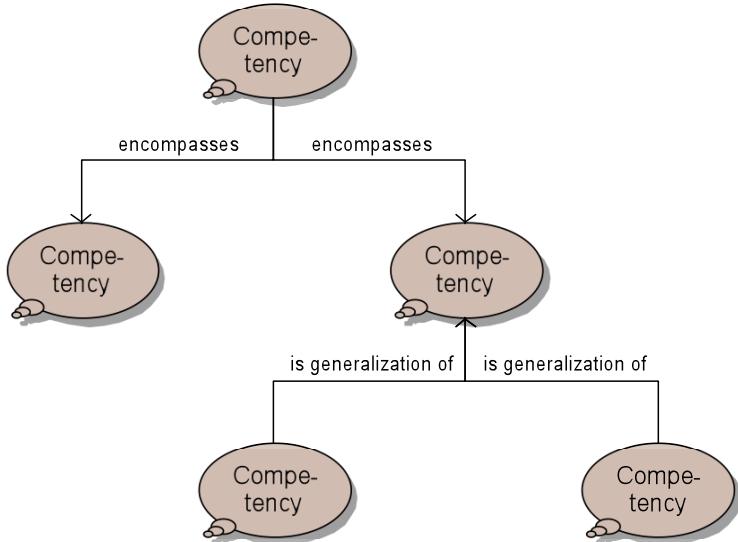


Figure 2: Competency Structure Diagram

Competency Map models are used to document where competencies are required (competency sinks) or provided (competency sources) in an organisation. They describe relations between organisational units and competencies. Consequently they contain *Competency* objects as well as the organisational unit object types. In order to distinguish the provision and demand of competencies two connection types can be used:

- *requires*: Expresses that an organisational unit needs the connected competency in order to act effectively. With this connection type it is possible to model requirement profiles for a certain role, for example.
- *disposes of*: Expresses that an organisational unit provides the connected competency.

Competency objects do not contain information about proficiency levels for a particular organisational unit. Such information is maintained in the connections between competencies and organisational units. Both the *requires* and *disposes of* connection types provide the attribute types *Degree of coverage* and *Quality of coverage* for doing so.

Figure 3 shows a *Competency Map* model with a *Person Type* object defining competency requirements for a role and a *Person* object providing competencies. The *Person Type* logically corresponds to the role concept and specifies which competencies it requires. The *object* represents a concrete employee and his competency profile.

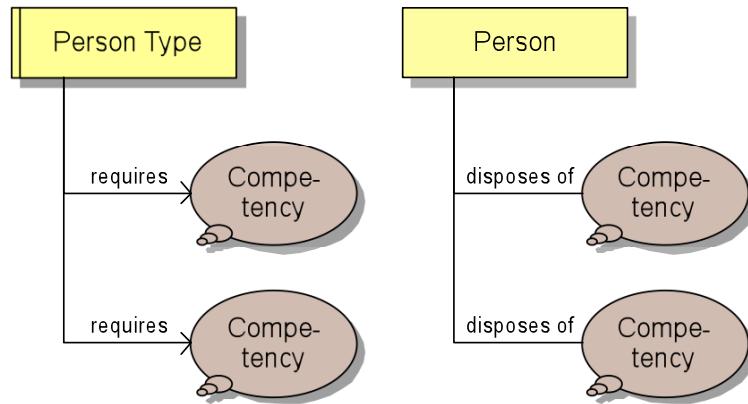


Figure 3: Competency Map

3.3 Competency-oriented Process Modelling

In the ARIS methodology, process models are located in the so-called control view which links all the other views (organisation, data, function, product/service). Process models therefore describe the flow of activities as well as required resources and data in order to produce a product or service. Competency-enriched process modelling extends this approach by taking competencies into account. Competencies are provided by organisational units as described above and they are required for the execution of activities.

The basis for competency-enriched process models are Event-driven Process Chains (EPCs). A detailed description of syntax and semantics of the control flow in EPCs can be found in Nüttgens & Rump (2002). The control flow forms the basic structure of an EPC. Information from other ARIS views are linked in by connecting objects from these views to objects of the control flow. With respect to competency-enriched process modelling human resources and competencies are the most significant ones.

Human resources, i.e. organisational units can be assigned to function objects by means of different connection types, e.g. *carries out*, *contributes to*, *decides on*, or *must be informed about*. Each of them has different semantics which also means that depending on how an organisational unit is involved in function execution, different competencies may be required.

The execution of functions by organisational units requires competencies. In order to express these relations *Competency* objects can be added to EPCs and connected with functions and organisational units. The connection between organisational unit and competency expresses the same type of relation as in a *Competency Map*. That means either a provision or a requirement of the competency by or for the organisational unit. The EPC in Figure 4 is modelled on a type level. Therefore a *Person Type* object is connected with the competency by a connection of type *requires*. If the EPC were modelled on an instance level, a *Person* object and a connection of type *disposes of* could have been used.

The connection between competency and function expresses that the attached competency is required for executing the function. The connection is of type *is required for*. Analogous to *require* and *disposes of* connection types, *is required for* connections provide the attributes *Degree of coverage* and *Quality of coverage* for specifying the required proficiency level.

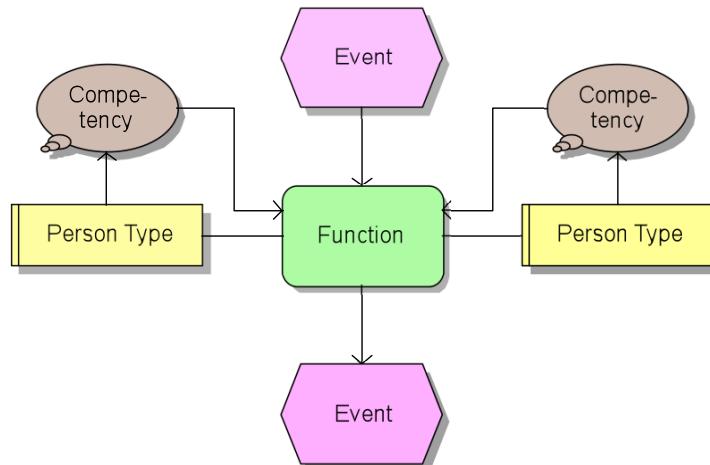


Figure 4: EPC with two Competency objects at one function

The competency analysis concept introduced in PROLIX works with pairs of artefacts and competencies. An artefact can be, among other things, a function or an organisational unit. The gap analysis, for example, can work by comparing competencies connected to two different organisational units. In PROLIX, competencies are matched automatically by a separate tool that is beyond the scope of this paper.

As the subsequent chapter will describe Simulation is another analysis step besides the competency analysis which relies on business process models as an information source. For simulation both connections are relevant since a change of the proficiency level of a competency may have an impact on properties of the function, e.g. the processing time. And the proficiency level of an organisational unit is a property of the connection between competency and organisational unit as explained before.

Finally, the option to model both connections is necessary because it enables identifying which competency, required for the execution of a function, has to be provided by which organisational unit in case more than one organisational unit is connected with the function. This is important for business performance analysis and performance monitoring.

4 Competency-oriented Business Process Simulation

4.1 Process Simulation – State-of-the-Art

Process simulation is used when the performance of to-be processes is to be analyzed. Since redesigning business processes involves changes in people, processes and technology over time, the interactions of individuals with processes and technology result in an infinite number of scenarios and outcomes that are impossible to comprehend and evaluate without the help of a computer simulation model [Tu95]. Thus, simulation helps bridging the gap between assessment of the existing and the design of the future. By introducing dynamic parameters of the process, like times, volumes, capacities and quality simulation fundamentally enhances process performance analysis. It provides a much better picture of bottlenecks, hand-over times and dynamic performance than a static analysis. [ARP99]

Process simulation is an instrument to estimate performance for a given process definition with a specific combination of parameters. It provides the possibility to test alternative solutions in different scenarios. Whereas process modelling allows for qualitative assessment, process simulation permits quantitative measurements through virtual execution of a large number of process instances. Thus, simulation is not to be mistaken for optimization as it does not provide an algorithm that calculates the optimal process solution. Moreover, one can gain insights into process characteristics by applying human intuition and experience. Playing with parameter variation helps identifying the sensitivity of the process model to performance [NRM05].

The starting points of process simulation are process models describing the sequence of events, activities and decisions leading to a certain outcome. For business process simulation to be useful, simulation-relevant attributes have to be maintained within the business process models. Such attributes are for example:

- processing times of each function
- cost of each function
- number of (human) resources involved in each function
- junction rules including the activation probability of each branch
- instantiation rules and frequency of start events.

4.2 Use Case

The performance of a process in execution not only depends upon the amount of resources but also upon the proficiency of people in competencies required for process execution. In order to improve performance of a process in execution it is therefore necessary to know what the ideal proficiency levels for the involved roles are. Such a to-be configuration can then be matched against as-is proficiency levels of people occupying the roles and makes it possible to derive training needs for the individuals. Simulation is used as a tool to find the to-be configuration.

The main use case for competency-oriented business process simulation is the refinement of competency requirements based on the estimated performance of process execution. However, competency-oriented process simulation can also be used for other purposes, e.g. staffing a process according to the competencies fitting best. Both use cases are described in the following paragraphs and serve as the basis for the requirements presented.

4.2.1 Process Simulation for Refining Competency Requirements

Based on a predefined objective (e.g. given by an objective function), several simulation experiments are run and the results are analyzed in order to improve the examined processes. Traditionally the analysis concerns the discovery of weak points, bottlenecks, unused or overworked resources, or other inefficiencies like media breaks or redundant tasks. In PROLIX the analysis also includes an evaluation of the competencies required to carry out tasks in the examined processes. Based on the analysis results the processes are improved by adapting the process structure and by changing resource assignments and capacities but also by specifying the optimal proficiency levels for competencies required and applied by roles involved in process execution.

Changing proficiency levels can affect time, cost and quality throughout process execution. In addition it may make it possible to change organizational configurations (e.g. a more skilled worker might perform additional production steps which formerly were done by additional personnel) or use different techniques requiring less process steps leading to a (further) adaptation of process structure.

4.2.2 Process Simulation for Optimal Staffing Decisions

Based on a predefined performance objective the goal is to find the best assignment of available personnel to activities in a process taking the different competencies of employees into account.

Such an assignment based on proficiency levels can help to improve performance in the short-term where it is not feasible to train the involved individuals. In order for the analysis to be carried out, the influence proficiency level changes have on factors affected by the respective competencies have to be defined and the proficiency levels of the involved employees have to be known which might not be the case due to privacy regulations.

4.3 From Competency-Oriented Business Process Modelling to Business Process Simulation

Before conducting the competency-oriented simulation a process can be analysed and improved with respect to general process inefficiencies like media breaks, redundant activities, etc. This can be done by running a conventional process simulation first. This step also helps to identify the bottlenecks the subsequent competency-driven simulation should focus upon [PRO08C].

Competency-enhanced Business Process Modelling

In a first step a business process model is created according to the competency-enriched process modelling methodology. This means modelling the control flow, the human resources involved in the process and the competency profiles required for task execution.

Competency-oriented Simulation Configuration

The simulation configuration is concerned with the determination of the performance and cost factors to be added to the process model in order to form a simulation model. For the factors depending on proficiency levels this means to find the value corresponding to a certain level.

Since the optimization of a process through simulation is an iterative approach involving the experimentation with different value configurations, it can be helpful to establish a functional relationship between proficiency levels and factor values which can also be done in the simulation configuration phase.

Competency-oriented Process Simulation

Once the business process model is configured and the simulation scenario is set up, simulation can be carried out. The results of simulation are performance and cost indicators which can be used to evaluate the simulated (competency) configuration according to the objectives defined before. In case the results are not satisfying, the configuration is changed and another simulation run is conducted. If the results are satisfying, the competency configuration is used for further steps according to the objective of the evaluation, e.g. the determination of competency gaps and training requirements of involved personnel or the selection of suitable people for certain roles.

4.4 Prototypical Implementation

The competency-enhanced business process simulation procedure is supported by the integrated interaction of three software tools in PROLIX. Business process modelling and simulation are done with the so-called Business Process Cockpit, a derivate of ARIS Business Architect and ARIS Business Simulator. For competency profile modelling and matching the Competency Analyzer, a competency and HR management system, is used. The simulation configuration is supported by the so-called Simulation Configurator which is developed from scratch for this purpose.

The configuration of the process model is a pre-processing step that includes competency-relevant performance data into the process simulation model. Since the Process Simulator processes performance attributes per function, the possible impact of competencies on the performance of a function must be translated into those performance attributes. Since this impact is likely to vary among different competency levels, ideally this translation must be done for each possible competency level. To encounter this complex challenge the simulation configuration is structured into three steps. First, scope and weights of competencies per competency profile are determined; secondly, all possibly needed performance data is specified on a functional level or by selecting a performance function; thirdly, competency levels are selected per competency profile, making up a specific competency configuration that is the basis for the first simulation.

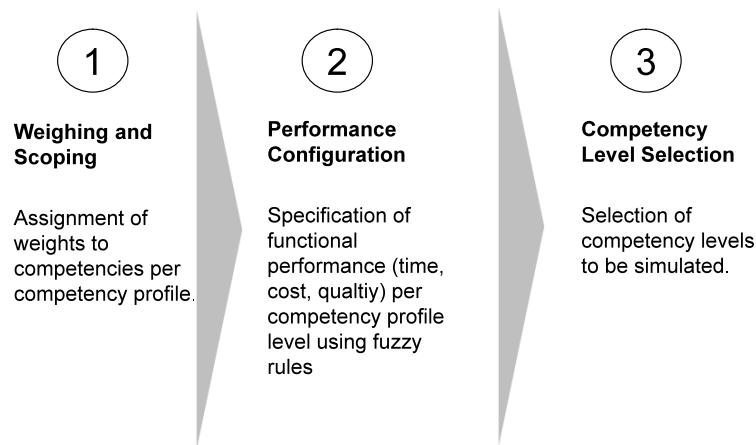


Figure 5: Three Steps of Simulation Configuration

4.4.1 Competency Weighing and Scoping

In order to reduce complexity and focus upon most critical competencies, this step allows for prioritizing competencies by weights.

Weighing and scoping comprises the following steps:

- For each competency profile derived from the imported business process model the corresponding competencies are requested from the CA and displayed in a tree hierarchy.
- For each competency profile, the HR analyst assigns weights to the competencies per competency profile (in the context of the function and the role). In particular, competencies can be weighted on a spectrum of 0-100% to determine their impact on the functional performance. Assigning 0% means to disregard this competency. Weights assigned must sum up to 100% within one competency profile. Thus, assigning 100% to one competency of a profile entails disregarding the others. The application controls that the weights are correct i.e. they sum up to 1.
- For matters of complexity reduction¹:
 - the X most important competencies can be selected automatically (the value X can be manually defined in the options menu)
 - The HR analyst can select the Y most important competencies manually by check marking.²
 - The HR analyst can predefined a specific threshold Z: The competencies whose weights are below this threshold are not further taken into account.

The weights of the reduced competency profile are recalculated automatically. The following steps consider only the reduced competency profile.

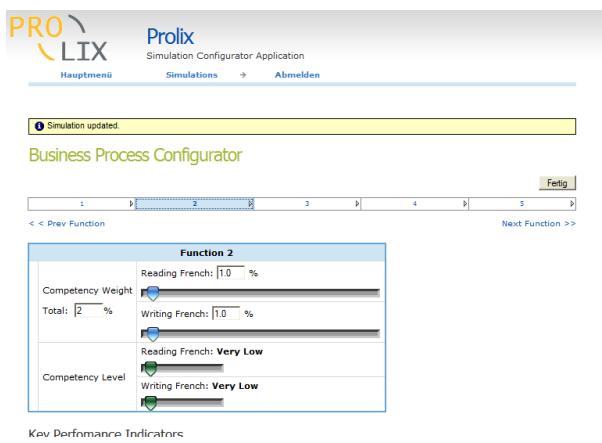


Figure 6: Screenshot of Weighing and Scoping Profiles

¹ This is necessary due to the exponential growth of the number of competency profile instantiations based on the number of competencies.

² The same effect is achieved by assigning weights of 0 to those competencies that are not significant for function performance.

4.4.2 Performance Configuration

The Performance Configuration is the core of the simulation configuration. For this translation of competency impact to functional performance, the Simulation Configurator displays for each function and for each performance dimension a dialog box. Key performance indicators include, but are not limited to cost, time, quality. In each dialog, each competency profile instantiations (set of competencies assigned to competency levels) is to be assigned to a performance figure, that is expected for the execution of the function given that the executing employee possesses competencies that correspond to this competency profile instantiation.

The challenge herein lies in the sheer amount of values that need to be specified. As outlined above, the number of possible competency profile instantiations for only one competency profile increases exponentially with the number of competencies and linearly with the number of competency levels ($|CP| = |L|^{|C|}$). Therefore, having the user enter all performance data manually appears hardly feasible. It is difficult enough to judge the possible performance impact of one competency profile instantiation. Doing so for a very large number of different instantiations may overstrain the user's cognitive ability. Therefore, various ways of how the user is supported in this task are mapped out. Figure 7 presents one possible type of support. With the help of fuzzy rules the user can specify the dependencies between competency and performance levels.

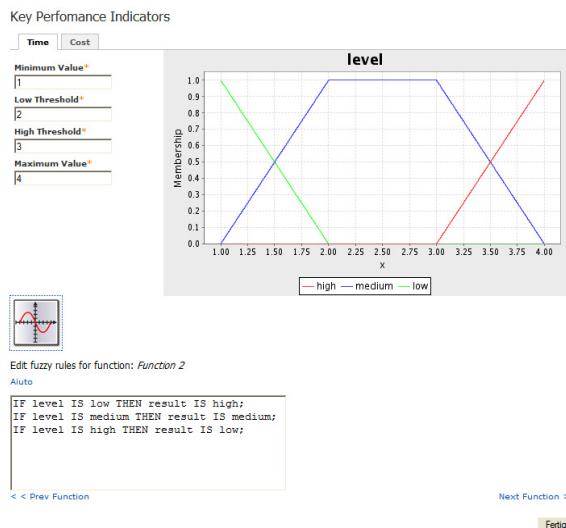


Figure 7: Screenshot of specifying performances predicted for competency profiles

4.4.3 Competency Level Selection

Once all values are specified for all possible competency profile instantiations, each of them can be simulated within the process.

- The application displays a tree of all (reduced) competency profiles and provides radio buttons to select levels for each competency.
- The user selects one target competency profile instantiation for each function that he considers most promising.

Swapping performance configuration and competency level selection would limit the tedious task of performance prediction to only those competency profile instantiations that are needed for the first simulation run. However, this approach would impede the continuing experiments with alternative competency parameters.

After the competency profiles are instantiated with competency levels to be simulated, the Simulation Configurator calculates the functional performance based on the previously specified performance values. The Simulation Configurator creates an XPDL-based business process model enhanced by simulation attributes for this specific set of competency profile instantiation (as chosen by the user). Those values are displayed per function and once accepted by the user, they are added to the XML process model as performance attributes. Those functional attributes are needed to run the simulation.³ Before the simulation is started, the application must check whether the simulation process model is complete. If it is missing performance values, the application must ask the user for these. All input needed for simulation has been gained by translating competency levels into performance-relevant data per function in the previous step. The enhanced business process model is passed to the process simulator that runs the actual simulation.

With the process simulator process models can be simulated interactively or non-interactively. In interactive runs, model objects are animated on activation, with result attributes being displayed and constantly updated for each object (cf. Figure 8). This is particularly useful for learning more about the runtime behaviour of a process. If only numerical results are required, animation can be deactivated to speed up the simulation run [IDS08].

³ The resulting competency profile instantiations have been assigned to performance figures on a functional level in the previous step, i.e. each function of the process can now be attributed with one performance figure.

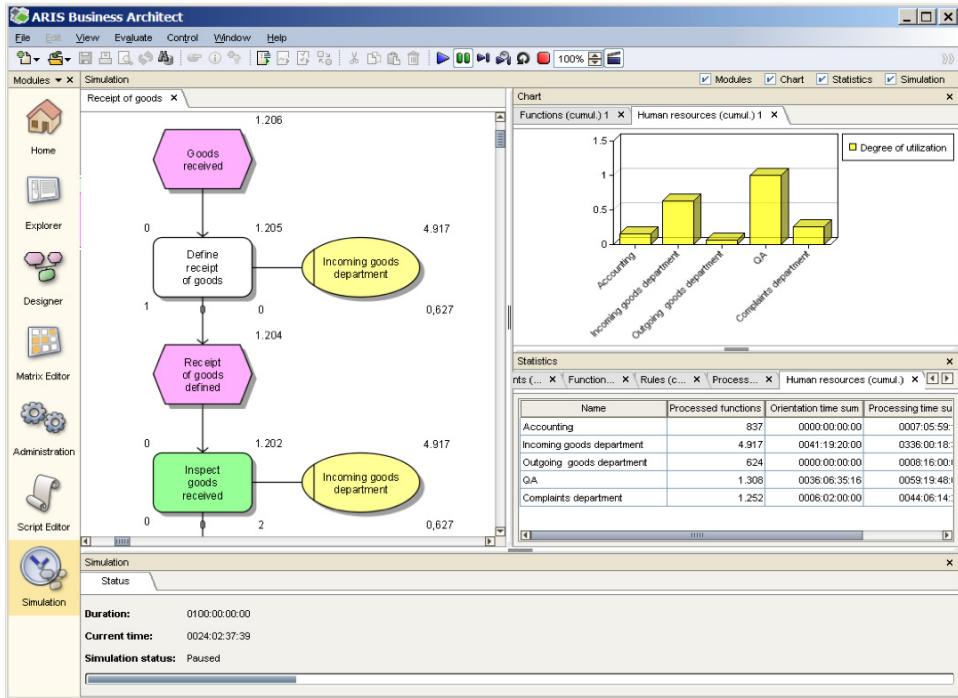


Figure 8: Standard Process Simulation

The results delivered by the process simulator for evaluation purposes are presented in the form of tables and provide a range of information, including data on the various object types and process instances. For example, function statistics provide information, such as, the number of function executions or wait times, while process statistics provide insights into the number of completed process instances or their throughput times [IDS08]. The analysis of this data makes it possible to find modelling errors, inefficiencies, and weaknesses. It also enables the user to measure preferability of a competency configuration with respect to the optimization objective.

5 Summary and Conclusion

The approach presented in this paper addresses the need to derive competency requirements from business processes in order to facilitate the alignment of employee competency profiles with business roles (staffing) as well as the effective closure of business-relevant competency gaps through tailored learning processes (business-driven personnel development). Therefore, we have introduced an extended process modelling methodology that allows for modelling competency requirements in business process models. In order to use this competency-oriented business process model for simulation purposes it needs to be enriched with performance projections based on alternative competency profiles instantiations. The simulation configurator supports the HR analyst providing this kind of performance data and generates a business process simulation model for a specific set of competency profile instantiations. The process simulator runs the simulation and delivers process performance results based on which the HR analyst decides to keep experimenting with them in the Simulation Configurator or is satisfied and fixes the competency requirement specification for further staffing and training purposes.

Ideally, the performance data entered manually for each possible combination of competency profile instantiations and function would be gathered in the course of multiple iterations of the PROLIX learning lifecycle. The PROLIX performance monitoring component would be a possible source for this kind of data. Beyond the project, any kind of business activity monitoring tool could gather performance data for different competency distributions.⁴ A machine-learning approach may then be able to derive rules that indicate how performance factors change with varying competency profiles. Thus, the HR analyst would have a solid rule base at hand instead of relying on his gut feeling. Thus, simulation configuration could be almost automated. All what remains to be done by the HR analyst is to play with different competency profile instantiations. Eventually, having rules for all possible combinations derived, an optimization algorithm could be set up to identify those competency profile instantiations per role that achieve optimal business process performance.

⁴ The ARIS platform offers a Process Performance Manager (PPM) that is specifically designed to gather performance data along a business process across various transactional business systems (e.g. ERP systems, Office software).

The concept of simulation configuration is not limited to integrate the impact of competencies into business process simulation. An open research question is whether it could be applied to any other additional factors that are considered to influence process performance by any means. For example, service selection in a service-oriented architecture often depends on their quality-of-service and economic constraints. The identification of the optimal service may be supported by a similar kind of extended business process simulation. It would allow to simulate alternative services within a business processes before making the decision. Tackling this issue will be subject to our future analysis and research.

References

- [Ab02] Abecker, A.: Geschäftsprozessorientiertes Wissensmanagement. Springer-Verlag, Berlin, Heidelberg, 2002.
- [ARP99] Aguilar, M.; Rautert, T.; Pater, A.: Business Process Simulation: A Fundamental Step Supporting Process Centered Management. In (Farrington, P.A.; Nembhard, H.B.; Sturrock, D.T.; and Evans, G.W. Eds.): Proceedings of the 1999 Winter Simulation Conference, 1999, p. 1383-1392.
- [BFS87] Bratley, P.; Fox, B.; Schrage, L.: A Guide to Simulation. Second Edition. New York, Berlin, Heidelberg, 1987.
- [GMU04] Gronau, N.; Müller, C.; Uslar, M.: The KMDL Knowledge Management Approach: Integrating Knowledge Conversations and Business Process Modeling. Berlin, Heidelberg, 2004.
- [IDS08] IDS Scheer 2008, ARIS Business Simulator, http://www.ids-scheer.com/set/6473/ARIS%20Business%20Simulator%20FS%20en_2008-02.pdf, retrieved on 2008-04-22.
- [HC03] Hammer, M.; Champy, J.: Reengineering the Corporation - A Manifesto for Business Revolution. Harper Business Essentials, New York, 2003.
- [He05] Heisig, P.: Integration von Wissensmanagement in Geschäftsprozesse. Technische Universität Berlin, Berlin, 2005.
- [KNS92] Keller, G; Nüttgens, M; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In (Scheer, A.-W. Ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 89, 1992.
- [NRM05] Neumann, S.; Rosemann, M.; Schwiegmann, A. (2005) Simulation von Geschäftsprozessen. In (Becker, J.; Kugler, M.; Rosemann, M. Eds.), Geschäftsprozessmanagement, Springer-Verlag, Berlin, Heidelberg, p. 434-453.
- [NR02] Nüttgens, M.; Rump, J. F.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In (Desel, J.; Weske, M. Eds.): 2002, Promise 2002 – Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens (Potsdam, October 2002), LNI Vol. P-21, Bonn, 2002, pp. 64-77.
- [PRO08] <http://www.prolix-project.eu>, Access: 09.10.2008
- [PRO07A] PROLIX Deliverable D1.2, Extended Modelling Notations, http://www.prolixproject.org/index.php?option=com_repository&Itemid=45&func=fileinfo&id=88, Access: 09.10.2008
- [PRO07B] PROLIX Deliverable D3.1, Extended Methodological Framework for Competency Evidence Elicitation, http://www.prolixproject.org/index.php?option=com_repository&Itemid=45&func=fileinfo&id=103, Access: 09.10.2008

- [PRO07C] PROLIX Deliverable D6.3, Refocused Simulation concept based on the PROLIX conceptual framework, http://www.prolixproject.org/index.php?option=com_repository&Itemid=45&func=fileinfo&id=99
- [Sc99] Scheer, August-Wilhelm: ARIS - Business Process Frameworks. Springer-Verlag, Berlin, Heidelberg, 1999.
- [Tu95] Tumay, K. (1995) Business Process Simulation. In: (Alexopoulos, C; Kang, K; Lilegdon, W.R.; Goldsman, D. Eds.): Proceedings of the 1995 Winter Simulation Conferences, pp. 55-60.

On the Choice Between Graph-Based and Block-Structured Business Process Modeling Languages

Oliver Kopp Daniel Martin Daniel Wutke Frank Leymann

Institute of Architecture of Application Systems

Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany

{kopp,martin,wutke,leymann}@iaas.uni-stuttgart.de

Abstract: The most prominent business process notations in use today are BPMN, EPC and BPEL. While all those languages show similarities on the conceptual level and share similar constructs, the semantics of these constructs and even the intended use of the language itself are often quite different. As a result, users are uncertain when to use which language or construct in a particular language, especially when they have used another business process notation before. In this paper, we discuss the core characteristics of graph-based and block-structured modeling languages and compare them with respect to their join and loop semantics.

1 Introduction

Workflow technology is a central aspect of Business Process Management (BPM) and an important technology in both industry and academia. Workflows are instances of workflow models, which are representations of real-world business processes [LR00, Wes07]. Basically, a workflow model consists of activities and the ordering amongst them. Workflow models can serve different purposes: on the one hand, they can be employed for documentation of business processes itself, e.g. for facilitating business process modeling by business analysts; on the other hand, workflow models defined by IT experts can serve as input for Workflow Management Systems (WfMS) that allow their machine-aided execution. The problem of facilitating the creation of executable business process models based on abstract business process descriptions, e.g. through enhancing them with enough information to facilitate their automated execution, is known as the *Business-IT gap* [DvdAtH05]. A number of workflow languages exists for the specification and the graphical representation of processes. One important aspect is the control flow, which specifies the execution order of the activities. Conceptually, workflow languages can be classified according to whether their control flow modeling style is centered around the notion of *blocks* or the notion of *graphs*. In block-structured languages, control flow is defined similar to existing programming languages by using block-structures such as *if* or *while*. In contrast, process control flow in graph-oriented workflow languages is defined through explicit *control links* between activities.

The intended use of a workflow language places a number of requirements and restrictions on the kind of language employed; whether used primarily for documentation purposes

(*abstract processes*) or whether it is used to provide a detailed process model that can be deployed on a WfMS for automatic execution (*executable processes*) highly depends on the process to be modeled and the intended use of the resulting model. Moreover, certain languages even allow for modeling abstract processes as well as executable processes. As a result, guidelines have to be provided to process modelers to allow them choosing the “right” language for their purpose.

In this paper, a number of workflow languages are compared with respect to their intended use, their notation and serialization, their basic modeling approach (block-structured vs. graph-oriented vs. hybrid), their supported structure of loops (structured loops vs. arbitrary cycles) and their support for expressing explicit data flow. Block-structured and graph-oriented workflow languages differ in their representation of loops, splits and joins. We see these aspects as the main distinction between these languages. Therefore, this paper focuses on the comparison of loops, splits and joins.

The compared workflow languages comprise *Event-driven Process Chains (EPC, [STA05, KNS92])* and the *Business Process Modeling Notation (BPMN, [Obj08])* on the side of languages targeted primarily on modeling processes for documentation purposes and the *Web Service Business Process Execution Language (BPEL, [Org07])* and the *Windows Workflow Foundation (WF, [Mic08])* as languages for modeling (also) executable processes.

1.1 Related Work

In this paper, we compare modeling languages with respect to their support of block-structured and graph-based modeling constructs. Other approaches to compare modeling languages are based on patterns. Currently, there exist control flow patterns [vdAtHKB03, Kie03], process instantiation patterns [DM08], correlation patterns [BDDW07], data handling patterns [RtHEvdA05], exception handling patterns [RvdAtH06] and service interaction patterns [BDtH05]. Workflow patterns focus on the expressiveness of the control flow constructs and do not explicitly distinguish between graph-based and block-structured modeling. The other patterns do not focus on the control flow, but on the capability of the language to specify process instantiation, process instance correlation and the handling of data, exceptions and interactions with other services, which are not captured in this work.

Different intentions of different process languages have been addressed in the context of BPMN and BPEL in [RM06] and [Pal06]. They address the intention of these modeling languages, but do not focus on the constructs to model control flow. The suitability of BPMN for business process modeling is investigated in [WvdAD⁺06]. However, BPMN is not compared to other languages in this work.

Besides the presented languages, there are several other graph-based and block-structured languages and formalisms. A prominent formal graph-based language is the Petri-net based workflow nets [vdA98]. Pi-calculus is block-based, since it offers a parallel and a split construct. However, due to its capabilities to generate channels, it can also be used to capture the semantics of graph-based languages [PW05]. An overview of all formalisms used on the workflow area is presented in [vBK06].

Transformation strategies between block-structured and graph-based languages as well as their limitations are presented in [MLZ08]. In general, all graph-based models can be mapped to block-structured models and vice versa. Typically, a mapping from a model A to a model B and mapping the model B back results in a different model A'. The main reason is that there are different strategies for the mapping and that arbitrary cycles are not supported by block-structured languages and thus have to be “emulated” by constructs offered by the block-structured language.

1.2 Structure of the Paper

The paper is organized as follows: in Section 2 we present a business process example, which is modeled using both, graph-based and block-structured languages. In graph-based languages, there are different rules of how to join control flows during execution. In Section 3 we present an overview of the problem and the current solutions. An overview of techniques to model loops is given in Section 4. Afterwards, we present a comparison of the workflow languages in Section 5. Finally, we provide a conclusion in Section 6.

2 Exemplary Process

In this section, we present a process that shows distinct features which we will discuss in the sections to follow. The process itself serves as a running example, being re-modeled in BPEL, EPC and BPMN to exemplify the use of graph-based and block-structured modeling approaches.

2.1 Graph-Based Modeling using BPEL <flow>

The process we use is a modified version of the “Loan approval” example process from [Org07], modeled using the graph-based constructs provided by BPEL. This graph-based part of BPEL originates from BPEL’s predecessor WSFL [Ley01]. WSFL is based on the Flow Definition Language (FDL), formalized in [LR00] as PM-Graphs.

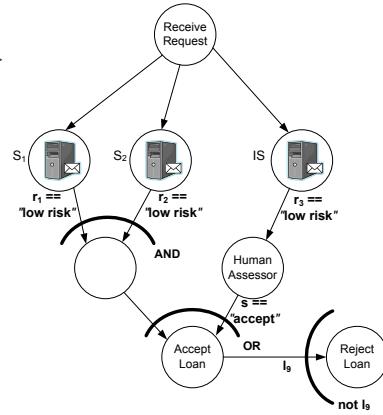
BPEL allows to define a workflow model using nodes and edges inside a `<flow>` element. Nodes are activities and edges are called “links”. The logic of decisions and branching is solely expressed through transition conditions and join conditions. Transition conditions and join conditions are both Boolean expressions. As soon as an activity is completed, the transition conditions on their outgoing links are evaluated. The result is set as the “status of the link”, which is `true` or `false`. Afterwards, the target of each link is visited. If the status of all incoming links is defined, the join condition of the activity is evaluated. If the join condition evaluates to `false`, the activity is called “dead” and the status of all its outgoing links is set to `false`. If the join condition evaluates to `true`, the activity is executed and the status of each outgoing link is evaluated. Regardless of the activity being

```

<process>
  <flow>
    <links>
      <link name="receive-to-S1">
        <link name="S1-to-AND" />
      ...
    </links>
    <receive name="ReceiveRequest">...</receive>
    <invoke ...>...</invoke>
    <empty name="AND">
      <targets>
        <joinCondition>
          $S1-to-AND AND $S2-to-AND
        </joinCondition>
        <target name="S1-to-AND" />
        <target name="S2-to-AND" />
      </targets>
      <sources>
        <source name="AND-to-OR" />
      </sources>
    </empty>
    ...
  </flow>
</process>

```

(a) BPEL code



(b) Graphical representation

Figure 1: Loan approval process in BPEL

executed, the target of each link is visited. The propagation of the dead status via `false` as link status is called dead-path elimination (DPE). DPE is conceptually detailed in [LR00], specified for BPEL in [Org07] and explained in detail in [CKLW03].

The process is presented in Figure 1: Figure 1(a) presents extracts of the BPEL code and 1(b) the graphical representation of the process. The process is initiated by the reception of a loan request at activity *receive request*. This loan request is checked in parallel by two *external* credit rating services (activities S_1 and S_2) and a company *internal* rating service IS . If the company internal rating service reports “low risk”, the subsequent activity is a risk assessment by a human assessor that manually checks the request, otherwise this step is skipped. In our case, we want to take conservative decisions, i.e. the loan request must only be accepted if either both external rating services report *low risk* or both the internal rating service and the subsequent human assessor report *low risk*. Of course we also accept the loan if all services report *low risk*—both external services and the internal rating service and assessor. The loan is to be rejected in any other case.

We implement these requirements using transition conditions on the links following each of the rating services which evaluate to `false` if anything else than “*low risk*” is reported. In case “*high risk*” occurs, the state of the link evaluates to `false`. The AND-Join following the two external rating services means that the join condition is a conjunction over the state of the links leaving from S_1 and S_2 , thus the link going from the AND-Join to the OR-Join evaluates to `true` only if both external rating services returned “*low risk*” and therefore implements the first part of our requirements. Similarly, the result of the OR-Join is only `true` if one or both of its incoming links are `true`. Again, this is only the case if either

both external or the internal assessment have returned “low risk”. The only part left from the requirements is to reject the loan request if it cannot be accepted. This is modeled by link l_9 , which is annotated with the default transition condition `true`. The join condition on activity “reject” is a negation of the link status of link l_9 (*not* l_9). In that way, “Reject Loan” is only executed iff “Accept Loan” is not executed: l_9 is set to `true` if “Accept Loan” is executed. Thus, *not* l_9 evaluates to `false` and “Reject Loan” is not executed. If “Accept Loan” is not executed, the status of l_9 is set to `false` and *not* l_9 evaluates to `true`, leading to the execution of “Reject Loan”.

2.2 Graph-Based Modeling using BPMN and EPC

In this section, we present the “Loan approval” process introduced above, modeled using BPMN and EPC as examples of a graph-oriented modeling language. The resulting BPMN graph (Figure 2(a)) looks considerably different compared to the BPEL model presented in Section 2.1. This is mainly due to the different way of modeling joins: through arbitrary Boolean expressions in the BPEL case, or through additional explicit join constructs such as AND, OR, XOR in BPMN. A complex join was chosen instead of a literal translation of the process using BPMN AND-Joins and OR-Joins: the “Reject Loan” activity that has to be executed only if “Accept Loan” was not executed. This behavior cannot be modeled without being able to refer to the state of a control flow link in the join condition. In BPMN, the solution for that is to use a complex gateway that refers to variables containing the state of each of the assessment services, updated by each of the services after their completion. These variables are then used to decide which outgoing sequence flow is to follow, e.g. either reject or accept the loan request.

Since EPCs do not provide support for arbitrary join conditions, the paths of all decisions have to be merged using an OR-Join. Afterwards, the decision whether to accept or reject is taken at the subsequent function “Take Final Decision” (Figure 2(b)). The concrete semantics has to be specified using additional text, which may be included in the diagram.

For the same reason, decisions in BPMN have to be modeled explicitly: i.e. in the BPEL graph model, we relied on dead-path elimination to skip the “human assessor” activity if the internal assessment returned “high risk”. In BPMN, this has to be modeled explicitly using a dedicated sequence flow. To summarize, the main difference between the BPEL graph-model and the BPMN model is the way how conditions are modeled: as a combination of expressions on control flow links and join conditions in the case of BPEL; or as a complex gateway in the case of BPMN. In the BPEL case, decision logic is distributed among links and activities, whereas it is represented in compact form as a complex gateway in BPMN.

2.3 Block-Structured Modeling using BPEL

Besides BPEL, the Windows Workflow Foundation (WF, [Mic08]) and “normal” programming languages support block-structured modeling. For better readability, we use a

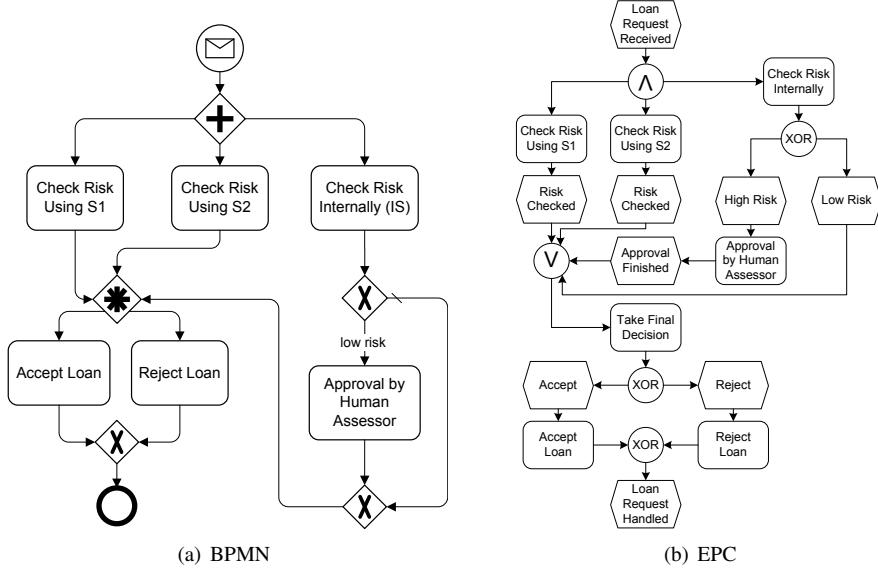


Figure 2: Loan approval modeled using BPMN and EPC

simplified version of the BPEL syntax in Figure 3. We use the names of the activities as function names and abstract from their XML syntax by representing their XML attributes by function parameters. Note that our way of representing the block-structured part of BPEL emphasizes on the similarity of block structured modeling languages with regular, procedural programming languages such as C. At the expense of a different representation using programming concepts such as variables, function calls and nested block structures, this kind of modeling however provides clear semantics to every modeler familiar with basic computer programming languages. Furthermore, since the representation already is in a form similar to a “real” program, transformation into executable code typically is easier to achieve [Ecl08].

Since WS-BPEL is essentially a hybrid language that was derived from a block-structured ancestor XLANG [Tha01] and a graph-oriented ancestor WSFL [Ley01], it allows users to freely choose between both approaches. It is even possible to mix both concepts, by allowing graphs to be freely drawn within the `<flow>` element. This element may in turn may be used as a block element nested within other blocks. However, the BPEL `<flow>` can also used as a block structure only to allow for parallelism; each element it contains is executed in parallel. Using the BPEL `<flow>` as a block structure simply means not using control flow links within the block, so that each decision is represented using explicit branch or loop constructs such as `<if>` or `<while>`.

On the other hand, the way a business process typically is drawn comes very close to graph form, with nodes as activities and directed edges as control flow dependencies between them. As shown in scientific literature, it is hard to assign clear and distinct semantics to these languages (e.g. [WEvdAtH05, Men07, Weh07]) mainly due to the ambiguous way

```

sequence {
    receive(C, loan_request);
    flow {
        flow {
            extRes1 = invoke(S1, loan_request);
            extRes2 = invoke(s2, loan_request);
        }
        sequence {
            intRes = invoke(IS, loan_request);
            if (intRes=='OK') {
                intRes = invoke(assessor, loan_request);
            }
        }
    }
    if ((extRes1=='OK' & extRes2=='OK') || (intRes=='OK')) {
        invoke(CS, accept_loan);
    } else {
        invoke(CS, reject_loan);
    }
}

```

Figure 3: Loan approval modeled in block-structured BPEL

to interpret loops as well as the joins and splits they are constructed of. Sometimes a specific language even explicitly refrains from defining clear semantics (e.g. BPMN). Thus, transformation of graph-based workflow descriptions into executable form generally can be considered harder to achieve.

3 Join Conditions

As mentioned before, the way how control flow joins are implemented in a workflow modeling language heavily influences how the semantics of a certain process are expressed in the model. This section therefore revisits the examples from Section 2 and highlights different join semantics of each approach.

3.1 Kind of Join Conditions

Generally, two main types of control flow joins can be distinguished in todays workflow languages:

Restricted Choice Languages such as EPC [STA05, KNS92] and YAWL [vdAtH05] only allow to join different threads of control flow using a restricted set of operators, typically in the form of AND, OR and XOR elements as part of their modeling language. An important property of these languages is that it is not possible to refer to negative link state, i.e. modeling a situation as depicted in Figure 1 is not possible; a modeler has to workaround this issue, possibly creating a much more complex model. If the set of join types in a language allowing only restricted choice joins is functionally complete, a Boolean expression representing a complex join condition can be constructed using combinations of

multiple join operators.

Arbitrary Expressions Languages allowing to define arbitrary Boolean expressions over the state of incoming links belong to this category. BPEL however is the only candidate that allows expressions over link state only, while BPMN allows to refer to process state (in form of process variables) in its join expressions. This has a noteworthy consequence: since it is very common to refer to process state as part of a join condition, complex join logic in BPEL has to be split among transition conditions of incoming links where process variable access is allowed, and the join condition as a Boolean expression over the state of all incoming links (and therefore the result of each of the transition conditions). In contrast to “join condition fragmentation” as in BPEL, other languages allow to model complex join conditions as one single, “compact” statement since process variable access is allowed.

BPMN is a hybrid in this case: While it offers a restricted choice (AND, OR, XOR and complex gateway), the “complex gateway” allows for defining arbitrary expressions. Naturally, restricted choice join operators are mostly used in languages whose primary intend is human-human communication of a certain process. In this case, a join refers to the availability of control flow only, in contrast to human-machine (i.e. executable) languages where joins need to be expressed in a very specific manner referring to process state and control flow and thus must be modeled in the form of a Boolean expression.

3.2 Complexity of Join Evaluation

The complexity of join evaluation has already been discussed extensively in literature. Especially, the semantics of the OR-Join in EPCs have raised many discussions and lead to different proposals for concrete executable semantics. An extensive presentation and comparison of the proposed semantics can be found in [Men07, MvdA07, vdAtH05, Weh07]. The inherent problem of the OR-Join generally is that it is hard to decide how long it should block the control flow. It is especially hard in processes containing cycles [Kin06]. Most discussions debate whether this should be resolved through local knowledge, i.e. by introducing additional arcs in the model or “negative control tokens” (as it is done by dead-path elimination) that make it possible to unblock and evaluate the join condition when it is clear that no more tokens can arrive. On the other hand, execution engines have been proposed that decide—by looking at the global state of the process—if a join can be unblocked since no more tokens will arrive on the input arcs. Naturally, these “global semantics” of join nodes introduce a significantly higher complexity of the evaluation of a join [MvdA07, DGHW07]. Languages that depend on such “global semantics” for join evaluation are BPMN and EPC.

“Local join” semantics that the execution relies on dead-path elimination (see Section 2.1) or on the introduction of additional arcs to tell the join node that no control flow will arrive on a certain path. BPEL realizes local join semantics by dead-path elimination. In that way, no additional arcs are introduced. Additional arcs are problematic when it comes to auditing the deployed process: the model deployed differs from the model finally executed by the engine.

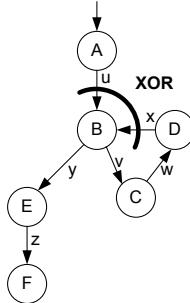


Figure 4: Example for a graph-oriented loop, modeled without an explicit loop construct through links between activities

4 Loops

A loop refers to a set of activities that are executed either while a certain loop condition holds or until a certain exit condition is reached. Two forms of loops can be found in common workflow languages: *block-structured* and *graph-based* loops. Block-structured loops, such as the *while* or *repeat until* loop, are characterized by an explicit loop construct and an exit condition at either the top or the bottom of the construct. From the process definition languages analyzed in the paper, BPEL, BPMN and WF provide support for structured loop constructs. In BPEL and WF, exit conditions can be specified to be evaluated either at the top of the loop through the *While* activity or at the bottom through the *RepeatUntil* activity. BPMN distinguishes *RepeatUntil* and *While* loops by attributes of the looping activity.

In contrast to block-structured languages, loops are modeled in graph-based languages without a dedicated loop construct by defining control flow links between activities. Typically, these links are associated with so-called *transition conditions* that define under which condition the corresponding link is to be followed by the navigator of the workflow management system. While the absence of the necessity of an explicit loop construct conceptually allows for the definition of arbitrary loops with multiple incoming and outgoing control links, such patterns are characterized by a number of problems. For instance, consider the example of a loop represented through control links between activities presented in Figure 4. In this example, link u denotes the loop entry and link y denotes the loop exit. Node B denotes the activity that evaluates the exit condition of the loop which repeatedly triggers execution of the loop activities C and D until the exit condition of the loop is reached and the loop is exited through link y . The process presented in Figure 4 can only be executed under certain assumptions. While e.g. BPEL allows for graph-based definition of process control flow within the BPEL `<flow>` construct, it does not allow for definition of graphs containing cycles; which restricts BPEL to block-structured loops. This is due to the dead-path elimination algorithm employed by BPEL [Org07]. This algorithm essentially demands each join operation—in case of the example activity B which joins the incoming links u (the loop entry) and x (the final link of the loop body)—to be synchronizing, i.e. to block execution of the join activity until the link status of each incoming link has been propagated to the join activity and hence the value of the join condition can be evaluated. As a result, execution of

the loop can never be started, since the start of the loop depends on a defined link status of u which can only be produced after evaluation of activity B . Other approaches [MvdA07] have solved the aforementioned problem for EPCs by introducing a non-synchronizing (XOR) join construct and an extended form of dead/wait status propagation.

The example presented in Figure 4 also shows a second problem related to dead-path elimination in cyclic graphs in combination with arbitrary split behavior [LR00]. Node B not only links to node C (which is inside the loop) but also to the loop external node E (which in turn links to node F) through the loop exit link y . Assume that B is an OR-Split. In this case, after B is executed, it is possible that both its outgoing links are activated; as a result activities C and E are executed. Assume that execution of activity C takes more time than execution of E . After successful execution of E , its outgoing link z is activated and activity F is executed. Given OR-Split semantics in B , the already executed path E, F would be executed again, which might or might not be desired by the modeler. This ambiguity can be solved by restricting exit nodes on a cycle to XOR semantics, meaning that either the loop is exited (through one of potentially a number of exit conditions) or the loop is continued with the next cycle/iteration.

Of the analyzed languages BPMN and EPCs allow definition of arbitrary cycles.

5 Comparison

Criteria	BPEL	BPMN	EPC	WF
Intention	human-machine ^{C01}	human-human	human-human	human-machine
Standardized Rendering	-	+	+	-
Standardized Serialization	+	+ ^{C02}	-	- ^{C03}
Graph Modeling	+	+	+	- ^{C04}
Well-Formed Only	- ^{C05}	-	-	+
Block Modeling	+	+ ^{C06}	-	+
Structured Loops	+	+	+	+
Arbitrary Cycles	- ^{C07}	+	+	+ ^{C08}
Parameterized Split	+ ^{C09}	+	+ ^{C10}	-
Parameterized Join	+ ^{C11}	+ ^{C12}	-	-
Join semantics	local (DPE)	various	various	synchronization
Explicit Data Flow	- ^{C13}	+	+	-

Table 1: Summarized comparison of BPEL, BPMN, EPC and WF.

In Table 1, a summary of the comparison of the workflow languages BPEL, BPMN, EPC and WF is presented with respect to their intention, standardized rendering and serialization, modeling paradigm, supported loops, splits, joins and whether they support explicit data flow. Many of the decisions are commented later in this section and referenced by Cxx,

with xx standing for the number of the comment. The criteria are explained in the following. *Intention* expresses whether the respective language has been designed primarily for human-human or human-machine communication. While languages classified as human-human are used mostly for business process documentation purposes, languages classified as human-machine are used for automatic execution of business processes. As such, they require a clearly defined execution semantics that gives precise and unambiguous instructions on how a process must be executed. Note that while BPEL's abstract process profiles also facilitate its use as a modeling language, it has been classified as human-machine, since its primary focus is on executable processes (C01). *Standardized rendering* and *standardized serialization* refer to whether the language standard defines a graphical notation or a machine-processable textual representation, respectively. Note that XPDL [Wor08] is the proposed standard serialization format for BPMN diagrams (C02). The WF is a proprietary language and thus does not provide a standardized serialization; process models are directly translated to executable code (C03). Apart from WF, all compared languages support *graph-oriented modeling* of process control flow with a restriction to acyclic graphs in BPEL due to the reasons outlined in Section 4. WF is restricted to purely block-structured modeling (C04). Languages that only allow well-formed process models restrict consecutive split and join operations to the same type are referred to as *well-formed* [vdA98]. Well-formed means for example that if control flow is split using a XOR-Split it must be joined through a XOR-Join; joining a XOR-Split with an AND-Join is disallowed. In BPEL arbitrary Boolean join conditions on the status of incoming links can be specified (including in particular those resulting in non well-formed process models, C05), BPMN and EPC themselves do not define any restrictions on the types of consecutive split/join pairs. *Block-structured modeling* constructs are supported by BPEL and to a limited extent also by BPMN: BPMN supports a *while* construct and sub-processes as the only block-structured constructs (C06). EPCs do not allow for block-structured modeling. All compares languages allow for structured loops; while BPMN allows for modeling loops both as graphs and through blocks, modeling structured loops in BPEL is limited to blocks (due to the aforementioned required acyclicity of graphs in BPEL, C07). For similar reasons BPEL does not allow modeling of *arbitrary cycles* (see Section 4); as a result loops have to be modeled using blocks. In WF arbitrary cycles can be realized using state machine-based modeling (C08). *Parameterized split* refers to the ability to specify the link status individually for each of potentially multiple outgoing links of an activity. In BPEL this can be achieved through different transition conditions on the individual links (where an exclusive split needs mutually exclusive transition conditions, C09). EPCs are restricted to AND-Splits, OR-Splits and XOR-Splits (C10). The same restrictions hold for EPCs with respect to their support of *parameterized join* operations, i.e. the ability of defining a join condition (see Section 3). Join conditions in BPEL are restricted to Boolean expressions over the status of incoming links of the join activity (C11); BPMN allows for defining join conditions also on process instance data (C12). Note that this functionality of BPMN can be emulated in BPEL by defining appropriate transition conditions on the incoming links themselves. BPEL, as an executable process language, has a precisely defined *join semantics* while BPMN and EPC as languages focused primarily on process modeling do not. However, a number of execution semantics (including join semantics in particular) have been proposed for BPMN and EPC to fill this gap ([MvdA07, Weh07]). In order to be more generic, we use the

semantics described in the respective specification of the language for comparison, not the various proposed executable interpretations or restrictions. WF only provides a block construct for parallel execution which completes its execution once each enclosed activity is completed. All compared languages express activity ordering through modeling process control flow. BPMN and EPC offer associations with data objects and thus allow to specify *explicit data flow*. In [KL06], *BPEL-D* has been proposed as an extension of BPEL that allows defining explicit data flow (C13).

6 Conclusion

In the paper we presented a comparison of four common languages for modeling business processes—BPEL, BPMN, EPC, and WF—with different fields of application and different modeling approaches. We specifically showed that BPEL supports both, block-structured and graph-based modeling. The implications of graph-based and block-structured modeling have been discussed by providing examples that highlight the languages’ key characteristics. Special attention has been paid to discussing problems related to joining multiple execution paths and loops. A summary of the comparison was given in Table 1. Based on these results, we are going to provide scenarios to guide process modelers to choose a process description language fulfilling their individual requirements. We specifically focused on graph-oriented and block-structured modeling based on a simple example. However the chosen example cannot cover the full problem at the appropriate level of detail. We therefore propose a survey of the suitability of the different modeling approaches for disparate user groups (business analysts, IT architects and others).

Acknowledgments This work is supported by the BMBF funded project Tools4BPEL (01ISE08B) and the EU funded project TripCom (FP6-027324).

References

- [BDDW07] A. P. Barros, G. Decker, M. Dumas, and F. Weber. Correlation Patterns in Service-Oriented Architectures. In *Proceedings of the 9th International Conference on Fundamental Approaches to Software Engineering (FASE)*, LNCS, pages 245–259, 2007.
- [BDtH05] A. Barros, M. Dumas, and A. H. M. ter Hofstede. Service Interaction Patterns. In *Proceedings of the 3rd International Conference on Business Process Management*, LNCS, pages 302–318, 2005.
- [CKLW03] Francisco Curbera, Rania Khalaf, Frank Leymann, and Sanjiva Weerawarana. Exception Handling in the BPEL4WS Language. In *International Conference on Business Process Management*, volume 2678 of *LNCS*, pages 276–290, 2003.
- [DGHW07] Marlon Dumas, Alexander Grosskopf, Thomas Hettel, and Moe Thandar Wynn. Semantics of Standard Process Models with OR-Joins. In *Proceedings 15th International Conference on Business Process Management (BPM)*, LNCS, pages 103–117, 2007.

national Conference on Coopartitive Information Systems (CoopIS), volume 4803 of *LNCS*, pages 41–58, 2007.

- [DM08] G. Decker and J. Mendling. Instantiation Semantics for Process Models. In *Proceedings of the 6th International Conference on Business Process Management (BPM)*, LNCS, pages 164–179, 2008.
- [DvdAtH05] M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede. *Process Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley-Interscience, 2005.
- [Ecl08] Eclipse Foundation. BPEL to Java (B2J) Subproject, 2008. <http://www.eclipse.org/stp/b2j/>.
- [Kie03] B. Kiepuszewski. *Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows*. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2003.
- [Kin06] E. Kindler. On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. *Data and Knowledge Engineering*, 56(1):23–40, 2006.
- [KL06] Rania Khalaf and Frank Leymann. Role-based Decomposition of Business Processes using BPEL. In *Proceedings of the IEEE International Conference on Web Services (ICWS '06)*, pages 770–780. IEEE Computer Society, 2006.
- [KNS92] G. Keller, M. Nüttgens, and A.-W. Scheer. Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik, 1992.
- [Ley01] F. Leymann. Web Services Flow Language (WSFL 1.0), 2001. IBM Software Group.
- [LR00] F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, 2000.
- [Men07] J. Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration, 2007.
- [Mic08] Microsoft. Windows Workflow Foundation, 2008. <http://www.microsoft.com/net/WFDetails.aspx>.
- [MLZ08] J. Mendling, K. B. Lassen, and U. Zdun. On the Transformation of Control Flow between Block-Oriented and Graph-Oriented Process Modeling Languages. *International Journal of Business Process Integration and Management (IJBPIM)*, 3(2), September 2008.
- [MvdA07] J. Mendling and W. M. P. van der Aalst. Formalization and Verification of EPCs with OR-Joins Based on State and Context. In *Proceedings of the the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007)*, volume 4495 of *LNCS*, pages 439–453, 2007.
- [Obj08] Object Management Group. *Business Process Modeling Notation, V1.1*, 2008. <http://www.omg.org/spec/BPMN/1.1/PDF>.
- [Org07] Organization for the Advancement of Structured Information Standards (OASIS). *Web Services Business Process Execution Language Version 2.0 – OASIS Standard*, 2007. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>.

- [Pal06] N. Palmer. Understanding the BPMN-XPDL-BPEL Value Chain. *Business Integration Journal*, November/December 2006.
- [PW05] F. Puhlmann and M. Weske. Using the pi-Calculus for Formalizing Workflow Patterns. In *Proceedings of the 4th International Conference on Business Process Management (BPM 2006)*, volume 4102 of *LNCS*, pages 414–419, 2005.
- [RM06] J. Recker and J. Mendling. On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In *CAiSE 2006 Workshop Proceedings – Eleventh International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2006)*, 2006.
- [RtHEvdA05] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst. Workflow Data Patterns: Identification, Representation and Tool Support. In *24th International Conference on Conceptual Modeling (ER 2005)*, volume 3716 of *LNCS*, 2005.
- [RvdAtH06] N. Russell, W. M. P. van der Aalst, and A. H. M. ter Hofstede. Workflow Exception Patterns. In *Advanced Information Systems Engineering (AISE)*, volume 4001 of *LNCS*, pages 288–302, 2006.
- [STA05] A.-W. Scheer, O. Thomas, and O. Adam. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Process Modeling Using Event-Driven Process Chains. Wiley-Interscience, 2005.
- [Tha01] S. Thatte. *XLANG Web Services for Business Process Design*. Microsoft Corporation, 2001.
- [vBK06] F. van Breugel and M. Koshkina. Models and Verification of BPEL. <http://www.cse.yorku.ca/~franck/research/drafts/tutorial.pdf>, 2006.
- [vdA98] W. M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8:21–66, 1998.
- [vdAtH05] W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
- [vdAtHKB03] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.
- [Weh07] J. Wehler. Boolean and free-choice semantics of Event-driven Process Chains. In *Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK 2007)*, pages 77–96, 2007.
- [Wes07] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin, 2007.
- [WEvdAtH05] M. T. Wynn, D. Edmond, W. M. P. van der Aalst, and A. H. M. ter Hofstede. Achieving a General, Formal and Decidable Approach to the OR-Join in Workflow Using Reset Nets. In *Applications and Theory of Petri Nets*, volume 3526 of *LNCS*, pages 423–443, 2005.
- [Wor08] Workflow Management Coalition. *XML Process Definition Language Version 2.1*, 2008. <http://www.wfmc.org/xpdl-developers-center.html>.
- [WvdAD⁺06] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, and N. Russell. On the Suitability of BPMN for Business Process Modelling. In *Fourth International Conference on Business Process Management (BPM)*, volume 4102 of *LNCS*, pages 161–176, 2006.

All links were followed on 10/13/08.

3D Representation of Business Process Models

Stefanie Betz, Daniel Eichhorn, Susan Hickl, Stefan Klink,
Agnes Koschmider, Yu Li, Andreas Oberweis, Ralf Trunko,

Institute of Applied Informatics and Formal Description Methods (AIFB)

Universität Karlsruhe (TH), Germany

{betz|eichhorn|hickl|klink|koschmider|li|oberweis|trunko}@aifb.uni-karlsruhe.de

Abstract: 3D technologies open up new possibilities for modeling business processes. They provide higher plasticity and eliminate some deficits of conventional 2D process modeling such as the limitation of the amount of information to be integrated into a process model in an understandable way. The aim of this paper is to show how the usage of an additional visual modeling dimension may support users in compactly representing and animating business process models. For this purpose, we propose an approach for 3D representation of business process models based on Petri nets. The need for the third modeling dimension is pointed out with three modeling scenarios for which we propose modeling improvements in 3D space. Early evaluations indicate the effectiveness of our approach, which goes beyond conventional modeling tools for business processes.

1 Introduction

The increasing interest in business process management by academia and industry has resulted in a multitude of modeling languages and tools supporting business process modeling. Modelers are frequently confronted with new modeling languages and sophisticated tools which may overwhelm especially those users inexperienced in process modeling. Most existing tool implementations disregard the fact that users need more support for process modeling than just a repository of graphical symbols [KHG08]. In this context, it is an essential requirement to provide a user-friendly business process visualization which facilitates a quick understanding and overview of a process model. Nevertheless, current process modeling languages are limited in their visualization capabilities because the amount of information to be integrated into a process model is usually much more than can be effectively displayed. To reduce the size of the language elements in order to integrate more information into a process model does not really solve the problem. In contrast, this approach does not disburden the life of process modelers (concerning learnability of the modeling language) or process stakeholders (concerning understandability of the process model). Thus, some approaches (e.g. [JaBu96, Sch99, MeS06, BRB07]) propose for instance different perspectives on a process model in order to improve its comprehensibility.

A deficit of current process modeling languages is that they use only two modeling dimensions (x-axis and y-axis) for visualization. This limits the amount of information to

be integrated into a process model in an understandable way. By introducing the third dimension into business process modeling, we are able to represent information of a process model more compactly than current process modeling languages allow. Additionally, 3D process model visualization allows the user to change her view-point. In this paper, we extend the conventional “flat” representation of Petri nets with a concept for spatial visualization of net diagrams. In order to improve the understandability of Petri nets for novices, expressive icons may be used for places and transitions. For an easier understandable visualization of the behavior of a Petri net, model animations using graphical elements of the related application domain might be used [Ver00]. With our approach, we intend not only to improve the layout of process models (e.g. by minimizing the number of crossings of arcs [Rei93, Röl07]) but also to increase the information content of a Petri net model. Despite the prejudice that 3D displays are prevalently regarded as “gimmicks” (nice layout, no additional value) [BES00], we are confident to generate an added value with our approach – last but not least by considering the three-dimensional grasp of humans. The integration of the third modeling dimension supports a more clearly arranged representation of modeling concepts and simulation results of Petri nets. This approach gains benefits for visualizing more compactly relevant information of the process models and allows integrating new objects for user interactivity.

The remainder of the paper is structured as follows. In Section 2, we present related work in the field of 3D visualization of business process models. Section 3 surveys some deficiencies of 2D process modeling with an example. Within Section 4, we describe 3D modeling of business processes including the modeling of roles, objects and hierarchies as well as relationships between business processes. In Section 5, we depict an initial implementation of our approach. The paper concludes with a summary and a critical consideration of the results obtained so far and gives an outlook on future research.

2 Related Work

Existing work in 3D visualization of business process models can be differentiated in three categories: (1) Business process modeling, (2) 3D modeling, and (3) Layout methods.

Several methods have been proposed to model business processes with complex objects and respective events [DGB07, LHG07, LeO03], but with limited capabilities in their visualization. These approaches lack features for effectively describing all relevant information integrated in one process model. Instead, large number of dependencies between activities and objects need to be hidden because the amount of information to be integrated into a process model is usually much more than can be effectively displayed. For instance, regarding a Petri net model (or an EPC model respectively) the user can specify resources and objects, which are required for handling specific activities, on one process level, but she needs to switch to another view in order to gain insight into the dependencies between resources.

The relocation of relevant information on different views is a popular technique for breaking down the complexity of business process models [JaBu96, Sch99, MeS06,

BRB07]. However, the switching between different views is a time-consuming task, because with every switching the user needs to spend time on orienting herself in the new view and comprehend the relationships between resources, roles and activities. Some tools use the third dimension for business process modeling [KGM99, BES00, KiP04, Röl07] in order to support the integration of new objects and to allow user interactivity. Our approach, which uses three dimensions for a compact representation of dependencies between objects, organization, and process models, builds on this existing research work.

Layout methods for the design of diagrams have been a research focus for several years [MaM07, HaL08]. The current tool support for diagram layout is still evolving. Especially, in 3D modeling one deficit for layout methods are efficient algorithms for handling a limited number of elements in all three dimensions. Therefore, we will use the algorithm proposed in [KaK89] in order to draw graphs, which has been evaluated as an effective technique in [Pur98].

3 Running Example

We will demonstrate benefits of 3D business process modeling by presenting a simple example. Figure 1 shows a conventional 2D representation of a decision process for flood crisis management. Precisely, the depicted process is part of a complex process of controlling rehabilitation measures [Pla02, NeK05].

This process is modeled with Petri nets where circles represent conditions and rectangles represent actions. A transition box inscribed with two vertical bars indicates that the respective transition is refined by another Petri net. Repeated refinement of transitions leads to a hierarchical representation of a process.

To solve the decision problems concerning an efficient resources and action allocation, the involvement of several organizational units (e.g. executive staff, administration member) is required. The first part of the process handles the priority of the disaster notifications, which are analyzed in the second part of the process. Before selecting an action scenario concerning the selected notifications, the availability of resources needs to be checked (e.g. manpower, technical and electrical equipment) where some specific activities are performed by roles (e.g. executive staff as *role 1* or PR and media staff as *role 2* in Figure 1). For comprehensibility reasons we annotated only few roles. Finally, a decision will be made and the responsible persons and involved parties will be informed.

Although this standard process model is on a high level of abstraction with a clearly arranged number of roles and activities, the understandability is not satisfying. Additionally, to gain insight into the relationship of roles and process hierarchies users need to open another process or organization model, which requires spending time in order to understand the relationships. In this context, the next section presents our approach of a compact representation of relevant information between a business process model and an organization model.

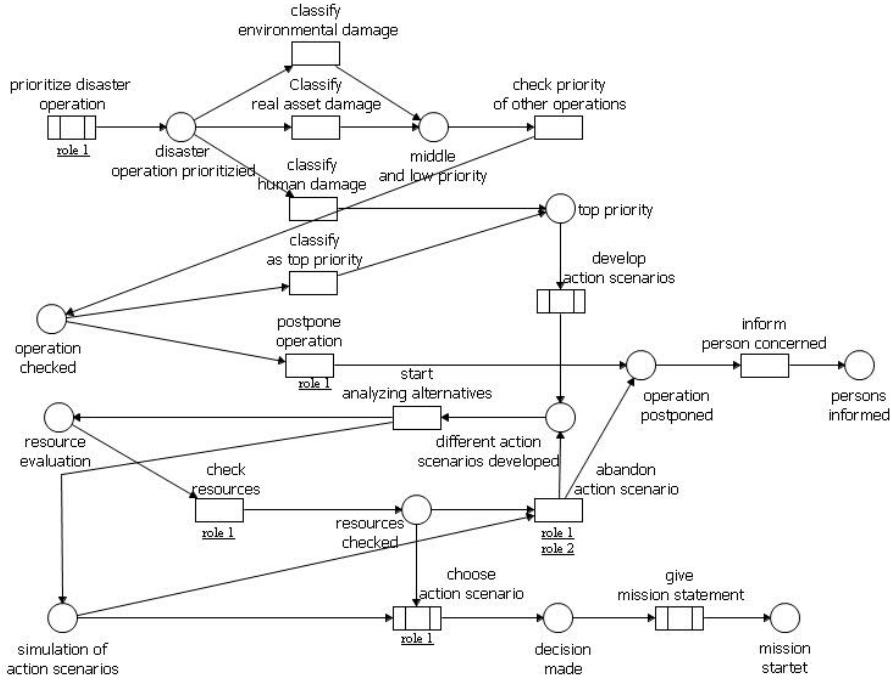


Figure 1: Decision process for flood crisis management

4 Modeling Business Processes with 3 Dimensions

In this section we first introduce Petri net constructs, which are used in the following in the three scenarios for 3D business process modeling with respect to modeling roles, resources, semantic relationships and hierarchies of business processes. Naturally, other processes or data resulting from process metrics (e.g. time, costs) can also be modeled as 3D objects. But in the initial stage of exploring 3D business process modeling, we restrict ourselves in this paper on the three scenarios mentioned before that will be explained in Section 4.2 to 4.4.

4.1 Petri nets

Petri nets are a widely accepted graphical language for the specification, simulation and verification of information systems' behavior. Formally, a Petri net is a directed bipartite graph with two sets of nodes (places and transitions) and a set of arcs which can be described by the triple $N = (P, T, F)$, where P is the set of places, T is the set of transitions (which is disjoint from P) and $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation. Elements of P are graphically represented as circles (e.g. *disaster operation prioritized* in Figure 1), elements of T as rectangles (e.g. *prioritize disaster operation*) and elements of F as directed arcs between places and transitions. A place p is an input place of a transition t , if

there exists an arc from p to t . A place p is an output place of a transition t , if there exists an arc from t to p . The set of all input places of a transition t is denoted by $\bullet t$ and is called pre-set. The set of all output places is denoted by $t\bullet$ and is called post-set. Numerous Petri net variants have been proposed, which can be subsumed in elementary or high-level Petri nets. In elementary Petri nets, places contain tokens (black dots), which represent anonymous objects, whereas the flow of the tokens simulates the process flow. In high-level Petri nets tokens represent identifiable objects.

In the following we will describe how users may benefit from the usage of the third dimension for modeling business processes. Our approach is applicable for both elementary and high-level Petri nets.

4.2 Modeling Roles and Objects

Usually, activities and roles are stored in different models (process model and organization model) and users manually insert a link between roles and activities. Roles may be assigned to activities in the process model but an organization model is still required to describe the relationships between organization units. Thus, the user needs to switch to another view in order to gain a deeper insight into roles or resources. Activities represent tasks, roles describe capabilities of a resource, which is an entity performing the assigned tasks in the business process.

With respect to the running example, the deficit of switching between different models is overcome by using a third dimension for representing relationships between process and organization model. Thus, users can easily catch the position of a specific role, e.g. always at the beginning/in the middle/at the end of a process or if the role appears in several parts of the process. Additionally, our modeling technique supports the generation of specific views on interesting details of relationships between activities and roles respectively resources. This is not directly possible in the business process model shown in Figure 1.

For this, we define the assignment of a role to an activity by a mapping $mapToRoles: A \rightarrow P(R)$ where $R = \{r_1, \dots, r_m\}$ is the set of all roles, m is the number of roles, A is the set of activities and $P(R)$ is the power set of R . This mapping can be visualized as shown in Figure 2, where the task *abandon action scenario* is performed by two roles ("execution staff" and "PR and media staff"), which are modeled in a different dimension than the process elements.

Additionally, the usage of the third dimension supports the aggregation or the generalization of roles, respectively. In Figure 2 the role *Execution staff* and the role *PR and media staff* are aggregated to the role *Civil servants*. The aggregation and generalization of roles are differentiated by the type of arc. In a generalization relationship the arcs are directed. For aggregation we use *simple* lines. The benefit of this representation is that users do not need to switch to another view in order to orient themselves.

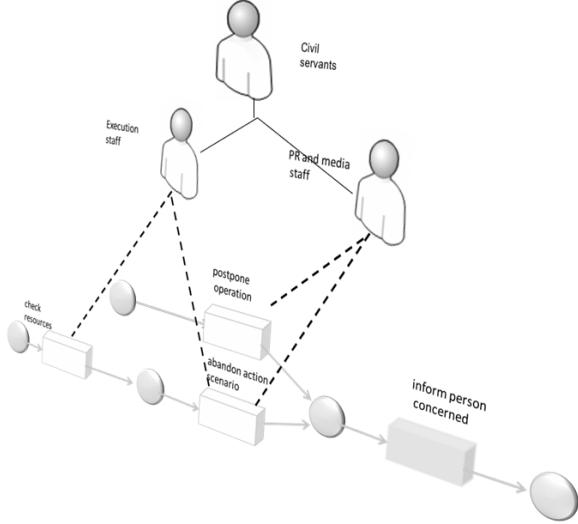


Figure 2: Using the third dimension for representing the relationship between roles and activities

On business process instance level, activity instances require to be associated with resources which can be used to perform the activities. Consequently, we define a mapping from roles to capable and available resources $\text{mapRoleToResource} : R \rightarrow P(\text{RES})$, where $\text{RES} = \{res_1, \dots, res_n\}$ is the set of all resources, n is the number of different resources and $P(\text{RES})$ is the powerset of RES . The visualization of the combination of the mappings mapToRoles and mapRoleToResource is shown in Figure 3, where, for instance, Anne and Gina have the capability of the role *Execution staff*.

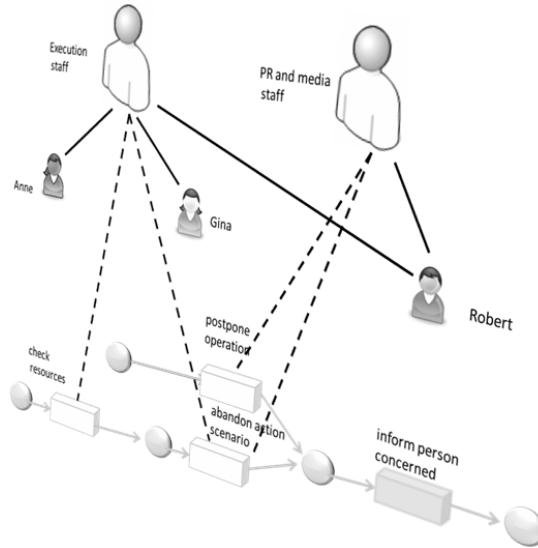


Figure 3: Representing the relationships between roles, resources and activities

The two mappings support the generation of specific views of interesting activities, roles or resources. Figure 3 shows a restricted view on a) roles performing specific activities, b) involved resources of specific roles and c) activities which require specific roles/resources. The benefit of this restriction is that users are able to see how frequently specific roles are involved in the process (i.e. by counting the number of relationships between specific roles and activities). With respect to the given example, this representation overcomes the limitation of switching between process and organizational models in order to achieve this information.

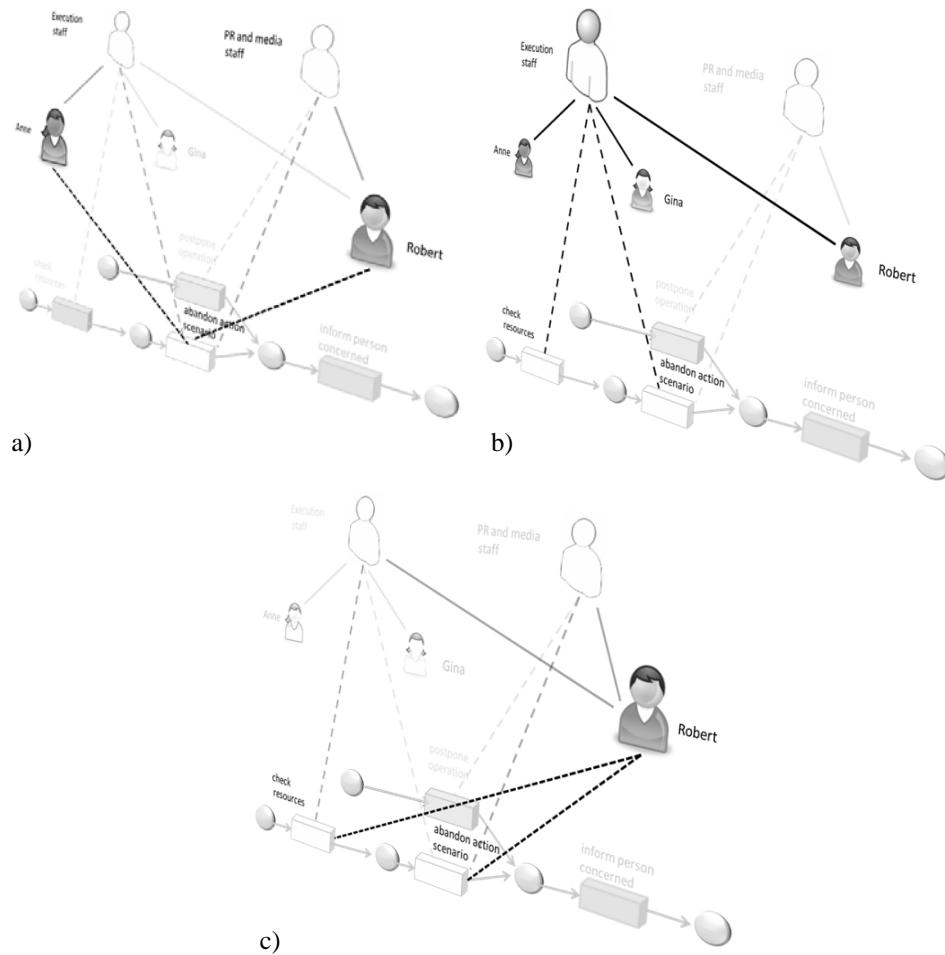


Figure 4: Using the third dimension for a view on a) roles, b) resources, and c) activities

4.3 Modeling Semantic Relationships between Business Processes

Besides 3D visualization of one single business process as described above, we will focus in this section on relationships between several business processes. Relationships

between two or more business processes increase the amount of information to be integrated into a process model. In this modeling scenario, business processes are represented in parallel planes and the third dimension is used to model relationships in particular semantic relationships, which overcome the limitation of a controlled vocabulary for process elements.

For this, we first calculate similarities between process elements according to [EKO07]. Four different similarity measures are used, which operate on the syntactical, the linguistic, the structural and the abstraction level of process element names. The syntactical measure unveils typos in process element names and the linguistic similarity measure detects synonyms by using the WordNet taxonomy¹. Homonyms are revealed by the structural measure, which considers a so-called context of process elements. The abstraction level similarity measure detects hyperonyms/hyponyms, which take into account the depth of terms in lexical reference systems such as WordNet. The total similarity value is calculated by a weighted sum of all four measures, which is only computed for the same type of elements (e.g. places vs. places or transitions vs. transitions). If a certain threshold of the total similarity value is reached², these two elements are defined as *similar* and a connection will be shown in the modeling environment.

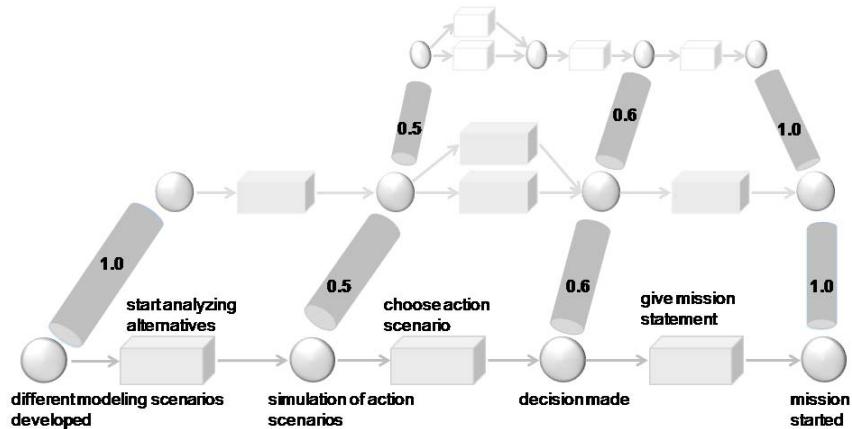


Figure 5 : Representation of similarities between process elements

The value of similarity is visualized analogously to social network analysis tools [WaF94, KRW06, SWS08], e.g., with thickness (more or less bold) or color of the connection cylinders (e.g. scale from bright over gray to black or from green over yellow to red). Users can visualize semantic similarities between places, transitions or both. Figure 5 shows only similarities between places visualized by more or less thick cylinders. The actual similarity value is shown inside the connection bar.

With respect to the given example, when calculating the similarities between another business process and the process from the example the insertion of semantic similarities

¹ <http://wordnet.princeton.edu/>

² Our experiments have shown that a threshold of 0.4 is a satisfying value.

would graphically complicate the understandability of the process models. Therefore, we regard a third dimension in this modeling scenario as effective for easily creating and viewing relationships between process models.

The next section describes how users can efficiently navigate through a hierarchically structured process model.

4.4 Modeling Hierarchies

For Petri nets, several formal net transformations such as refinement and coarsening were proposed to realize a stepwise hierarchical design of processes [Pet73]. In a transition refinement, a transition is refined to a sub-process. It is useful to consider only transition refinements with distinguished input and output transitions [Des05]. Thus, the pre-set of the input transition and the post-set of the output transition are equal to pre- and post-set of the transition to be refined. Coarsening is inverse to refinement where more specific process elements are subsequently linked together to coarse-grained process models.

The current tool implementations support the navigation through the different modeling levels of a business process. Continuing our running example, the navigation is a time consuming task because users need to click inside each sub-concept manually and only one single hierarchy level can be shown at one point of time. We use the third dimension for an efficient navigation through a sequence of refinements. Additionally, through the usage of the third dimension we can restrict the view of hierarchies on relevant refinements and highlight interrelated border elements of the sequence of refinements.

To realize the navigation through a sequence of refinements, we first introduce the notion of a sequence of refinements. A refinement (respectively a coarsening) is a surjective net morphism introduced by [Pet73]³. Let B_1, \dots, B_n be business process models with disjoint sets of nodes and f_1, \dots, f_{n-1} mappings such that (B_{i+1}, B_i, f_i) , $i=1, \dots, n-1$ are quotients. A net morphism (B_2, B_1, f) is called a quotient (B_2 is called a refinement of B_1), iff the mapping $f: X_2 \rightarrow X_1$ denoted by $B_2 \rightarrow B_1$ is surjective on both the nodes and the arcs. Then $SR = (B_1, f_1, \dots, f_{n-1}, B_n)$ is called a sequence of refinements.

Next, we can highlight the same pre- and post-sets of all sequences of refinements. Thus, users can easily catch the relationship between processes. To highlight the same pre- and post-sets we consider a set of business processes and a sequence of refinements. If a business process has a refined transition then we assign the same color to the pre-set of the input transition to be refined and to the post-set of the output transition to be refined.

Figure 6 shows a sequence of refinements and the highlighted pre- and post-set of transitions to be refined. When double clicking on a refined transition, the user can open the next hierarchy level. Users can restrict the navigation through the sequence of refine-

³ Roughly speaking, a net morphism is a mapping from elements of a source net to elements of a target net which respects the bipartition and the flow relation of the source net [DeM91].

ments by moving processes in the background and thus processes appear “*invisible*”. In Figure 6, the user opened four refinements and moved one refinement in the background.

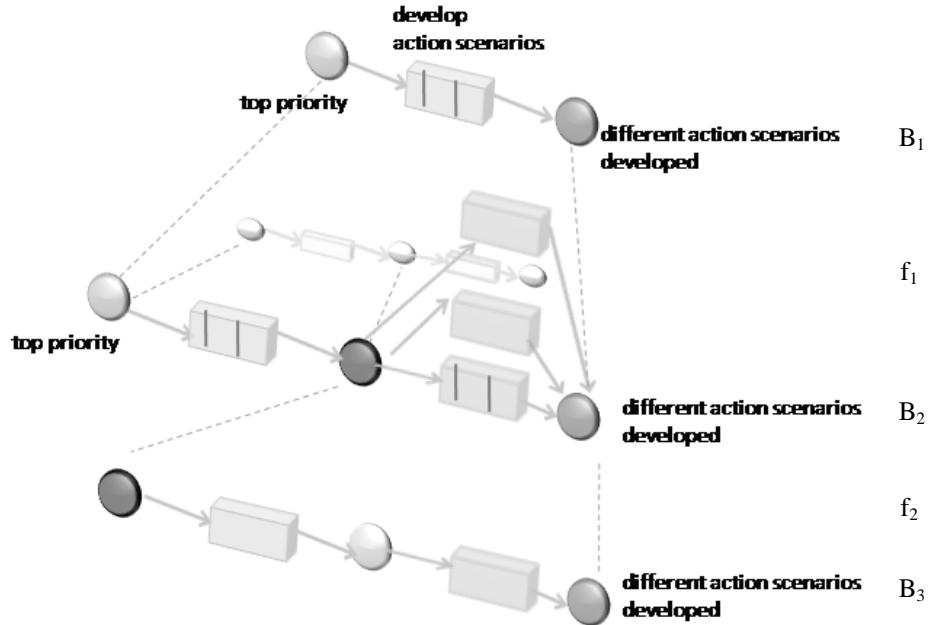


Figure 6: A sequence of refinements and highlighted pre- and post-sets of refined transitions

The next section presents some implementation aspects of our 3D process modeling environment.

5 Implementation

The Petri net Markup Language (PNML) [JKW00] is a standardized XML-based format for persistently storing Petri net models and exchanging them among different tools. The current version of PNML defines a large variety of concepts for describing Petri nets, e.g., Core Model, Multisets, Partitions, High-level Core Structure, Symmetric Nets, etc. Nevertheless, it still provides no means to specify properties for 3D representation of Petri nets, e.g., graphical position and size of Petri net elements in 3D space. Therefore, as the first step of implementing 3D visualization of Petri nets, we extend the PNML grammar with 3D-specific concepts. The extension is limited within the element type "toolspecific.element" that is abstractly defined in all versions of PNML Core Model to store tool specific information. In this way, the concepts for 3D representation of Petri nets are relatively independent from other standardized PNML definitions and can be flexibly reused if the PNML used by a tool needs to be upgraded. By adhering to this limitation, 3D Petri net models can also be interpreted and used as 2D models (despite semantic loss) by other PNML-based tools if they provide no support for 3D visualiza-

tion of Petri nets. To conform to standard PNML grammar, the concepts for 3D representation of Petri nets are defined in RELAX NG [CIM01], a simple schema language for XML, which has in some aspects advantages over XML Schema, e.g., support for unordered and non-deterministic content, ability to be algorithmically converted into both XML Schema and DTD.

Figure 7 shows on the left side a UML class diagram that defines the package "3DToolInfo" as an extension of the PNML Core Model. On the right side, a code fragment of the corresponding RELAX NG schema is displayed. For simplification purposes, we define in this package only 3D-specific concepts, which can be combined and interleaved with other concepts defined elsewhere in the same-named type "toolspecific.element" by using the RELAX NG-specific attribute "combine" with the value "interleave". The element "toolspecific" defined in "toolspecific.element" has the two child elements "referenceNet" and "elementGraphicsContribution". The element "referenceNet" is used to store reference relationships between Petri net models and contains information to identify and locate the reference net, e.g., ID and path of the reference net, ID and reference ID of reference nodes in the net. In 3D space, these relationships can be depicted by drawing for example dotted lines between nodes and corresponding reference nodes. The element "elementGraphicsContribution" contributes to Petri net element graphics with dedicated properties defined on the third coordinate axis. For instance, "zCoordinate" specifies the position of Petri net elements in the third dimension; "depth" defines together with "width" and "height" the 3D size of the elements; "zOffset" is used to position labels whose location is determined by the offset to the position of their owners; "zScaling" specifies the scale of arrow heads of arcs or directed lines in the third dimension.

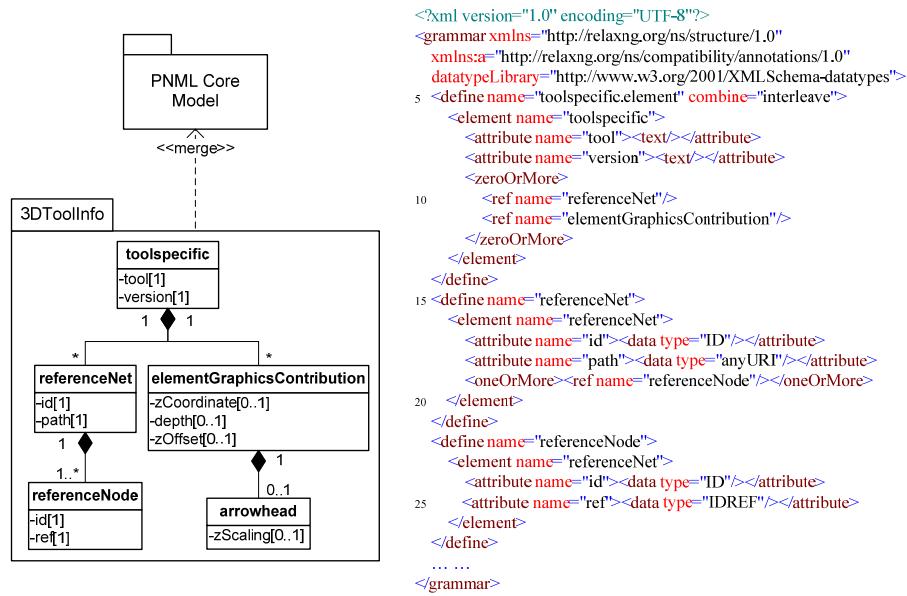


Figure 7: The package 3DToolInfo and corresponding RELAX NG schema

Organization models, roles and resources are important process objects in business process modeling. With our technique, they are persistently stored in separated XML files typified by RELAX NG schemas and bound to PNML documents via links. To better visualize these process objects in 3D space, 3D graphics and images are used whose path, z-offset, depth and other properties like transparency are also specified in the respective RELAX NG schemas.

Based on the extended PNML grammar and related RELAX NG definitions for process objects, a prototype for the purposed 3D visualization of Petri net has been implemented. Figure 8 shows a screenshot of a 3D business process modeled with our prototype.

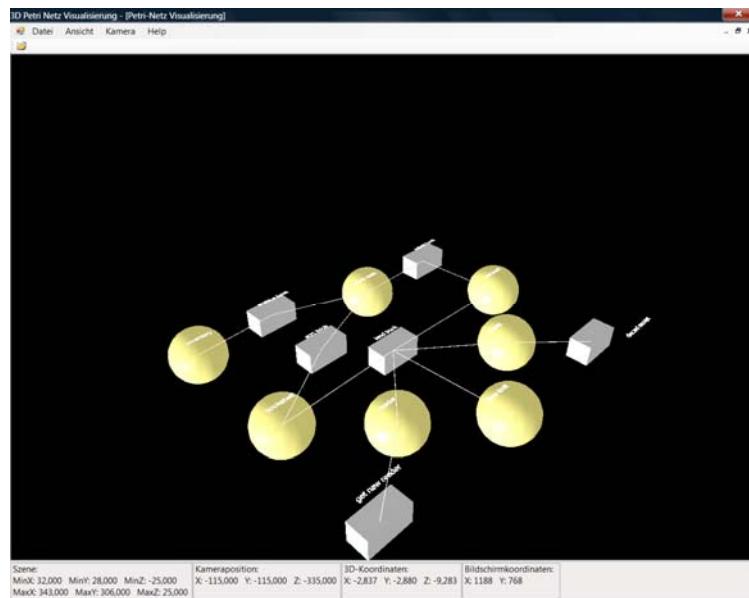


Figure 8: Screenshot of the 3D modeling environment

6 Conclusion and Outlook

In this paper we presented a Petri net-based approach for the integration of a third dimension into the representation of business process models. We presented related work in the areas of business process modeling, 3D modeling, and graphical layout methods. Based on this, we discussed some deficits of current 2D process modeling and showed how to improve layout and to increase information content of Petri net models by introducing a third modeling dimension. The need for a third modeling dimension was pointed out in three different modeling scenarios: modeling relationships between roles, resources and activities, modeling semantic relationships between business processes, and modeling hierarchies. Within these scenarios we showed that the 3D representation of processes facilitates the access to process-specific information. Whilst in conventional 2D representation process-specific information interfere with each other, in 3D the in-

formation becomes straightforward. By turning the process model in its 3D environment, one can examine different views and gather easily process-specific information. For the implementation of our approach we extended the PNML grammar by defining concepts for visualization of 3D Petri nets in RELAX NG.

Future work comprises the evaluation of our approach and the integration of the implemented prototype into the Petri net-based process modeling framework INCOME2010 [KLO08]. In addition, 3D visualization and animation of other process objects (e.g. process metrics such as time, cost, etc.) will be explored in our future research. In this contribution, we are focusing on a 3D representation concerning the control flow of processes. A further step in the future will be a 3D representation concerning the data flow of processes (e.g. XML documents in XML nets). Furthermore, for simulation purposes we intend to introduce a dynamic third dimension for process elements (e.g. places, symbolized as balls in 3D, change their volume and/or color according to the number of tokens they contain).

References

- [BES00] Ballegooij, A.; Elliens, A.; Schönhage, B.: 3D Gadgets for Business Process Visualization – A Case Study. In: Proceedings of the fifth symposium on Virtual reality modelling language, pp. 131-138, 2000.
- [BRB07] Bobrik, R.; Reichert, M.; Bauer, T.: View-based process visualization. In: 5th International Conference on Business Process Management, Springer, pp. 88-95, 2007.
- [CLM01] Clark, J.; Murata, M.: RELAX NG Specification, OASIS Committee Specification, 2001.
- [DGB07] Decker, G.; Grosskopf, A.; and Barros, A.: A Graphical Notation for Modeling Complex Events in Business Processes. In: Proceedings of the 11th IEEE international Enterprise Distributed Object Computing Conference, IEEE Computer Society, Washington, DC, 2007.
- [DeM91] Desel, J.; Merceron, A.: Vicinity respecting net morphisms. In: Proceedings on Advances in Petri Nets, G. Rozenberg, Ed. Springer-Verlag New York, New York, NY, 2001, 165-185.
- [Des05] Desel, J.: Process Modeling Using Petri Nets, In: M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede, Process-Aware Information Systems, pp. 147-177, 2005.
- [EKO07] Ehrig, M.; Koschmider, A.; Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling, volume 67, pp. 71-80. Australian Computer Science Communications, Ballarat, Australia, 2007.
- [HaL08] Hadar, I.; Leron, U.: How intuitive is object-oriented design? Communications of the ACM, vol. 51(5), pp. 41-46, 2008.
- [JaB96] Jablonski, S.; Bussler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation, Itp New Media, October 1996.
- [JaG07] Jablonski, S.; Goetz, M.: Perspective Oriented Business Process Visualization. In: Proceedings of the Business Process Management Workshops, Lecture Notes in Computer Science 4928, pp. 144-155, Springer, BPM 2007.
- [Jen94] Jensen, K.: An Introduction to the Theoretical Aspects of Coloured Petri Nets. In: Reflections and Perspectives, Lecture Notes in Computer Science, pp. 230-272. Springer, 1994.

- [JKW00] Jüngel, M.; Kindler, E.; Weber, M.: Towards a Generic Interchange Format for Petri Nets. In: Meeting on XML/SGML based Interchange Formats for Petri Nets, pp.1-5, Aarhus, Denmark, 21st ICATPN, 2000.
- [KaK89] Kamada, T.; Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters 31, 1989.
- [KHG08] Koschmider, A.; Habryn, F.; Gottschalk, F.: Real Support for Perspective-compliant Business Process Design. In: 4th International Workshop On Business Process Design. Milan, Italy, 2008.
- [KiP04] Kindler, E.; Páles, C.: 3D-Visualization of Petri Net Models: Concept and Realization. In: Proceedings of International Conference of Applications and Theory of Petri Nets, pp. 464-473, Springer, 2004.
- [KRW06] Klink, S.; Reuther, P.; Weber, A.; Ley, M.: Analysing Social Networks Within Bibliographical Data, Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA), Krakow, Poland, September 2006.
- [KLO08] Klink, S.; Li, Y.; Oberweis, A.: INCOME2010 - a Toolset for Developing Process-Oriented Information Systems Based on Petri Nets. In: Proceedings of International Workshop on Petri Nets Tools and Applications (PNTAP 2008, associated to SIMU-Tools 2008). Marseille, France, 2008.
- [KGM99] Krallmann, H.; Gu, F.; Mitritz, A.: ProVision3D - Eine Virtual Reality Workbench zur Modellierung, Kontrolle und Steuerung von Geschäftsprozessen im virtuellen Raum. Wirtschaftsinformatik 41(1): pp. 48-57, 1999 (in German).
- [LeO03] Lenz, K.; Oberweis, A.: Inter-Organizational Business Process Management with XML Nets. In: Petri Net Technology for Communication-Based Systems, Advances in Petri Nets volume 2472 of LNCS, pp. 243-263. Springer- Verlag, 2003.
- [LHG07] Li, L.; Hosking, J.; Grundy, J.: Visual Modelling of Complex Business Processes with Trees, Overlays and Distortion-based Displays. In: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing VLHCC, IEEE Computer Society, Washington, DC, 137-144, 2007.
- [MaM07] Maier, S.; Minas, M.: A Pattern-Based Layout algorithm for Diagram Editors. In: Proceedings of the Workshop on the Layout of (Software) Engineering Diagrams, 2007.
- [MeS06] Mendling, J.; Simon, C.: Business Process Design by View Integration, In: Business Process Management Workshops, pp. 55-64, Springer, 2006.
- [Mil08] Miles, R.: Microsoft XNA Game Studio 2.0. In: Microsoft Press Corp., 2008
- [NeK05] Nestmann, F.; Kron, A.: Frühwarnung und Management von Katastrophen am Beispiel Hochwasser. In: Naturgefahren und Kommunikation, Beilage der Zeitschrift GAIA – Ökologische Perspektiven für Wissenschaft und Gesellschaft Heft 14/3, 2005.
- [Pet73] Petri, C.A.: Concepts of Net Theory. Mathematical Foundations of Computer Science. In: Proceedings of Symposium and Summer School, High Tatras, Mathematical Institute of the Slovak Academy of Sciences, pp.137-146, 1973.
- [Pla02] Plate, E.J.: Flood risk and flood management. In: Journal of Hydrology 267 2–11, Elsevier, 2002
- [Pur98] Purchase, H. C.: Performance of Layout Algorithms. In: Comprehension, not Computation, Journal of Visual Languages & Computing Volume 9, Issue 6, pp. 647-657, 1998.
- [Rei93] Reiss, S.: A Framework for Abstract 3D Visualization. In: Proceedings of the IEEE Symposium on Visual Languages, pp 108-115, IEEE Computer Society Press, 1993.
- [ReR98] Reisig, W.; Rozenberg, G. (Eds.): Lectures on Petri Nets I: Basic Models. Volume 1491, Lecture Notes in Computer Science, Springer, 1998.
- [RiB06] Rinderle, S.; Bobrik, R.; Reichert, M.; Bauer, T.: Business Process Visualization - Use Cases, Challenges, Solutions. In: Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS'06), pp. 204-211, 2006.
- [Röl07] Rölke, H.: 3-D Petri nets. In: Petri Net Newsletter, No. 72, pp 3-9, April 2007.
- [Sch99] Scheer, A.-W.: ARIS – Business Process Modeling, Springer Verlag, Berlin, 1999.

- [SWS08] Said, Y. H., Wegman, E. J., Sharabati, W. K., and Rigsby, J. T. 2008. Social networks of author-coauthor relationships. *Comput. Stat. Data Anal.* 52, 4 (Jan. 2008), 2177-2184. DOI= <http://dx.doi.org/10.1016/j.csda.2007.07.021>
- [Ver00] Verbeek, E.: ExSpect 6.4x product information. In: Mortensen, K. (Ed.): Petri Nets 2000: Tool Demonstrations, pp 39-41, 2000.
- [WaF94] Wassermann, S; Faust, K.: Social Network Analysis: Methods and Applications, Cambridge University Press, New York, 1994.

Designing and Utilising Business Indicator Systems within Enterprise Models – Outline of a Method

Ulrich Frank, David Heise, Heiko Kattenstroth, Hanno Schauer

Chair of Information Systems and Enterprise Modelling

Institute for Computer Science und Business Information Systems (ICB)

University of Duisburg-Essen,

Universitätsstr. 9, 45141 Essen, Germany

{ulrich.frank|david.heise|heiko.kattenstroth|hanno.schauer@uni-duisburg.esen.de}

Abstract: The design of effective indicators and indicator systems requires a profound understanding of the relevant business context. Numerous relations and dependencies within an indicator system exist, which need to be analysed thoroughly: Many relations are based on implicit assumptions or sometimes not known by the management at all. This is of particular relevance for business success, since improperly used indicator systems may lead to ‘dysfunctional effects’ like opportunistic behaviour. This paper outlines a method for designing and utilising indicator systems. It fosters a convenient and consistent definition and interpretation of indicator systems. Furthermore, it serves as a conceptual foundation for related performance management systems, such as dashboard systems.

1. Motivation and Scope

In recent years, there has been an increasing demand for indicators to guide and justify management decisions. These business indicators – often referred to as Key Performance Indicators (KPI) – promise to reduce complexity and to promote a focus on relevant goals. Therefore, they are regarded by some as a key instrument of professional management, especially with regard to supporting, measuring, and monitoring decisions ([Si90], p. 12). For instance, in the field of Performance Measurement (PM) indicators are supposed to reflect aspects that are pivotal for certain decisions; therefore, strategies and goals are repeatedly refined and put in more concrete terms until they become measurable by KPIs. As a result, managers receive a set of various indicators that aim at the measurement of an enterprise’s performance (called ‘indicator system’). Well-known examples of indicator systems are ZVEI [ZE89] and the *RL* indicator system [Re06] for the financial domain as well as the *Performance Pyramid* [LC91] or the *Balanced Scorecard* [KN92] for the PM domain. Indicators are generally understood as quantitative measures that are specified using several mathematical constructs (e.g., ordinal or cardinal scale) for different types of reference objects and on various levels of abstractions in the enterprise (e.g., resources, processes, or business units; [Gr02] pp. 98ff.). By using indicator systems managers can define expected target values – e.g., by referring to benchmarks –, continuously monitor them and, in case of discrepancies, intervene into the enterprise’s process execution.

However, the design of indicator systems is not a trivial task. Already the specification of an indicator does not only require a profound understanding of the corresponding decision scenario but also requires considering its relations to other indicators. Furthermore, it recommends taking into account how an indicator affects managerial decision making. If managers, for instance, regard an indicator as an end in itself, it will result in opportunistic actions that are likely to not be compliant with the objectives of a firm. This is even more important since indicator systems often have a strong influence on the enterprise [KN92], because managers and other stakeholders are incited to predominantly align their behaviour with specific (maybe mandatory) indicators and associated target values only. If indicator systems do not adequately address these challenges, they are likely to fail their purpose. Against this background, we assume that indicators should not be specified separately, but with particular attention to the relations that exist between them and to the business context they are utilised in. Both, complexity and criticality of the task to design systems of interrelated indicators recommend making use of a dedicated method that explicitly addresses those challenges. In this paper, we present a proposal of a method for designing and utilising indicator systems, which is based on a domain specific modelling language and aims at fostering transparency, validity, and reliability of indicator systems. It is intended to become part of the *multi-perspective enterprise modelling* method *MEMO* [Fr02] in order to facilitate semantically rich linking and integration of models of indicator systems with models of the corresponding business context – and vice versa.

The remainder of the paper is structured as follows: In section 2 important domain specific requirements for dealing with indicator systems are discussed. Based on these requirements, the elements of a method for the design and utilisation of indicator systems are presented in section 3, accompanied by first suggestions for a possible solution. The paper closes with related work (section 4) and an evaluation, concluding remarks, and an outlook to future work (section 5).

2. Business Indicator Systems: Requirements for a method

Based on assumptions about the quality of indicator systems used in business practise and resulting challenges, we develop requirements of the domain for the intended method.

The specification of an indicator system is a complex task that requires accounting for many aspects. Indicator systems that lack important aspects or are partially inconsistent jeopardize their very purpose.

Requirement 1: The method should support – and if possible: enforce – the design of comprehensive and consistent indicator systems. For instance, there should be no contradictions or conflicts between indicators – if conflicts cannot be avoided, they should be made explicit (cf. [Ge86] pp. 114ff.).

The adequate use of an indicator system implies a knowledgeable and differentiated interpretation. For many decision scenarios a plethora of indicators is available

(cf. [LO06] p. 1). This hampers the selection of indicators that are significantly expressive regarding the underlying scenario, business strategies, and goals. Unsuitable indicators hold the dangers of promoting misleading conclusions and unfortunate decisions that – in the worst case – impede the achievement of the enterprise's goals (cf. [Ec05] pp. 197f.).

Requirement 2: To support the user with an appropriate interpretation, the documentation of an indicator system should be enriched with relevant context information. This requires not only to offer concepts that represent indicators, but also to account for concepts modelling the business context (e.g., strategies, goals, business processes).

The adequacy of indicators as well as their relations (e.g., between themselves and to the goals) might be of question. Both are often based on subjective judgement and, thus, presumptive, whilst others are based on empirical evidence and ‘notedly’ objective (cf. [Gr02] p. 128; [LO06] p. 15; [Wa01] pp. 66f.; [So05] p. 226).

Requirement 3: The method should provide concepts that allow for a differentiation of the rationale underlying an indicator and its relations.

Indicator systems are relevant for and used by various groups of stakeholders. The specific preferences as well as the level of expertise vary within these groups. For instance, a process manager will have different demands on indicators than an IT manager or a business manager regarding the types of indicators as well as the levels of detail and abstraction (cf. [Gl01] p. 8).

Requirement 4: The method should support a meaningful representation of indicator systems on various levels of abstraction to satisfy the needs of multiple groups of prospective users. Further, the method should support the integration of an indicator system designed for a certain perspective with indicator systems of other perspectives.

For effective decision support, instances of indicator systems, i.e., concrete indicators should be provided and visualized by special tools such as dashboard systems.

Requirement 5: The method should foster the construction of corresponding software systems by providing a conceptual foundation as well as guidelines for visualising indicators.

3. Elements of the method

The approach we chose to develop such a method is to enhance an existing method for enterprise modelling (EM) by concepts and further components for designing and utilising indicator systems. An enterprise modelling method in particular provides some prominent advantages:

- A (graphical) modelling language promises – similar to the use of conceptual models in software engineering – to support a rich and intuitive documentation, which fosters the communication between stakeholders with different professional backgrounds (cf. *Req. 4*). Further, it can depict manifold relations in a more comprehensible way than, e.g., a sequential textual description.
- A modelling language is based on a formal syntax and precise (if not formal) semantics, which allows for automated analyses, for transformations into models used for software development such as, e.g., object models for IT management software, as well as for interchanging indicator systems and their data between different information systems or enterprises (cf. *Req. 5*).
- By embedding it into an existing EM method, the proposed method additionally benefits from taking into account the relevant business context (cf. *Req. 2*).

Furthermore, we decided to use a *domain specific* modelling language, since general purpose modelling languages (GPML) like the ‘Unified Modelling Language’ (UML, [OM07]) or the ‘Entity-Relationship-Models’ (ERM, [Ch76]) show serious deficiencies with regard to our purpose (cf. [EJ01], [Lu04]). First, a GPML would not effectively support the construction of consistent models, since its syntax and semantics allows for expressing almost anything (*lack of integrity*). Second, it would be rather inconvenient to describe indicators using only generic concepts such as ‘Class’, ‘Attribute’, or ‘Association’ (*lack of convenience*) – the main concepts of the application domain are to be reconstructed by means of conceptual modelling in order to facilitate comfortable, intuitive, and secure modelling (cf. *Req. 1*). Third, the rather generic graphical notation (concrete syntax) of a GPML does not contribute to an illustrative visualisation (*poor comprehensibility of models*).

The method we suggest for multi-perspective indicator modelling consists of several parts. It is embedded in a context of additional abstractions and artefacts (Figure 1): The modelling concepts (abstract/concrete syntax) are defined in the Performance Modelling Language which we – within this paper – call SCORE-ML. A corresponding process model, which covers the entire lifecycle of indicator systems (*design, use and refinement*), guides the application of these concepts. In addition, there is an extensible set of reference indicator systems (‘reference models’) that are reconstructions of existing indicator systems (such as ZVEI) by means of the SCORE-ML. Further, the literature supplies a large variety of suggestions (‘libraries’) for indicators to be used in specific scenarios like – for IT management – the indicators suggested in the ‘IT Infrastructure Library’ (ITIL, [OGC07]). Both, the reconstructed reference systems and the method itself, may be linked to those indicator libraries to enhance descriptions of indicators and to indicate their relevance. On the other hand, indicator libraries may also make use of the method in order to provide a richer and more illustrative specification of indicators and to guide their appropriate use. Thereby, designers of indicator systems can take advantage of both reusing existing indicators or indicator systems, respectively, and of a domain specific modelling language that guides the construction or configuration of consistent models.

In this paper, we will focus on the method in the narrow sense (shaded rectangle in Figure 1): in section 3.1, we describe the main concepts of the SCORE-ML in their current state; in section 3.2, we introduce the process model; reference indicator systems and indicator libraries are briefly addressed in the last chapter again.

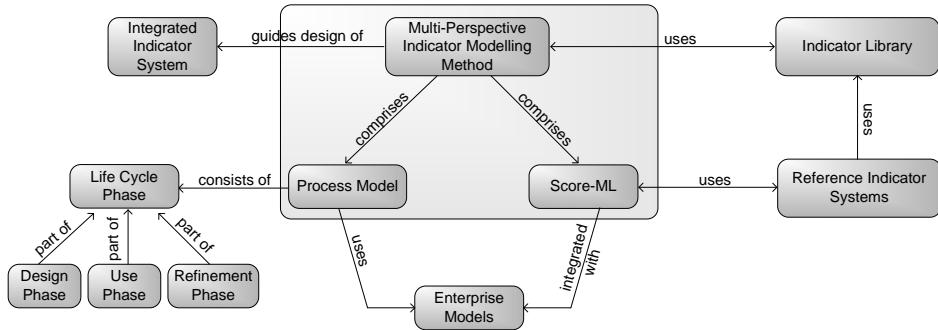


Figure 1 – Elements of the multi-perspective indicator modelling method

3.1 Modelling language: SCORE-ML

The specification of a language for modelling indicator systems faces a number of challenges.

Firstly, there is need to clarify the very conception of an indicator. From a modelling perspective, an indicator can be an attribute of an object, e.g., ‘monthly revenues’ of a product; it can be an aggregation of attributes of a collection of objects, e.g., the sum of ‘monthly revenues’ of all products; or it can represent a snapshot of certain features of an enterprise or the development of states over time, e.g., ‘average monthly increase of revenues’. Since our focus is on the design of indicator systems, we decided to introduce a specific abstraction rather than regarding indicators as attributes of objects or object collections. This requires developing a conception of indicators that fosters a sophisticated specification. Such a conception includes core attributes and in particular a differentiated conception of relationships between indicator types. Secondly, we need to analyze how flexible and adaptable an indicator system is supposed to be. In the case of a database, the user can build individual indicators by using a data manipulation language (like in data warehouses). On the one hand, we do not want to leave the user alone on the level of a database schema, but provide him with more meaningful concepts to define indicators. On the other hand, it should be possible to adapt indicators to individual needs. Thirdly, it is required to differentiate between types and instances of indicators. An indicator system defines types of indicators, while indicators that are used to actually measure performance are instances. Fourthly, the semantics of indicators depends chiefly on the objects, they refer to, and the context they are used in. Hence, there is need for associating the concept indicator with reference objects and context information (cf. *Req. 2*). Figure 2 provides a suggestion for a meta model of the SCORE-ML that contains first – but not final – solutions for these challenges. Due to the given restriction in this paper we simplified the representation of certain aspects (see below).

Ad 1: To provide the users of the method with a rich, yet extensible concept of indicator, we defined a set of attributes and association types. Among others, indicator attributes include name, description, purpose, examples, availability (of required data), potential bias or preferred visualisation (e.g., ‘traffic light’, ‘bar chart’, ‘speedometer’). Predefined association types for indicators comprise ‘computed from’ and ‘similar to’. The relations realized by the meta type ‘CustomizedRelationship’ enable the qualification of further customized relations and, as a special case, cause-and-effect relations between indicators: The attribute ‘relationSpecification’ can be used to provide information about the type of the relation – for instance, an indicator can be a leading or lagging indicator with a positive or negative effect direction. The attribute ‘presumptions’ may be used – similar to the corresponding attribute of ‘Indicator’ – to provide information about the underlying rationale (e.g., ‘empirical evidence’ or ‘subjective estimation’ in combination with a statement about the prevalent assumptions). Note, this meta type as well as the association types are simplifications in tribute to the given restrictions of this paper. The original meta model of the SCORE-ML contains more differentiated meta types for such relations between indicator types, including a differentiation into bi-, unidirectional, and value driver relationships and a more differentiated arithmetic relationship (exemplarily denoted as ‘computed from’). Additionally, a user can define individual attributes by instantiating the meta type ‘IndicAttribute’; association types that are restricted in cardinalities other than the ‘1..1’ in ‘CustomizedRelationship’ can be instantiated from the meta type ‘IndicLink’. Note that these meta concepts are also not shown in the meta model excerpt.

Ad 2: An indicator system is supposed to represent all indicator types for a certain group of users. The system can be adapted to individual requirements on two levels. On the type level, an indicator type can be associated to a reference object type via the meta type ‘SpecificIndicator’. This would allow one, for instance, to define an indicator type such as ‘AverageCost’ assigned to a business process type. On the instance level, an indicator can be defined by assigning it to a set of instances of a reference object type, e.g., to a projection on the instances of a business process type (see below). Reference object types can be further differentiated into ‘dimensions’ such as time period, regions etc. (cf. the multi-dimensional modelling in the field of data warehouses). Note that this is out of the scope of the indicator system itself. Instead, this kind of flexibility depends on the concepts applied to models of reference objects. On instance level, it depends on retrieval/navigation capabilities provided by corresponding information systems.

Ad 3: There are certain apparent features of indicators that we cannot express with the specification of indicator types, since they are used to represent instance states. Especially with respect to *Req. 5* it would not be satisfactory to neglect such instance level features. This applies, e.g., to the particular point in time an instance of ‘SpecificIndicator’ is created at or to its concrete values. To meet this challenge, we suggest the concept of ‘intrinsic feature’ [Fr08]. An intrinsic feature is an attribute or an association that reflects a characteristic that we associate with a type that applies, however, only to the instance level. Hence, an intrinsic feature within a meta model is not instantiated on the type level, but only one level further down, i.e., on the instance level. In the meta model in Figure 2, the intrinsic association ‘measures’ allows for assigning sets of instances of a reference object type to an instance of ‘SpecificIndicator’. In the MEMO Meta Model-

ling Language [Fr08], which is used to specify the present meta model, intrinsic features are marked by an ‘i’ that is printed white on black.

Ad 4: To enrich indicator types with semantics, they can be associated to goals, decision scenarios, and reference objects. Every indicator type can be associated to a goal type. In an ideal case, there is an elaborate model of a company’s goal system (which would be out of the scope of an indicator system). The meta type ‘DecisionScenario’ can be instantiated to a scenario type (e.g., ‘assessment of IT resources’), which in turn can be instantiated to a particular decision. An indicator type (‘SpecificIndicator’) can also be applied to certain reference objects. In the meta model, this is indicated through the abstraction ‘ReferenceObject’, which serves as a surrogate for meta types such as ‘BusinessProcess’, ‘Resource’, ‘Product’ etc. Note that the attributes ‘benchmark’ and ‘value’ apply to the type level, e.g., a reference value and the actual value for the average throughput time of the business process type ‘order management’. The intrinsic feature ‘particularValue’, on the other hand, serves to describe the value of a (projection of) process instance(s) – which may be interesting for a drill-down scenario in, e.g., a dashboard tool. The meta model includes a number of additional OCL constraints. The excerpt in Figure 2 shows only one constraint, which ensures that a ‘SpecificIndicator’ type can only be associated to a decision scenario type, if the corresponding ‘Indicator’ type is associated to the decision scenario type, too.

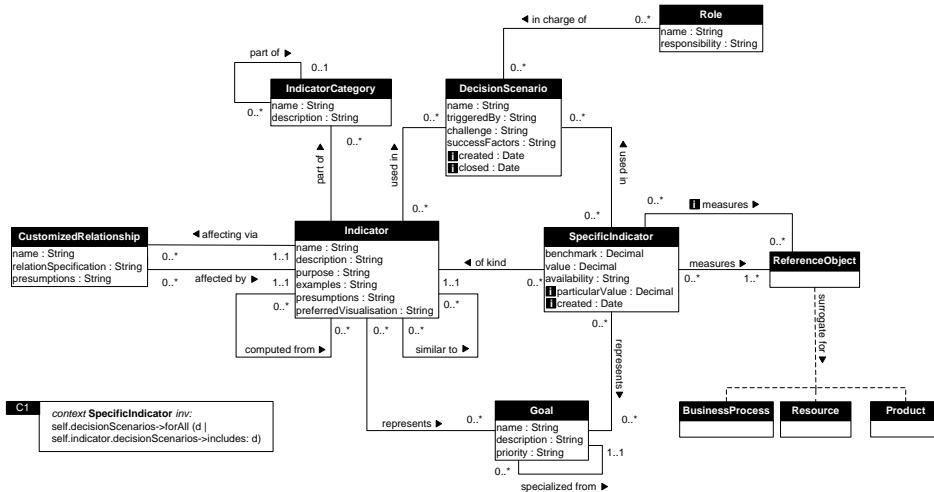


Figure 2 – Excerpt of the SCORE-ML meta model

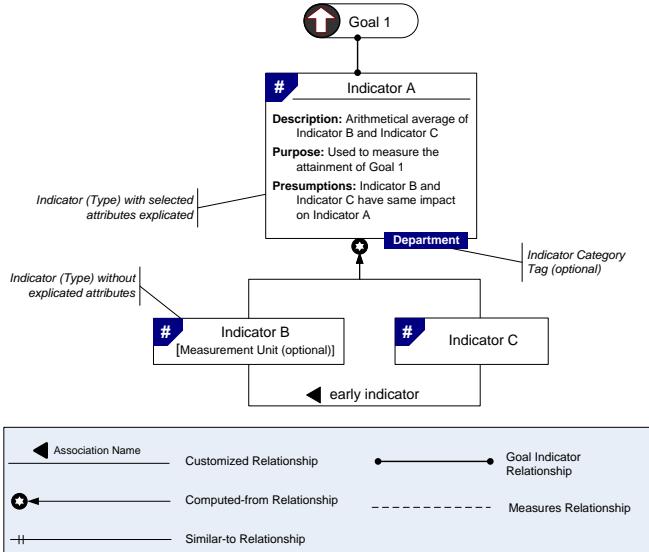


Figure 3 – Illustration of the graphical notation of the SCORE-ML

The specific notation of the SCORE-ML is illustrated in Figure 3. An exemplary model of an integrated indicator system is depicted in Figure 4. It shows the utilisation of the meta concepts goal (*Goal System*), indicators, and their relations (*Indicator System*). The business context is added in the ‘*IS & Business*’ layer, showing the association of specific indicators to reference objects. We enriched one specific indicator and one customized relationship with a full description of their attributes in order to illustrate the structured documentation provided by the SCORE-ML. Further, the example includes an instance level representation (*Instances*): For business process *A*, two instances are shown. They are enriched with an additional symbol (traffic light-metaphor) that visualises the current status (i.e., instance status) of the assigned ‘*SpecificIndicator*’ (here: throughput time).

3.2 Corresponding Process Model

The application of the presented modelling concepts is embedded in a process model that envisions the use of the SCORE-ML either ‘stand-alone’ or in combination with existing enterprise models (e.g., resource, process, strategy, and other models).

The process model can be divided into three phases, according to the life cycle of indicator systems: The first phase is the *design* of the indicator system (conceptual), the second phase describes the *use* of the indicator system, and the third is the continuous *refinement* of the indicator systems (both: ‘empirical’). Figure 5 gives an overview about the phases and their steps.

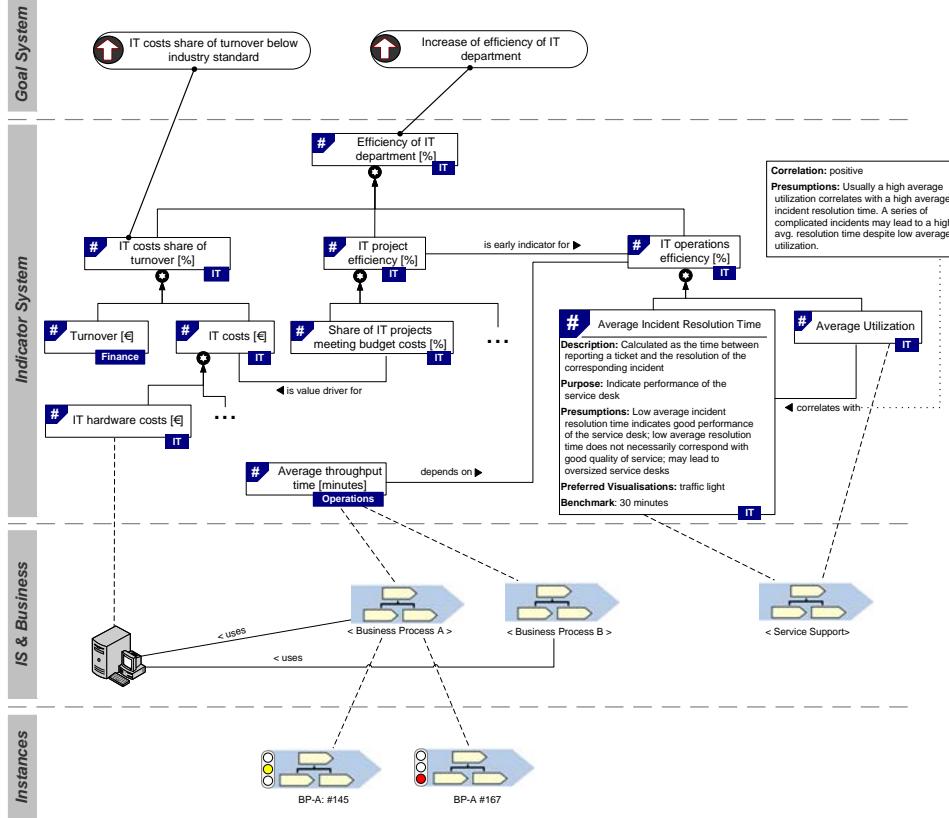


Figure 4 – Exemplary model of an integrated indicator system (inspired by [Kü06])

Design Phase

This phase can be differentiated into two shapes (cf. Figure 5): first, the ‘real’ design process (for advanced users, such as consultants and experts), wherein an indicator system is completely set up from scratch in order to match specific demands or realise advantage over existing indicator systems; second, a selection/configuration process for less ambitious users (e.g., end-users), wherein an existing indicator system – e.g., a reference indicator system – is reused and (where necessary) adapted for the specific needs. The latter one is mostly self-explanatory (and, for this reason, will not be further discussed in this paper). Hence, in the following we focus on the ‘real’ design process.

Step 1: Identification of relevant decision scenario, indicators, and data

In the process of setting up an indicator system, the identification of the relevant decision scenarios and corresponding indicators is one of the first and pivotal tasks. There are two directions to identify and define indicators with respect to a certain scenario: (i) defining indicators ‘top-down’ by operationalising strategies and goals (‘What do we want to measure?’), and (ii) ‘bottom-up’ by finding measurable aspects (‘What can we

measure?’) or by using mandatory indicators that are provided by the management. Both directions are not necessarily disjunctive, they can rather be used together.

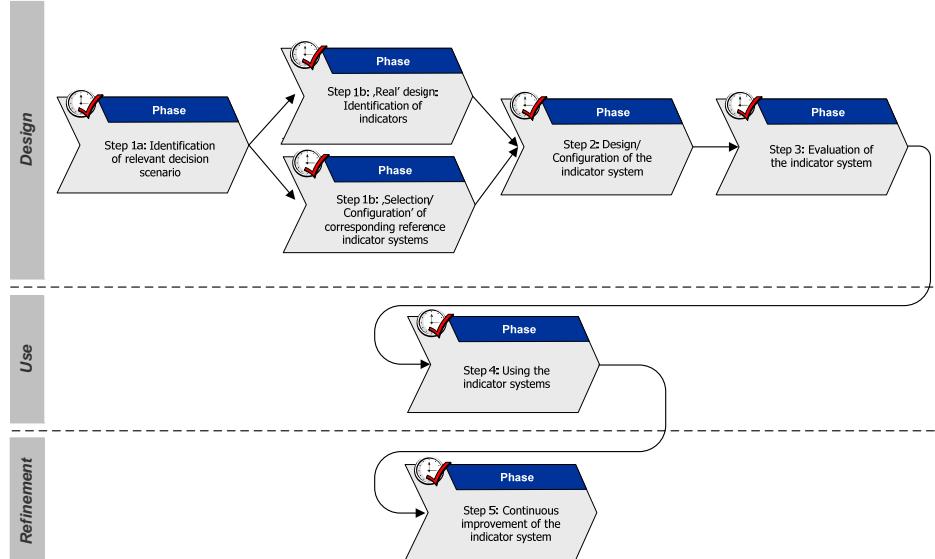


Figure 5 – Process model of the SCORE-ML

In the top-down direction, the enterprise’s strategies and goals are refined until the matter of their measurement occurs. While some are apparently easy to measure (e.g., the increase in revenue), others are hardly measurable to a satisfactory extent. In SCORE-ML, each ambiguous indicator can (and should) be marked by means of the according attribute (*presumption*). Thereby, the user is able to express and, to a certain degree, forced to think about the potential distortions of the indicator. In the bottom-up approach, the user searches for all those indicators he *can* measure. In both approaches, the identification of indicators can be supported by enterprise models. For instance,

- (IT) resource models (such as provided by the MEMO RESML [Ju07] and the MEMO ITML [Ki08]) offer an overview about the resources, their main properties (e.g., capacity, costs) and their relationships; this also includes information about their usage in processes ('resource allocation').
- business process models (such as in MEMO ORGML, cf. [Fr02]) describe the course of action(s) in a company; they provide details about the activities like times (e.g., execution, transport), exceptions, costs, and risks.
- strategy models (such as in MEMO SML, cf. [FL07]) represent the company’s strategies, goals, measures (in terms of strategic actions), and their relations, for instance, to the processes and resources in an enterprise (esp. supporting the top-down approach).

These models further support the search for potential indicators by showing which reference objects are of relevance: the business context provided by reference objects supports the user in assessing the relevance, usefulness, and appropriateness of a related indicator. Furthermore, the models also offer potential indicators in terms of attributes and quantifications (e.g., time-related attributes). Additionally, the aforementioned indicator libraries can serve as a basis for the identification of indicators: for certain domains and decision scenarios they offer lists of recommended indicators. Instead of simply applying these indicators directly, the user is supported in relating them to other indicators and the business context and enriching those indicators with his presumptions.

Step 2: Design of the indicator system

After the identification of potential indicators, the user designs the indicator system by defining relations between indicators. In some cases a relation seems to be obvious (e.g., due to ‘common sense’). However, in this method the user is recommended to think about the rationale underlying this relation, consciously decide for it, and make it explicit (‘presumptions’). If he concludes after all that the relation is not as obvious as it seems, he is able to expatiate his presumptions. In doing so, he fosters the maturity of the emerging indicator system (cf. refinement phase). Furthermore, the user can incorporate the related business context to identify (potential) relations between indicators in two ways:

- (i) For each indicator associated to a reference object on a specific level of abstraction, the user can check for and assess potentially affected indicators by ‘following’ the associations of this reference object within the enterprise model. For instance, an indicator measuring a specific characteristic of a resource (e.g., ‘IT hardware costs’ – cf. Figure 4) might have an influence on indicators measuring the processes the resource is allocated to (‘Average throughput time’). The user can then enrich the indicator system by making this relation between these two indicators explicit, i.e., associating them via a customized relationship – in this case denoting a ‘bidirectional correlation’ with the rationale ‘subjectively estimated’. In addition, some enterprise models already express causal relationships (e.g., cause-and-effect relations of costs [He08], risks, or goals [FL07]). Here, the user can comfortably assess these relations between reference objects whether they are also applicable to indicators.
- (ii) Enterprise models allow for integrating the different perspectives on a company (e.g., finance, operations, and IT). If the indicator systems of the different perspectives are all modelled in the SCORE-ML and stored in an enterprise-wide repository, indicators from other perspectives measuring the *same reference object* as well as relations to indicators in other indicator systems can be identified (e.g., further cause-and-effect relations associated with the indicator ‘Turnover’ (cf. Figure 4) within a financial indicator system). Thus, a multi-perspective evaluation is possible that goes clearly beyond the expressiveness of, e.g., the balanced scorecard – allowing for a tool-supported identification of potential conflicts, similarities, or need for support.

Step 3: Evaluation of the indicator system

The implementation and use of an indicator system has a substantial impact on the management of a firm. Therefore, it is mandatory to thoroughly evaluate an indicator system before using it. This is even more the case as the design of indicator systems will often involve different stakeholder with specific goals and interests. At first, the associations within an indicator system need to be analysed with respect to their consistency. The attribute ‘relationSpecification’ allows for analysing if an indicator might lead to actions (e.g., improving this indicator) that affect other indicators. Second, the system can be analysed for cyclic relations. Afterwards, conflicts should either be resolved or – if they are not resolvable – explicitly modelled in the indicator system. Besides this preliminary evaluation of indicator systems in the design phase, an accompanying evaluation is conducted in both the use and the refinement phase (which is omitted in Figure 5).

Use phase

The indicator systems can be used for different purposes (cf. [Gl01] p. 6; [Ge86] pp. 104ff.; [Re06] pp. 23f.), including communication between different stakeholders, reporting, and management decision support. Through its explicit and transparent representation of indicator types and relationships between them an indicator system fosters solidly founded decisions. For instance, an IT manager is guided to focus not only on his IT-related indicators but also to account for other affected business perspectives. Furthermore, an indicator systems can be used to estimate the (business-) impact of ‘tuning’ an indicator by analysing the effects on other indicators, goals, and even on the reference objects they measure (e.g., the various effects on a process and its other assigned indicators resulting from an optimization of its throughput time). Vice versa, the indicator systems can help to find the root-cause for a bad performing indicator. They may also help to find indicators on the operational level that serve as an instrument to achieve the desired effect on strategic level. Since underlying assumptions along with the maturity of indicators and relations are explicitly modelled (‘presumption’), both can be taken into account in these analyses and, thus, should promote a better understanding and more profound decisions. As an additional feature of according tools (cf. *Req. 5*), a dashboard or other performance management software may provide navigation/retrieval functionalities that enable users to choose specific projections of instance data related to a specific indicator (e.g., filtering of instance data such as ‘Average throughput time *during the last week and for product X*’).

Refinement phase

During the application of the indicator systems, the users might get a more precise understanding of the indicators and their relations – especially concerning the underlying rationale and assumptions. After using the indicator system for some time, initially suspicious indicators and relations can turn out to be evident and new relations might be identified. On the other hand, some of the assumptions that influenced the design of an indicator system may turn out to be inappropriate. By adapting the indicator system and updating this information, the maturity of the indicators and, as a result, of the indicator system as a whole can increase over time.

4. Related Work

While there is a plethora of publications on business indicators, only few approaches exist that aim at modelling indicators. For instance, Pourshahid et al. [Po07] introduce a meta model for modelling KPIs on an abstract level, but they do not allow for modelling relations between them. Wetzstein et al. [We08] and Nescheuler [Ne95] suggest more elaborate approaches. However, in contrast to the claim of defining a meta model [We08] or modelling language [Ne95], they both focus on indicator instances. The main purpose of the models they target is to serve as a foundation for developing software. Hence, the resulting representations are most likely not adequate for prospective users (while the related software, on the other hand, might be; cf. *Req. 5*). [Ai97] and [Kr05] each introduce a method that accounts for the business context. Unfortunately, neither presents a language specification and – as far as one can conclude from their application examples – they only provide one type of relation between indicators (i.e., subordination). Modelling tools like ARIS or ADONIS (ADOscore) offer concepts for specifying indicators; to a certain extent, their indicator types can be assigned to concepts representing the business context. However, they usually do not support different types of relationships between indicators – as well as language specifications are not available, too.

Modelling indicators is a pivotal aspect of data warehouse modelling. In order to provide a more convenient and consistent approach to modelling multi-dimensional data warehouses, Sapia et al. [Sa99] present an extension of the EERM. Their focus is on dimensions, which express the multi-dimensional structure of data and the navigation between these dimensions (e.g., roll-up, drill-down); concepts for the business context are not provided. A number of similar approaches exist. Mansmann et al. [MNS07], for instance, present a method for transforming business process models into data warehouse models in order to add specific analytical capabilities to business process management systems (e.g., for querying multi-dimensional data). They provide a foundation for modelling data warehouses and data collections – primarily of business process instances – and can complement our method in the task of data-management. However, their approach focuses on business processes only and does not intend to support users with a more meaningful concept of indicators.

Goal modelling is a further field that is related to indicator modelling. It is a research subject both in Artificial Intelligence and in Requirements Engineering. If applied to business contexts, the basic structure of according methods is fairly similar to the one of PM methods. However, the design rationale and application objective of goal modelling methods differ decisively from PM approaches: Their main focus is on fostering requirements specification within software development processes and/or facilitation of automated reasoning based on formally specified goals (e.g., through software agents, cf. [vL01] for an overview). Common approaches include i* [Yu95], Goal Oriented Requirements Engineering (GORE, e.g., [Ju08]), and KAOS (e.g., [vL03]). Respective modelling languages offer differentiated support for formalizing goals through logical expressions. However, these languages are not well suited as a management tool, since their application requires specific expertise, and – in particular – these approaches lack support for modelling contingent subjects or circumstances that – by their nature – impede adequate formalization.

5. Evaluation, Concluding Remarks, and Future Work

In this paper we outlined a modelling method for designing and utilising business indicator systems. It aims at supporting the design of indicator systems as control instruments that conform to the goals of an enterprise and account for the relevant domain context, which is – in an ideal case – represented by an enterprise model.

The method was designed to fulfil five requirements (cf. section 2): The core concepts of the SCORE-ML have been embedded into an existing EM method that also supports modelling of processes, resources, and goals, so that it allows for associating indicators with the relevant business context (*Req. 2*). Further, we introduced the concept ‘presumption’ of indicators and their relations in order to expatiate the underlying rationale (*Req. 3*). Due to the integration of the SCORE-ML with other modelling languages (here: the family of MEMO languages, which explicitly encourages different perspectives and levels of abstractions in the enterprise) the method supports different perspectives on business indicator systems (e.g., from IT management and business management), and hence, allows for the integration of different indicator systems into an enterprise-wide indicator system (*Req. 1 & 4*). The semantics of the SCORE-ML allows for transforming an indicator model into, e.g., a database scheme or a class diagram in order to develop software for managing and visualizing indicators (*Req. 5*).

Compared to other approaches for describing indicators, such as ‘traditional’ indicator descriptions (e.g., as used for ZVEI, RL or the Balanced Scorecard) or modelling languages for designing multi-dimensional data models, our approach shows clear advantages – mainly by featuring a higher level of semantics. Table 1 shows a comparative (though preliminary) evaluation based on the identified requirements (cf. section 2) and the challenges (cf. section 3.1).

In our future research, we will focus on the advance of the method and on a tight integration with existing MEMO modelling languages. This will probably result in giving up the SCORE-ML for an integration of its concepts with other languages, such as the ORGML, the RESML or the ITML. Although the method promises to overcome the shortcomings of the current design and utilisation of indicator systems, its application is bound to certain restrictions: The task of designing an elaborate indicator system is still time-consuming and demanding – especially for untrained personnel. To foster the acceptance of the method, we plan to integrate its concepts into MEMO Center, the MEMO modelling software environment. Additionally we work on reconstructing further established indicator systems in order to build reference indicator systems – in the sense of reference models – that allow for safe and convenient reuse; the first reconstructions are available at <http://openmodels.org>.

Similar to other modelling methods, the proposed approach should be well suited for education purposes, e.g., for analysing and discussing existing indicator systems. Especially the explicit consideration of underlying assumptions and possible conflicts fosters a better understanding of indicator systems from business practise and the preconditions of their appropriate use.

	Traditional Indicator System	Multi Dimen- sional Data Model	Domain Specific Modelling Language
<i>Reuse: Elaborate concept of ‘indicator’</i>	–	o	+
<i>Integrity: Protection against inconsistent models (cf. Req. 1)</i>	–	–	+
<i>Integrity: Support for appropriate interpretations (cf. Req. 2)</i>	–	–	+
<i>Analysis: Differentiated, formal description of indicators</i>	–	o	+
<i>Use: Support for adequate visualization (cf. Req. 4)</i>	o	o	+
<i>Reuse: visualization suited for indicator instances, too</i>	Usually no differentiation between type and instance level	–	+
<i>Software Development: Concepts can be transformed into elaborate implementation level representations (cf. Req. 5)</i>	–	o	+

Table 1 – Comparison of approaches for indicator systems
 (–: not satisfactory; o: accounted for; +: good)

Acknowledgements

We would like to thank Kirk Wilson for valuable comments on a previous version of the manuscript.

References

- [Ai97] Aichele, C.: Kennzahlenbasierte Geschäftsprozessanalyse. Gabler, 1997.
- [Ch76] Chen, P. P.: The Entity-Relationship Model – Toward a Unified View of Data. In: ACM Transactions on Database Systems 1 (1976) 1, pp. 9-36.
- [Ec05] Eckerson, W. W.: Performance Dashboards: Measuring, Monitoring, and Managing Your Business. Wiley & Sons, Hoboken, NJ, 2005.
- [EJ01] Esser, R.; Janneck, J. W.: A Framework for Defining Domain-Specific Visual Languages. In: Proceedings of the Workshop on Domain Specific Visual Languages at OOPSLA 2001, Tampa Bays, 2001.
- [Fr02] Frank, U.: Multi-Perspective Enterprise Modeling (MEMO): Conceptual Framework and Modeling Languages. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), Honolulu, 2002.

- [Fr08] Frank, U.: MEMO Meta Modelling Language (MML) and Language Architecture (revised version). ICB Research Report. Universität Duisburg-Essen. 2008. URL: http://www.icb.uni-due.de/fileadmin/ICB/research/research_reports/ICB-Report_No24.pdf
- [FL07] Frank, U.; Lange, C.: E-MEMO: a method to support the development of customized electronic commerce systems. In: Inf. Syst. E-Business Management 5 (2007) 2, pp. 93-116.
- [Ge86] Geiß, W.: Betriebswirtschaftliche Kennzahlen. Theoretische Grundlagen einer problemorientierten Kennzahlenanwendung. Lang, Frankfurt am Main u.a., 1986.
- [Gl01] Gleich, R.: Das System des Performance Measurement : Theoretisches Grundkonzept, Entwicklungs- und Anwendungsstand. Vahlen, München, 2001.
- [Gr02] Grüning, M.: Performance Measurement-Systeme: Messung und Steuerung von Unternehmensleistung. Deutscher Universitäts-Verlag, 2002.
- [He08] Heise, D.; Strecker, S.; Frank, U.; Jung, J.: Erweiterung einer Unternehmensmodellierungsmethode zur Unterstützung des IT-Controllings. In (Bichler, M. et al. Eds.): Proceedings of the Multikonferenz Wirtschaftsinformatik 2008, Berlin, 2008; pp. 1017-1028.
- [Ju07] Jung, J.: Entwurf einer Sprache für die Modellierung von Ressourcen im Kontext der Geschäftsprozessmodellierung. Logos, Berlin, 2007.
- [Ju08] Jureta, I. J.; Faulkner, S.; Schobbens, P. Y.: Clear justification of modeling decisions for goal-oriented requirements engineering. In: Requirements Engineering 13 (2008), pp. 87-115.
- [KN92] Kaplan, R.; Norton, D.: The balanced scorecard – measures that drive performance. In: Harvard Business Review 70 (1992) 1, pp. 71-79.
- [Ki08] Kirchner, L.: Eine Methode zur Unterstützung des IT-Managements im Rahmen der Unternehmensmodellierung. Logos, Berlin, 2008.
- [Kr05] Kronz, A.: Management von Prozesskennzahlen im Rahmen der ARIS-Methodik. In (Scheer, A.-W. et al. Eds.): Corporate Performance Management. Springer, 2005, pp. 31-44.
- [Kü06] Kütz, M.: IT-Steuerung mit Kennzahlensystemen. dpunkt, Heidelberg, 2006.
- [LO06] Liebetruth, T.; Otto, A.: Ein formales Modell zur Auswahl von Kennzahlen. In: Controlling 18 (2006) 1, pp. 13-23.
- [Lu04] Luoma, J.; Kelly, S.; Tolvanen, J. P.: Defining Domain-Specific Modeling Languages: Collected Experiences. In: Proceedings of the OOPSLA Workshop on Domain-Specific Modeling (DSM'04), Vancouver, 2004.
- [LC91] Lynch, R. L.; Cross, K. F.: Measure Up!: Yardsticks for Continuous Improvement. Wiley, Blackwell, 1991.
- [MNS07] Mansmann, S.; Neumuth, T.; Scholl, M.: Multidimensional Data Modeling for Business Process Analysis. In: Proceedings of the ER 2007, Auckland, New Zealand, 2007; pp. 23-38.
- [Ne95] Neuscheler, F.: Ein integrierter Ansatz zur Analyse und Bewertung von Geschäftsprozessen. Forschungszentrum Karlsruhe, Karlsruhe, 1995.
- [OM07] Object Management Group: Unified Modeling Language Infrastructure. 2007. URL: <http://www.omg.org/docs/formal/07-11-04.pdf> (2008-07-28).
- [OGC07] Office of Government Commerce (Eds.): ITIL – Service operation. The Stationery Office, London 2007.

- [Po07] Pourshahid, A.; Chen, P.; Amyot, D.; Weiss, M.; Forster, A.: Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools. In: Proceedings of the 10th Workshop of Requirement Engineering.(WERE'07), Toronto, 2007; pp. 80-91.
- [Re06] Reichmann, T.: Controlling mit Kennzahlen und Management-Tools: Die systemgestützte Controlling-Konzeption. Vahlen, München, 2006.
- [Sa99] Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. In: Advances in Database Technologies (LNCS 1552). Springer, 1999, pp. 105-116.
- [Si90] Siegwart, H.: Kennzahlen für die Unternehmungsführung. Haupt, Bern, 1990.
- [So05] Son, S.; Weitzel, T.; Laurent, F.: Designing a Process-Oriented Framework for IT Performance Management Systems. In: Electronic Journal of Information Systems Evaluation 8 (2005) 3, pp. 219-228.
- [vL01] van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. In: 5th IEEE International Symposium on Requirements Engineering (RE 2001), 27-31 August 2001, Toronto, Canada. IEEE Computer Society, 2001, p. 249.
- [vL03] van Lamsweerde, A.: From System Goals to Software Architecture. In (Bernardo, M., Inverardi, P. Eds.): Formal Methods for Software Architectures. Springer, 2003, pp. 25-43.
- [Wa01] Wall, F.: Ursache-Wirkungsbeziehungen als ein zentraler Bestandteil der Balanced Scorecard: Möglichkeiten und Grenzen Ihrer Gewinnung. In: Controlling 13 (2001) 2.
- [We08] Wetzstein, B.; Ma, Z.; Leymann, F.: Towards Measuring Key Performance Indicators of Semantic Business Processes. In (Abramowicz, W., Fense, D. Eds.): Proceedings of the BIS, Innsbruck, Austria, 2008; pp. 227-238.
- [Yu95] Yu, E. S.-K.: Modelling strategic relationships for process reengineering. 1995.
- [ZE89] Zentralverband der Elektrotechnischen Industrie: ZVEI-Kennzahlensystem. Sachon, Mindelheim, 1989.

Business Process Compliance Checking: Current State and Future Challenges*

M. El Kharbili, A.K. Alves de Medeiros, S. Stein and W.M.P. van der Aalst
IDS Scheer AG, Altenkesseler Str. 17, D-66115 Saarbrücken, Germany.
E-mails: {marwane.elkharbili, sebastian.stein}@ids-scheer.com
Eindhoven University of Technology, P.O. Box 513, 5600MB, Eindhoven.
E-mails: {a.k.medeiros, w.m.p.v.d.aalst}@tue.nl

Abstract: Regulatory compliance sets new requirements for business process management (BPM). Companies seek to enhance their corporate governance processes and are required to put in place measures for ensuring compliance to regulations. In this sense, this position paper (i) reviews the current work in the context of BPM systems and (ii) suggests future directions to improve the current status. During the literature review, techniques are classified as supporting *forward* or *backward* compliance. The latter is a post-execution compliance (i.e. based on execution histories of systems) and the former takes place at design- or run-time. In a nutshell, this position paper claims that four main aspects need to be incorporated by current compliance checking techniques: (i) an integrated approach able to cover the full BPM life-cycle, (ii) the support for compliance checks beyond control-flow-related aspects, (iii) intuitive graphical notations for business analysts, and (iv) embedding semantic technologies during the definition, deployment and executions of compliance checks.

1 Introduction

Complying to regulations of all sorts is usually needed for various purposes. These range from ensuring that specific norms are met (e.g. quality standards such as ISO9000:2005¹) to proving correct implementation of internal controls imposed by active legislations (e.g. SOX Sec.404²) [SGN07]. BPM is an integrated approach to both managing business and governing underlying IT layers supporting it [Wes07]. Therefore, ensuring the compliance of business processes (BPs) in companies is a crucial feature for a BPM system. This position paper provides an overview of the state-of-the-art efforts in the BPM community for supporting compliance checking of BPs and based on this, discusses aspects that still need to be addressed by future research. The remainder of this paper is organized as follows. Section 2 reviews the related work in for compliance checking of BPs. Section 3 provides an outlook on the current state of research and identifies points that still need to be addressed. Finally, Section 4 concludes this paper and indicates next steps of our work.

*Acknowledgements: Our thanks go to the EU commission for supporting our research within the SUPER project (www.ip-super.org).

¹International Organization for Standardization: www.iso.org.

²The sarbanes-Oxley Act: www.soxlaw.com.

2 State-of-the-Art

Our review of related work classifies them into two approaches: *forward compliance checking* and *backward compliance checking*. This classification basically indicates if the techniques have a *pro-active* or *reactive* approach for compliance checking. *Forward* compliance checking techniques target the verification of rules during *design* time or *execution* time. Therefore, these techniques can prevent the actual execution of non-compliant behavior. Subsection 2.1 contains our review for these kind of techniques. *Backward* compliance checking techniques can *detect* that non-compliant behavior has taken place by looking at the history of BP instances' execution and are unable to prevent actual non-compliant behavior. Subsection 2.1 describes the main techniques following this approach.

2.1 Forward Compliance Checking

2.1.1 Design-Time Compliance Checking (DTCC)

These techniques have a preemptive nature and aim at guaranteeing that process instances will be regulatory compliant. In this sense, some approaches guide the user during modelling phase to limit non-compliance of deployed models, while others use techniques like model checking to verify certain properties in already designed (but not yet deployed!) models. Ghose et al. [GK07] propose an approach based on the so-called *compliance patterns* (i.e., pre-defined BP models for which compliance to regulations has been proven). The main idea is to compute the deviation of a given BP model to a certain compliance pattern. Governatori et al. [PR] view BPs as social interaction processes and present a framework for managing the compliance of contractual relationships in BPs. For this, deontic assignments are defined in a multi-modal logical framework in the form of policies/rules. In a following work, Sadiq et al. [GG06, MOS06] present an approach to formalize contract documents and those aspects of BPs that relate to these business contracts. For this purpose, the semantics of business contracts and their violations are described using a specialized logic, the Formal Contract Language (FCL) [GZ05]. Furthermore, the authors have shown how these formal specification of contracts can be used to generate compliant processes. Namiri et al. [NS08] presents a formalization for definition BP compliance checking that relies on control patterns. Control patterns constitute a generic and reusable solution to a specific problem and, therefore, can be used to ensure that BP models containing them are regulatory compliant. In [NS07a], the authors add a semantic layer to the BPM stack in which process instances are interpreted according to a defined set of controls. The actual implementation (e.g. database procedures, temporal logic or rules) of the controls is independent from the modelling. In [SN07], the authors show why automation and semantic enactment are necessary for effective BP compliance. They formalize modelling of controls for compliance by motivating the need for logics that are stronger than standard deontic logic. Finally, the work by Schmidt et al. [SBO07] is one of the rare semantic approaches to BP compliance where a compliance ontology is designed and proposed to be integrated in BP models.

2.1.2 Run-Time Compliance Checking (RTCC)

These techniques target executable BP models and, consequently, depend on the BP execution architecture and mechanisms. They typically work by annotating BP models or steps with atomic compliance assertions that are destined to be either used by compliance checking engines for verification or at later stages during execution. In this sense, regulations can either be defined into BP models (e.g. control flow properties such as BP anti-patterns [JK07] which seek to achieve better quality of processes or organizational properties such as Separation of Duty (SoD)), or they can require run-time information (e.g. quality assertion enforcement while executing a BPEL process such as in [DF06]). The authors of [LMX07] identify the need for separate modeling of compliance and processes. Process models are transformed from BPEL into the Pi-Calculus [Mil93] (an algebra for modelling concurrent communicating processes) and compliance rules are modeled in temporal logic using a special graphical notation. Model checking techniques are then used to formally check a process pool. The work in [NS07b] (cf. DTCC) also encompasses execution aspects of BP compliance by proposing a solution for ensuring effectiveness of controls during BP execution and a reaction strategy in case of violation. The approach in [PSSvdA07] introduces a framework in which process models are defined in a declarative way. The authors argue that constraint-based workflow models are more expressive and more flexible than procedural ones. [PFLN04] explains how technical Service Level Agreements (SLA) can be leveraged to the business level. Contracts are specified between entities participating in inter-organizational BP cooperations, and a language for use as part of a business contract architecture is defined. The work builds on an underlying policy framework for description of behavioral constraints. In [Mil05a], policy definitions are integrated into BPs and rely on BP events and transactions for run-time compliance monitoring. In fact, this work poses initial questions about architectures for process compliance monitoring integrating events and policies such as the need for a formal definition of events, event triggers and related resources, event patterns, message handling as well as state management. Additionally, business rule management systems are widely used in the industry for production rule execution. Some compliance measures can be modeled as business rules and be coupled to BP definitions (as is done in the ARIS business rules designer³).

2.2 Backward Compliance Checking

Backwards compliance checking (BCC) techniques verify if executions of BP (i.e., process instances) are in accordance with certain constraints or rules. The works in [RA08, ABD05, CMMS07, ea07] are good representatives of current approach for BCC. Rozinat et al. [RA08] has developed *conformance checking* techniques that quantify how much the behavior of a given control-flow process model matches the behavior registered in process instances of a given history log. Whenever differences (or non-compliance) are detected, the developed conformance checking techniques provide an exact indication of where the differences are. This approach has the advantage that a graphical notation is used for specifying the models. However, the provided compliance is restricted to control-flow-related constraints. Thus, no rules involving data fields or performers information can be checked.

³www.ids-scheer.com/en/Software/ARIS_Software/ARIS_Business_Rules_Designer/3747.html

Aalst et al. [ABD05] have created the *LTL Checker*, a technique based on Linear Temporal Logic [GH01]. The approach verifies if a given LTL rule holds for a set of process instances. The result is a partition of this set into two other subsets: one containing the compliant process instances and another containing the non-compliant ones. Although the rules can also contain references to data fields, performers and temporal aspects, no graphical notation is supported, making this techniques less suitable for direct use by business analysts. The works by Alberti et al. [ea07, CMMS07] combine the power of computational logics with graphical notations to specify rules. Models can be made in GOSpeL, a graphical language created by the authors. Afterwards, these graphical models are translated into SCIFF, a declarative language based on computational logic, and applied over process instances. Although mainly targeted at BCC, the approach can also be used for forward compliance checking by translating the models into the G-SCIFF logic language.

3 Outlook

None of the approaches introduced above tackles compliance checking as having the whole BPM lifecycle as a scope. Concretely this means that that compliance is a vertical concern and starts already at a strategy level and goes down BPM layers to include the design, implementation, configuration, execution and analysis layers taking into account the orthogonal organizational, both design-time (e.g. EPC) and run-time (e.g. BPEL) control and event flow, functional and data aspects. Such an observation is emphasized when seeing natural-language regulations (e.g. laws, standards, norms etc.) as the initial input of every compliance checking effort. These regulations have then different representations depending on the layer where they have to be enforced (i.e contracts language in [GZ05] or temporal logic constraints on BPEL processes in [LMX07]). A *Compliance checking framework shall cover all phases of the BPM life-cycle* because this will lead to a fully integrated approach, where the complexity of compliance architectures, algorithms and coverage is reduced.

Besides, declarative means of modeling compliance present an appealing alternative to "hard-coded" checking algorithms because they scale with change in both the compliance requirement and the BP dimensions. While relying on the definition of compliance as SWRL rules, Yip et al. [FY07] recognize the limited expressiveness of the language and propose the use of extensions. As the logic necessary to model compliance varies depending on the application case (e.g. temporal, deontic, horn logic, etc.) a *holistic approach should take this into consideration and offer the possibility to use formalisms of different degrees of expressiveness and computational complexity*. Although the need for formal modeling is clearly identified by many of the approaches listed, the complexity of the developed tools and the pre-required knowledge (e.g. about formal logics) is frequently underestimated as an adoption barrier. This observation is made in [LMX07] and pushed the authors to develop a graphical language for modeling temporal logic constraints on BPEL processes. A *graphical notation for modeling compliance that is easy to learn and use by business analysts will help endorse existing approaches, as it allows hiding the complexity of defining logical constraints formally*. Such an assertion can of course only be validated by rigorous empirical research. A similar observation can be made on the works by Alberti et al. [ea07, CMMS07]. In this regard, the work by Koehler et al. [JK07]

on anti-patterns for BP control flows deserves mention and could be built on to modeling reference compliant BP patterns in order to measure compliance by comparing BPs against the latter.

To scale with the complexity of continuously changing and ambiguous regulations as well as re-engineered BPs while keeping compliance costs reasonable, there is a need for a declarative environment for formal modelling of BP compliance taking regulations as input. Such an environment should support both inter- and intra-organizational collaborations (e.g. markets), as well as take enterprise models into consideration, and *not be only tailored towards structural or syntactical properties of BPs*. Policies as discussed by Miloševic in [Mil05b] would allow to separate between compliance management and decision solving problems (i.e. when is a state or behaviour non-compliant? which policy should be responsible for ensuring compliance? etc.) from compliance checking or enforcement problems (i.e. policy implementation, e.g. as business rules).

Another aspect of compliance are domains such as quality or security. In fact, compliance needs to be modelled differently depending on the domain it has to be enforced in. Aspects of BP compliance that are of fundamental nature such as control flow are generic problems whereas modeling compliance to the ISO27001 standard requires specific constructs to be regarded for BPs. Yip et al. [F.Y07] study compliance for the security domain and extract specific security rules semi-automatically from a security standard. In such cases, semantics play an important role. Also, an important facet in guaranteeing BP regulatory compliance is *enabling business experts to directly supervise the modelling and implementation of compliance management measures*, because requirements on BPs are best made by experts possessing the necessary knowledge of a company's business. Compliance needs to be enforced in domains of higher abstraction than what elementary BPM languages are able to express. The scope of BPs that is considered expands beyond workflows (to incorporate organizational and resource/data aspects) making it necessary to have a sound understanding of the semantics of the handled assets. *Making these semantics machine processable by means of formal conceptual modeling (e.g. ontologies) allows compliance checking to take these into account*. This observation is also made in [LGRMD08] where requirements for semantic constraint support are analyzed. This helps handling the ambiguity of natural language regulations but semantic modeling of compliance is unfortunately still highly ignored by current approaches.

4 Conclusions and Future Work

This position paper has focused on answering the following research question: How are current approaches within the BPM research area handling compliance checking? To answer this, a state-of-the-art review of research on this topic has been performed. This review indicates that current works have made a good progress in this direction but no approach is still able to cover *all* phases of the BPM life-cycle. Furthermore, most approaches seem to be yet quite far from their target end users: business analysts. We have provided an outlook explaining the four main factors that need to be incorporated by current compliance checking techniques: (i) an integrated approach able to cover the full BPM life-cycle, (ii) the support for compliance checks beyond control-flow-related aspects, (iii) intuitive graphical notations for business analysts, and (iv) embedding of semantic technologies during the definition, deployment and executions of compliance checks. There-

fore, having chosen policies and business rules as formal means of modeling compliance, our future research agenda will be centered around defining compliance checking techniques that are semantic, declarative, policy-based and cover the full BPM lifecycle.

References

- [ABD05] W.M.P. van der Aalst, H.T. de Beer, and B.F. van Dongen. Process Mining and Verification of Properties: An Approach Based on Temporal Logic. In R. Meersman et al., editor, *OTM Conferences (1)*, volume 3760 of *LNCS*, pages 130–147. Springer, 2005.
- [CMMS07] F. Chesani, P. Mello, M. Montali, and S. Storari. Testing Careflow Process Execution Conformance by Translating a Graphical Language to Computational Logic. In R. Bellazzi, A. Abu-Hanna, and J. Hunter, editors, *AIME*, volume 4594 of *Lecture Notes in Computer Science*, pages 479–488. Springer, 2007.
- [DF06] Wilhelm Rossak Daniel Foetsch, Elke Pulvermueller. Modeling and Verifying Workflow-based Regulations. In *Proceedings of the international workshop on regulations modeling and their validation and verification. REMO2V06.*, pages 825–830. CEUR-WS.org/vol-241, Luxemburg, June 2006.
- [ea07] M. Alberti et al. Expressing and Verifying Business Contracts with Abductive Logic Programming. In G. Boella et al., editor, *Normative Multi-agent Systems.*, volume 07122 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany, March 2007.
- [F.Y07] N. Parameswaran & P. Ray F.Yip. Rules and Ontology in Compliance Management. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, number 1541-7719, page 435, 2007.
- [GG06] Shazia Sadiq Guido Governatori, Zoran Milosevic. Compliance checking between business processes and business contracts. In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, pages pp. 221–232, 2006.
- [GH01] D. Giannakopoulou and K. Havelund. Automata-Based Verification of Temporal Properties on Running Programs. In *ASE '01: Proceedings of the 16th IEEE international conference on Automated software engineering*, page 412, Washington, DC, USA, 2001. IEEE Computer Society.
- [GK07] A.K. Ghose and G. Koliadis. Auditing business process compliance. In *Proceedings of the International Conference on Service-Oriented Computing (ICSO-C-2007)*, volume 4749 of *Lecture Notes in Computing Sience*, pages 169–180, 2007.
- [GZ05] G. Governatori and Milosevic. Z. Dealing with contract violations: formalism and domain specific language. In *Proceedings of the Conference on Enterprise Computing EDOC 2005.*, page 4657. IEEE Press., 2005.
- [JK07] Jussi Vanhatalo Jana Koehler. Process anti-patterns: How to avoid the common traps of business process modeling. *IBM WebSphere Developer Technical Journal.*, 28 Feb 2007.
- [LGRMD08] L.T. Ly, K. Göser, S. Rinderle-Ma, and P. Dadam. Compliance of Semantic Constraints A Requirements Analysis for Process Management Systems. In Sadiq S., Indulska M., and zur Muehlen M., editors, *Proceedings of the International Workshop on Governance, Risk and Compliance - Applications in Information Systems (GRCIS08).*, June 2008.

- [LMX07] Y. Liu, S. Müller, and K. Xu. A static compliance-checking framework for business process models. *IBM Syst. J.*, 46(2):335–361, 2007.
- [Mil93] R. Milner. The polyadic pi-calculus: a tutorial. In F. L. Bauer, W. Brauer, and H. Schwichtenberg, editors, *Logic and Algebra of Specification*, pages 203–246. Springer-Verlag, 1993.
- [Mil05a] Z. Milosevic. Towards Integrating Business Policies with Business Processes. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management*, volume 3649, pages 404–409, 2005.
- [Mil05b] Zoran Milosevic. *Towards Integrating Business Policies with Business Processes*, pages 404–409. 2005.
- [MOS06] Sadiq S. W. Milosevic, Z., J. L. Fiadeiro Orlowska, M. E. In: S. Dustdar, and A. P. Sheth. Towards a methodology for deriving contract-compliant business processes. In *Proceedings of the 4th International Conference on Business Process Management (BPM06)*, Vienna, Austria., 2006. 5-7 September, 2006.
- [NS07a] K. Namiri and N. Stojanovic. Pattern-Based Design and Validation of Business Process Compliance. In R. Meersman and Z. Tari, editors, *OTM Conferences (1)*, volume 4803 of *Lecture Notes in Computer Science*, pages 59–76. Springer, 2007.
- [NS07b] Kioumars Namiri and Nenad Stojanovic. *Pattern-Based Design and Validation of Business Process Compliance*, pages 59–76. 2007.
- [NS08] K. Namiri and N. Stojanovic. Towards A Formal Framework for Business Process Compliance. In M. Bichler, T. Hess, H. Krcmar, U. Lechner, F. Matthes, A. Picot, B. Speitkamp, and P. Wolf, editors, *Multikonferenz Wirtschaftsinformatik*. GIT-Verlag, Berlin, 2008.
- [PFLN04] J. Cole S. Gibson S. Kulkarni P. F. Linington, Z. Milosevic and S. Neal. A unified behavioural model and a contract language for extended enterprise. *Data & Knowledge Engineering*, Volume 51, Issue:5–29, October 2004.
- [PR] Governatori G. Sadiq S. Colomb R. M. Padmanabhan, V. and A. Rotolo. Process Modelling: The Deontic Way. In *The Third Asia Pacific Conference on Conceptual Modelling (APCCM 2006)*.
- [PSSvdA07] M. Pesic, M. Schonenberg, N. Sidorova, and W. van der Aalst. *Constraint-Based Workflow Models: Change Made Easy*, pages 77–94. 2007.
- [RA08] A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95, 2008.
- [SBO07] R. Schmidt, C. Bartsch, and R. Oberhauser. Ontology-Based Representation of Ccompliance Requirements for Service Processes. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management*, pages 28–39, 2007.
- [SGN07] S.W. Sadiq, G. Governatori, and K. Namiri. Modeling Control Objectives for Business Process Compliance. In G. Alonso, P. Dadam, and M. Rosemann, editors, *BPM*, volume 4714 of *Lecture Notes in Computer Science*, pages 149–164. Springer, 2007.
- [SN07] Governatori G. Sadiq, S. and K. Namiri. *Modeling Control Objectives for Business Process Compliance*, pages 149–164. Lecture Notes in Computer Science. Springer, 2007.
- [Wes07] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag, Berlin, 2007.

An Engineering Approach to Enterprise Architecture Design and its Application at a Financial Service Provider

Stephan Aier*, Stephan Kurpjuweit*,
Otto Schmitz**, Jörg Schulz**, André Thomas**, Robert Winter*

* Institute of Information Management
University of St. Gallen
Müller-Friedberg-Straße 8
CH-9000 St. Gallen
{stephan.aier, stephan.kurpjuweit, robert.winter}@unisg.ch

** Deutsche Leasing AG
Fröhlingstraße 15-31
DE-61352 Bad Homburg v.d. Höhe
{otto.schmitz, joerg.schulz, andre.thomas}@deutsche-leasing.com

Abstract:

In analogy to classical engineering disciplines, this contribution discusses characteristics and requirements of an engineering approach to enterprise architecture design and proposes components and a top-level structure of an approach to address these requirements. The proposed components can partially be realized by existing work; partially they lay out the research path towards a mature engineering discipline of EA design. Core components of the proposed approach have been applied and evaluated at Deutsche Leasing AG, a globally operating financial service provider based in Germany.

1 Introduction

Organizations¹ are subject to constant evolution: changing business models, technological innovations, increasingly individualized products and services, the globalization of sourcing, sales, and operations as well as deregulation are only a few drivers of transformation [RWR06]. Due to the differences in impact, organizational change can be distinguished into incremental change (optimization) and fundamental change (transformation). While most functional business administration methods such as finance or human resources provide support for optimization, innovative and fundamental change requires systematic approaches to design, plan and implement the transformation [Wi08]: It is essential for an organization to systematically analyze the impact of upcoming changes. A prerequisite to achieve this is a thorough understanding and therefore an explicit documentation of all structures of interest as well as their relationships. Struc-

¹ In the following *organization* refers to companies, public administration, etc. and comprises the entirety of business related and IT related components of an organization. Therefore non-IT related components are referred to as business components.

tures of interest typically include product structures, business processes and structures, the relationship between business objects and data structures, application landscapes and software architectures as well as the supporting IT infrastructure systems and technologies.

For many organizations enterprise architecture management (EAM) is an important means to ensure the correct and up-to-date documentation of and the alignment between the various structures [AW09]. Enterprise architecture is defined as the fundamental structure of an organization from a holistic perspective as an aggregate model [WF07]. While there is a broad variety of EA literature focusing on evaluation [Sc04] and generalization [If99] of EA frameworks or discussing EA modeling [ABB07], only few publications address how EA should be managed, designed and analyzed systematically to facilitate innovative and fundamental change.

In this contribution we analyze mature engineering disciplines to characterize the role that EA should play in the systematic transformation of organizations. Based on this positioning, we derive requirements for an engineering approach to EA design (section 2). The scope of EA documentation is discussed which is necessary to fulfill such requirements (section 3), and components as well as a top-level structure of an appropriate EA design approach are proposed (section 4). Such components can partially be realized by existing work; partially they lay out the research path towards a mature engineering discipline of EA design. Core components of the approach have been applied and evaluated according to a design science approach at Deutsche Leasing AG, a globally operating financial service provider based in Germany (section 5).

2 Characteristics of Classical Engineering Disciplines and Requirements for an Engineering Approach to EA Design

Shaw analyzed the development of classical engineering disciplines [Sh90]. She found that engineering disciplines produce cost efficient solutions for relevant problems by using scientific knowledge in the artifact design process in service to society. These aspects are now further characterized:

- 1) “Cost efficient solutions”: Engineering does not only imply the construction of suitable solutions, but emphasizes reasonable handling of given resources and conditions.
- 2) “For relevant problems”: The constructed solution addresses problems relevant in practice.
- 3) “By using scientific knowledge”: The construction process is comprehensible and traceable based on scientific construction languages, methods, and frameworks so that the solutions will most likely fit the requirements.
- 4) “In service to society”: The engineer acts in a responsible way by providing useful innovations to society and environment.

The following subsections outline how classical engineering disciplines address these characteristics and which requirements for an engineering approach to EA design can be derived.

2.1 Standardized Construction Plans and Construction Languages

Mature engineering disciplines produce a high level construction plan (architecture) of the artifact under construction.² This plan depicts the main components of the artifact and the relationships between these components. The architecture explains how the artifact achieves the desired behavior. All mature engineering disciplines have developed standardized construction languages for architectural descriptions. For example, in mechanical engineering detailed standards exist on how to document construction plans [GMS08].

EA can be regarded as the central construction plan for transformation in a “business-to-IT” approach. The EA describes the main business and IT components as well as their relationships and explains how these components interact. Despite some standardization and unification endeavors like TOGAF [Op07] and GERAM [If99], no comparably accepted and powerful standard language to design, communicate and teach EA designs exists. While TOGAF describes, how to develop IT related aspects of an EA, it does neither comparably comment on business related structures, nor does it specify a standardized construction language. GERAM defines a meta-framework to relate EA frameworks like TOGAF to each other. However, GERAM remains abstract to a large extend and does not provide implementable guidance to EA description and development.

2.2 Reuse of Engineering Knowledge

Classical engineering disciplines distinguish innovative construction from routine construction.³ Innovative constructions address new solutions while routine construction involves reusing existing solution patterns for known problems [Zw48].

Routine construction is the typical design task in classical engineering disciplines, while innovation is rather rare. To make the construction process as efficient as possible, the collection, organization, and packaging of knowledge is necessary to make it available to less experienced engineers. All disciplines found appropriate media for this knowledge transfer, e.g. engineering handbooks [ABS07; DKB94] and tool support for collaborative engineering [MKW93].

² Some engineering disciplines including civil engineering and software engineering use the “architectural blueprint” or “architectural design” (short “architecture”) as central construction plan. In the following the term “architecture” is used synonymously for the central construction plan of all engineering disciplines.

³ Please note that the distinction between innovative construction and routine construction is orthogonal to the distinction between optimization and transformation introduced in section 1. For example, organizational transformation can be achieved by means of both routine construction and innovative construction.

Documentation standards which foster the reuse of existing architectural solutions (e.g., architectural patterns or styles) to known problem classes must be found also for EA. Additionally, like in other engineering disciplines there will be no one-size-fits-all solutions. Hence it is important to specify the context factors for which solutions are applicable. It is necessary to adapt the generic architectural solutions to specific situations (e.g., company-specific requirements) and to integrate partial solutions into a complete architectural design.

2.3 Division of Labor

Besides structuring the system to be designed, the construction plan is also used to structure the development process: the components of a system are typically constructed in separate teams and then assembled in order to become a whole according to the architecture. The division of labor during the construction process is a core feature of classical engineering disciplines, since it is the only way to construct complex systems in large teams.

In the context of organizational transformation, division of labor takes place in individual transformation projects which are typically carried out independently in disjoint teams. The role of EA is to ensure overall consistency of project results.

2.4 Systematic Design and Analysis

Designing the architecture is a critical task in engineering, because it involves the transformation of requirements (problem space) into a high level blueprint of the system to be designed (solution space). Architecture design thus involves fundamental design decisions which impact the quality attributes of the system under construction (e.g. Which changes to the system can be made easily, which not? What is the system's performance?). Typically the requirements of different groups of stakeholders must be observed in a system's design. These requirements often contradict each other, so that tradeoff decisions must be made [KKC00].

Great attention must be paid to architecture and typically the most experienced engineers are involved in the architecture design process. By involving experts as well as complex analysis frameworks, engineers seek to ensure the quality of the architectural blueprint so that the architecture satisfies all relevant requirements.

EA is also the result of design decisions which determine fundamental characteristics of the organization such as strategic positioning, business process efficiency and effectiveness, business/IT alignment, and information systems capabilities. Indirectly, EA therefore implies an organization's capability to rapidly launch new products, to adapt to changed regulations, or to exploit business potentials of IT innovations.

Concrete requirements of internal and external stakeholders must be the starting point for EA design. There are different types of requirements. One type focuses on the functional development of the organization. Examples are the development of new markets and

sales channels or business process outsourcing. Another type focuses on the optimization of current structures, e.g. by consolidating redundant structures or reusing existing resources to improve flexibility and to prepare the organization for future changes.

The requirements tend to involve tradeoffs which must be incorporated in an overall architectural design. Architecture analysis models should be available that determine the prospective quality attributes of the organization. Only such analysis models allow for the transformation of organization with predictable properties and are thus a prerequisite for an engineering discipline of EA design.

3 Width and Depth of Enterprise Architecture

Enterprise architects often have difficulties answering the following questions: Which objects and which relationships between these objects should be documented? How detailed should the documentation be? From the engineering perspective discussed above and from our experience gained in various EA projects, the following heuristics can be stated.

3.1 Criterion of Width

Concerns of a large and diverse group of stakeholders must be addressed in organizational transformation projects. These include systems architects, project managers, sponsors, implementers, and change agents who are participants in the transformation project, as well as customers, employees, managers, system operators, outsourcing partners, or the workers' council which are stakeholders concerned with the properties of the organization. For software and information systems engineering, catalogs of – mostly technical – concerns have been published [Al00; CE00]. These include quality concerns like system performance [Al00] as well as design related concerns like the structure and representation of data [CE00]. In the context of organizational transformation, also business concerns like business service implementation and business process efficiency should be considered [DIL04]. Based on the definition suggested by [SR01], we define a concern as a matter of interest in an organization. Accordingly, a stakeholder is defined as a person who has a certain concern [KW07].

The EA documentation must address the information needs of its stakeholders. These information needs can be derived from the stakeholders' concerns. Following the criterion of width, all objects and relationships required to address these information needs must be part of the EA documentation; the sum of all information needs therefore determines the maximum scope of EA documentation. Consequently, the scope of EA must thus be broader than solely the IT architecture of an enterprise. For instance, [WF07] identify five enterprise architectural domains which are also the basis of the work presented in this contribution: business architecture, process architecture, integration architecture, software architecture, and technology architecture.

3.2 Criterion of Depth

When the EA documentation is solely based on the criterion of width, chances are high that a large number of detailed design and implementation structures or detailed inventories of single artifact types are included. Therefore another criterion must be applied to sort out detail structures which do not have architecture impact.

EA documentation must depict how architectural strategies have been applied to address the requirements. Architectural strategies affect the design of the overall system or of a group of congenerous objects such as all core business processes, all domain-spawning data flows, or all products which are distributed over a certain channel. Structures which only focus on implementation details of one object and which are only relevant for this object, should not be considered part of the EA. Exceptions might however be acceptable in certain situations, e.g. in order to support specific concerns of a key stakeholder.

Whether an object should be considered part of the EA or not is indicated by the impact that a change to an object of that type has on other EA objects. If a change to an object does not influence other EA objects, it should most likely not be considered part of EA. Following the idea that EA is the blueprint for transformation projects, problems can arise from over-specifying design decisions which should better be made in the context of an individual project. Therefore, details such as class structures, detailed data structures, mapping information of network adaptors to servers, workflow specifications, or product variant configuration should usually not be considered part of the EA. Figure 1 illustrates this “broad and aggregate” understanding of EA.

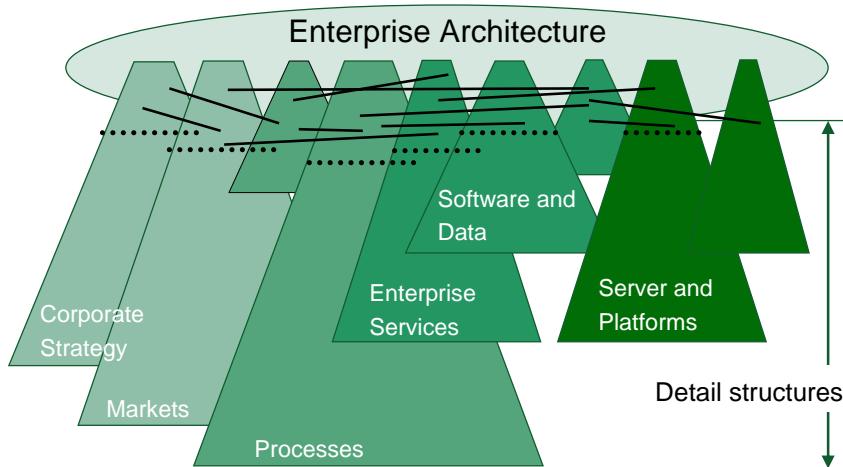


Figure 1: Enterprise Architecture is Broad and Aggregate

In two cases it can nevertheless be useful to include detail structures in the EA documentation. In both cases, changes to the detail structure may cause changes to other EA objects – which means that the heuristic introduced above remains valid:

- 1) If relationships to other design objects occur at a detailed level: Examples can be found when deploying single software components on servers or assigning sub-goals of a balanced scorecard to the responsible business units. A detailed relationship (e.g. between software components and servers) can always be considered at an aggregate level (e.g. between applications and server platforms). Detail structures should only be considered part of the EA when they codify design decisions that impact properties of the overall system. This is for instance true for the deployment of software components on servers, since the design of these relationships might have considerable impact on the ability of the organization to keep its business running in case of EDPC failures. An example for a relationship on detailed level without significant impact on the properties of the overall system is the assignment of application functions to activities of business processes. In this case, the aggregate relationship between applications and business processes delivers sufficient information to answer all architectural questions, while the detailed documentation is usually misleading.
- 2) If objects on a detailed level can be reused in multiple objects: The detail level should only be taken into account, if reuse has significant impact on the properties of the overall system. This is for instance the case when reusing product components as part of a platform strategy. It is generally not the case when reusing software libraries in multiple software components.

Moreover, it cannot be recommended to include many objects of a detail structure that all have similar topological relationships within the architecture. This is for example the case when considering all client computers of an enterprise (as an inventory).

3.3 Pragmatic Criterion

Organizations are subject to constant change. Therefore EA models need to be updated regularly. Many projects show that continuous maintenance efforts lead to high costs. For that reason it must be considered if the benefits from covering a stakeholder's concern exceed the costs of gathering the information continuously. Quantifying the costs and benefits of information needs is not trivial [Sc05]: Benefit analyses often result in "reverse considerations" (What if we did not have this information?). Cost drivers are the type of information, its origin, necessary extraction efforts and frequency.

EA serves as a high level blueprint for the transformation of an organization. High change frequencies typically indicate that the level of aggregation is too low. From our experience, in most of these cases it is sufficient to use more aggregate structures (as proposed in the criterion of depth).

4 Components and Top-Level Structure of an Engineering Approach to EA Design

The proposed approach to EA design contains various components which collectively address requirements of EA design as an engineering discipline. The components can partially be realized by existing work; partially they lead to research needs.

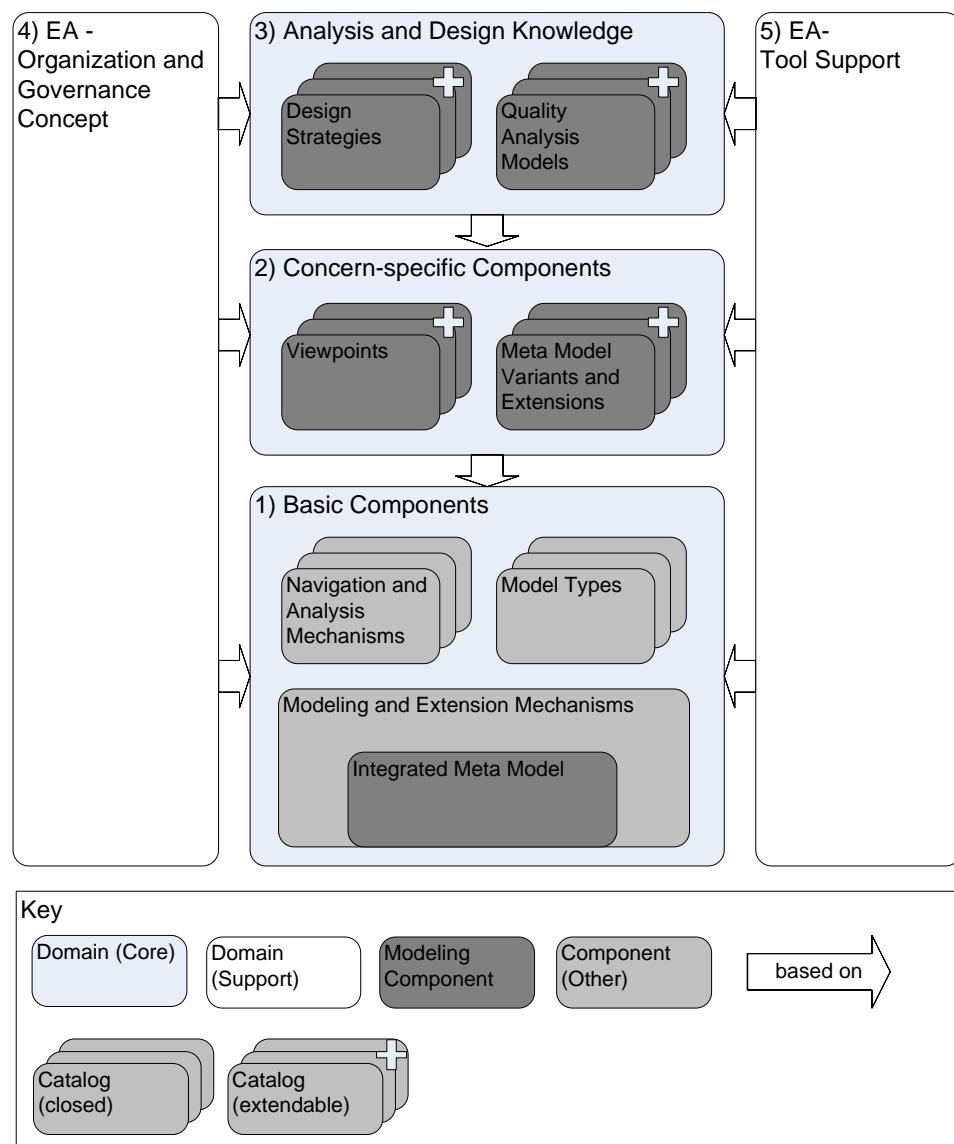


Figure 2: Top-level structure of the proposed engineering approach to EA design

Figure 2 illustrates the top-level structure the proposed approach which is further described in the following subsections.

4.1 Basic Components

Basic components contain the mechanisms for EA modeling, analysis and design.

- *Integrated Meta Model:* The integrated meta model specifies the vocabulary to consistently describe all domains of EA. In the context of the proposed approach a meta model is a model of a modeling language [Bé05; Kü06]. It specifies the available object types, relationship types as well as consistency constraints on their usage [Si99]. The meta model of the approach is based on the architectural domains introduced in section 3.1. Other EA frameworks and meta models may be applied to realize this component, as long as they satisfy the criteria stated in section 3.
- *Modeling and Extension Mechanisms:* A meta model independent description language encapsulates basic structuring and modeling mechanisms that have turned out to be useful to EA modeling. These include hierarchical refinement of objects using “part-of” and “is-a” relationships. Necessary extensions or adaptations of the integrated meta model (e.g. to company-specific requirements) can be done systematically by means of extension mechanisms [KHO07].
- *Navigation and Analysis Mechanisms:* Generic mechanisms support the analysis of the structural model information using predefined viewpoints and ad-hoc queries.
- *Model Types:* Model types represent the model information appropriately to be useful for the stakeholders. Examples for generic model types include matrix diagram, dependency diagrams, list reports, or spider web diagrams [BEL08; WBF07]. Viewpoints are stakeholder-oriented instances of model types (e.g., a matrix diagram showing the relationship between applications and business processes).

4.2 Concern-specific Components

Concern-specific components apply the generic mechanisms defined in the basic components to address specific stakeholder concerns.

- *Meta Model Variants and Extensions:* Extensions of the integrated meta model support the application of the engineering approach in specific contexts (e.g. industry, company size) and in specific projects (e.g. business driven changes, IT driven changes). The meta model variants and extensions are created by means of extension mechanisms.

- *Viewpoints*: A viewpoint captures an information need of one or more stakeholders (derived from their concerns) and defines an EA view which addresses this information need. The EA view is determined by its model type, the underlying meta model, and an analysis mechanism (specifying how the view is created on basis of a model that corresponds to the meta model) [Ba04]. The viewpoints are organized as an extendable catalog from which appropriate viewpoints can be selected and assembled to a company-specific or project type-specific solution.

4.3 Components of Analysis and Design Knowledge

Components of analysis and design knowledge help to keep record of the engineers' knowledge.

- *Design Strategies*: Proven design solutions (architectural strategies and principles) for known problems are organized as a knowledge repository [BEL08].
- *Quality Analysis Models*: Quality attribute analysis models relate architectural design decisions to quality attribute metrics.

The proposed approach can be understood as interface between organizational concepts for EA design and underlying EA tools: On one hand, the approach defines requirements for software tools and gives guidance how to use them. On the other hand, the approach serves as a “service layer” for EA organization and governance concepts, which anchor EA design within the company by defining processes, organizational roles and interfaces to other activities like strategic planning.

5 Applying the Engineering Approach to EA Design at a Financial Service Provider

The engineering approach to EA design introduced before has been applied and evaluated at Deutsche Leasing AG, a globally operating financial service provider based in Germany. The following sections describe this case study and the lessons learned. The description puts a special focus on how the components introduced in the previous section have been instantiated.

Goal of the project was the initiation of EAM to ensure transparency of all structures in context of a comprehensive outsourcing strategy and to gain indications for consolidation potentials. Therefore a company-specific meta model had to be defined and supported through customization of an EA tool.

5.1 Enterprise Architecture Modeling

As discussed before, there is no standard EA modeling language. Thus, Deutsche Leasing AG had to select an existing modeling approach. Though other approaches could have been applied as well, the St. Gallen Core Business Meta Model (CBMM) [ÖWH07] has been selected as *Integrated Meta Model* for the following reasons: (1) It covers all EA domains from business architecture to technology architecture [WF07]. (2) It conveys the documentation of EA at a high level of abstraction. (3) It can be adapted to the specific requirements. (4) EA tool support for the model is available.

In a series of workshops, the CBMM was validated against the concerns and information needs of the stakeholders within Deutsche Leasing AG. Additional object types (e.g. application environments, virtual servers) and relationships were identified to address the information needs of the stakeholders. Other design objects were eliminated. These activities led to a CBMM *Meta Model Variant/Extension*. This variant can be seen as an adaptation of the CBMM for the project type “IT master plan” with special focus on the IT-related domains of EA. The width and depth of the EA model was a major issue in these discussions. As pragmatic answer to these questions the three criteria presented in section 3 were applied.

Due to its complexity, the meta model is not suited for discussions with stakeholders. Therefore it has been divided into manageable groups of semantically related objects that are typically designed together (“meta model fragments”). For each of these meta model fragments, a model type has been defined. Figure 3 illustrates example models of the model types that were applied at Deutsche Leasing AG.⁴

The following *Modeling Mechanisms* have turned out to be useful in the context of Deutsche Leasing AG. First, the refinement of core artifacts (products, business processes, applications and software components, servers, data objects) into two refinement dimensions: The “part of” dimension has been used to model the internal structure of objects down to a refinement level as specified by the criterion of depth (cf. section 3.2). The “is a” dimension has been used to include reference models (especially of business processes) and their instantiation within different business units. Second, domain clustering has been applied to aggregate and align design objects at a high level of abstraction.

⁴ Although it neither intended nor possible to read the details of each model, these screenshots should give an idea of the model types employed and the various possibilities of visualization. The same is true for figure 4.

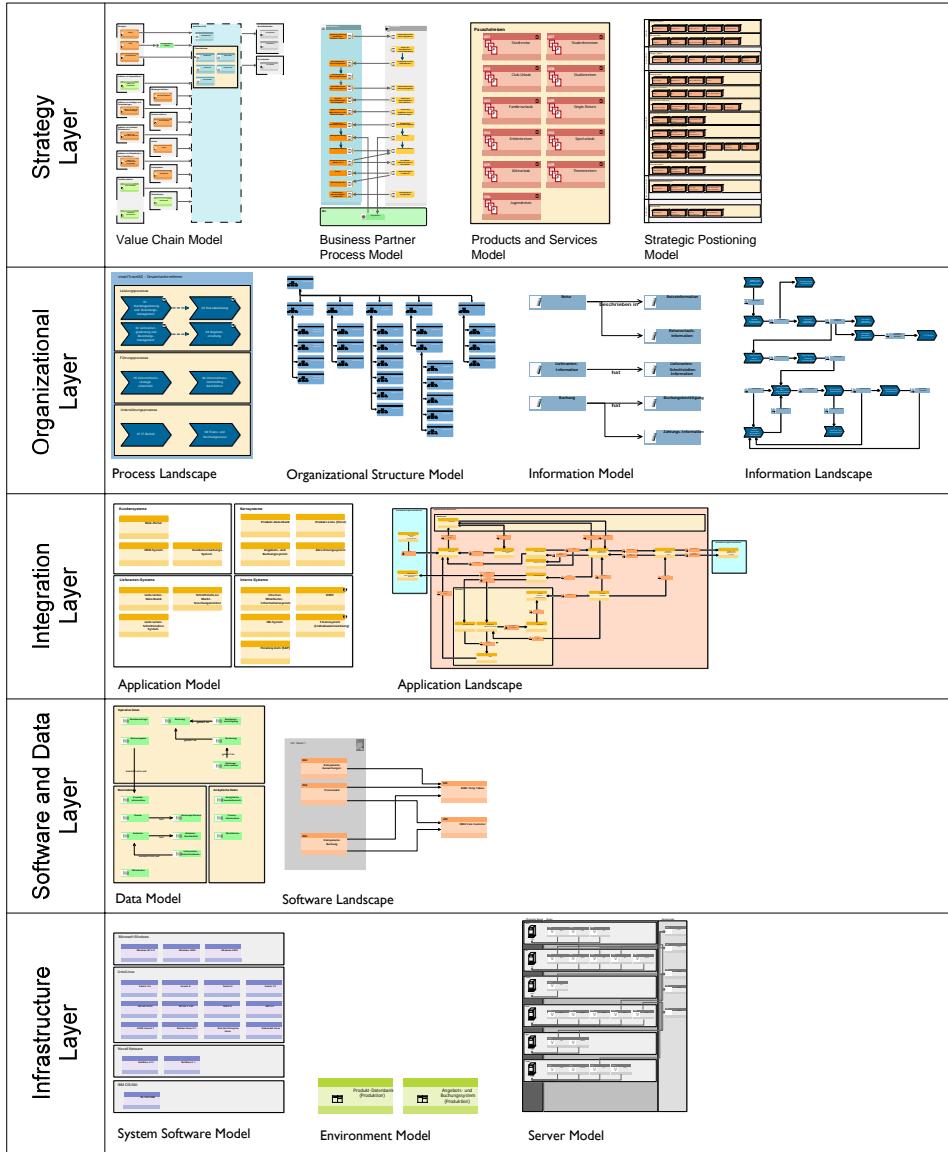


Figure 3: Model types applied at Deutsche Leasing AG

5.2 Enterprise Architecture Analysis

To analyze EA with respect to the concerns of the stakeholders, a set of pre-defined queries (*Viewpoints*) of the structural model information has been defined. Each query answers a question which has been developed in collaboration with a stakeholder. In detail the following questions are answered: How does a business unit interact with its

business partners? What is the strategic position of different products? Which applications are used along a process in different organizational units? Which applications are used along a process for different products? Which organizational units participate in which business processes? How are business processes refined hierarchically? How are applications deployed on servers? How are business objects represented as data objects?

Different model types have been applied in the viewpoints: two-dimensional matrixes (relating two objects types by means of a cross in the matrix cells), three-dimensional matrixes (relating instances of two object types by means of a third object type, figure 4), reports (lists of selected objects together with attributes of these objects), dependency diagrams, and cluster maps. *Navigation and Analysis Mechanisms* define how the viewpoints are derived from the instance of the integrated meta model.

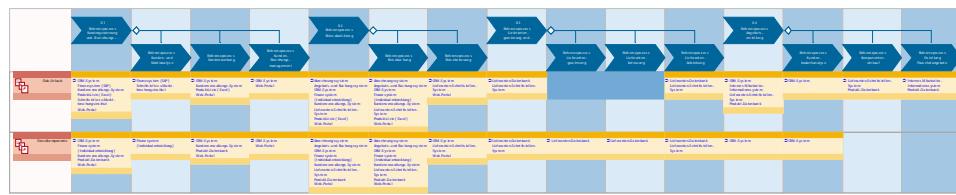


Figure 4: Three-dimensional matrix analysis showing product lines (y-axis), business processes (x-axis) and applications which are used in a process for a product line (matrix cells)

Quality Analysis Models help to interpret the viewpoints in the light of quality attributes. In the project at hand, early experiences with quality models could be made. For example, metrics on media breaks along the process, business continuity considerations in case of server failures, and consolidation potentials of applications could be considered. Especially matrix analyses have turned out to be a valuable tool to foster and rationalize the communication between the IT unit and the business units.

5.3 Analysis of Potential and Mapping with Design Knowledge

The analysis results are a basis to apply architectural *Design Strategies*. In the project at hand, early experiences with simple design strategies were made. For example, the analysis of the application landscape revealed manual data flows between applications for which automatic interfaces would be valuable. By means of a mapping between business processes and the supporting applications, redundancies and gaps were identified.

5.4 Enterprise Architecture Organization and Tool Support

Deutsche Leasing AG has implemented a decentralized approach to EA management. The EA domains are maintained by different stakeholder groups. For example, the business units maintain their business processes and products, while application managers maintain the software architecture of their applications. This decentralized maintenance

concept implements the engineering principle of division of labor. EA maintenance processes and roles were defined in an *EA Organization and Governance Concept*.

The *Integrated Meta Model, Modeling Mechanisms, Model Types, Navigation and Analysis Mechanisms, and Viewpoints* have been implemented using the *EA Tool ADOben®⁵*. The screenshots shown in Figure 3 and 4 are created using this tool.

6 Conclusions and Future Work

The application of the engineering approach to EA design at Deutsche Leasing AG has led to the following conclusions from a practitioners' point of view: The EA should be positioned as a planning tool, not as a tool focused on operative tasks (like for example a CMDB system). To achieve this, the three criteria defining EA scope have proven to be valuable. The criterion of width requires that the EA meta model and the viewpoints are developed in close collaboration with stakeholders of the EA. To get the buy-in of the stakeholders, the introduction of EAM should be taken as a chance to revise the documentation processes within the organization in order to ensure that the EAM organization concept is integrated seamlessly and does not cause additional work load for the stakeholders.

From a research point of view, the top-level structure of the proposed approach leads to the following research opportunities: In particular the components of analysis and design knowledge (figure 2) must be fleshed out. First, EA design strategies must be gathered and packaged to provide hands-on guidance for enterprise architects. Though EA frameworks provide a good starting point on what should be considered part of an EA, they do not provide concrete guidance how to address concrete requirements by making architectural design decisions. Complementary, quality analysis models must be developed that relate the structural EA design decisions to the quality attributes of the organization under construction.

7 References

- [ABB07] Arbab, F. et al.: Integrating Architectural Models. Symbolic, Semantic and Subjective Models in Enterprise Architecture. In: Enterprise Modelling And Information System Architectures 2 (2007) 1, pp. 40-56.
- [ABS07] Avallone, E. A.; Baumeister, T.; Sadegh, A.: Marks' Standard Handbook For Mechanical Engineers. 11th edition, McGraw-Hill Professional 2007.
- [Al00] Aldrich, J.: Challenge Problems for Separation of Concerns. In: Proceedings, OOPSLA 2000 Workshop on Advanced Separation of Concerns, Minneapolis, USA 2000.
- [AW09] Aier, S.; Winter, R.: Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. In: Business & Information Systems Engineering 51 (2009, forthcoming) 2.
- [Ba04] Bayer, J.: View-based Software Documentation. Ph.D. Thesis, Universität Kaiserslautern, Kaiserslautern 2004.

⁵ ADOBen® is a registered trademark of BOC AG, additional information: <http://adoben.iwi.unisg.ch/>.

- [Bé05] Bézivin, J.: On the Unification Power of Models. In: Software and System Modeling 4 (2005) 2, pp. 171-188.
- [BEL08] Buckl, S. et al.: Enterprise Architecture Management Pattern Catalog. Munich 2008.
- [CE00] Czarnecki, K.; Eisenecker, U.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley 2000.
- [DIL04] ter Doest, H. et al.: Viewpoints Functionality and Examples. Technical Report TI/RS/2003/091, Telematica Instituut 2004.
- [DKB94] Dubbel, H.; Kuttner, K. H.; Beitz, W.: Dubbel. Handbook of Mechanical Engineering. Springer, Berlin 1994.
- [GMS08] Giesecke, F. E. et al.: Technical Drawing. 13th edition, Pearson Education, Denver, CO 2008.
- [If99] IFIP-IFAC Task Force: GERAM: Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3.
<http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/GERAMv1.6.3.pdf>, Date of Access: 03.12.2007.
- [KHO07] Kurpjuweit, S.; Höning, F.; Osl, P.: Metamodell-Erweiterungsmechanismen. St. Gallen 2007.
- [KKC00] Kazman, R.; Klein, M.; Clements, P.: ATAM: Method for Architecture Evaluation. Technical Report, Software Engineering Institute Carnegie Mellon University 2000.
- [Kü06] Kühne, T.: Matters of (meta-)modeling. In: Software and Systems Modeling 5 (2006) 4, pp. 369-385.
- [KW07] Kurpjuweit, S.; Winter, R.: Viewpoint-based Meta Model Engineering. In: Proceedings, Enterprise Modelling and Information Systems Architectures – Concepts and Applications, Proceedings of the 2nd Int'l Workshop EMISA 2007, Bonn 2007, pp. 143-161.
- [MKW93] McGuire, J. G. et al.: SHADE: Technology for Knowledge-based Collaborative Engineering. In: Concurrent Engineering 1 (1993) 3, pp. 137-146.
- [Op07] The Open Group: The Open Group Architecture Framework TOGAF - 2007 Edition (Incorporating 8.1.1). Van Haren, Zaltbommel 2007.
- [ÖWH07] Österle, H. et al.: Business Engineering: Core-Business-Metamodell. In: Wisu – Das Wirtschaftsstudium 36 (2007) 2, pp. 191-194.
- [RWR06] Ross, J. W.; Weill, P.; Robertson, D. C.: Enterprise Architecture as Strategy. Creating a Foundation for Business Execution. Harvard Business School Press, Boston, MA 2006.
- [Sc04] Schekkerman, J.: How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework. 2nd edition, Trafford Publishing, Victoria, British Columbia 2004.
- [Sc05] Schekkerman, J.: The Economic Benefits of Enterprise Architecture: How to Quantify and Manage the Economic Value of Enterprise Architecture. Trafford Publishing, Victoria, British Columbia 2005.
- [Sh90] Shaw, M.: Prospects for an Engineering Discipline of Software. In: IEEE Software 7 (1990) 6, pp. 15-24.
- [Si99] Sinz, E. J.: Architektur von Informationssystemen. In: Rechenberg, P.; Pomberger, G. (eds.): Informatik-Handbuch. 2nd edition, Hanser, München, Wien 1999, pp. 1035-1046.
- [SR01] Sutton, S. M.; Rouvellou, I.: Issues in the Design and Implementation of a Concern-Space Modeling Schema. In: Proceedings, Advanced Separation of Concerns Workshop, Toronto, Canada 2001, pp. 119-124.
- [WBF07] Winter, R. et al.: Analysis and Application Scenarios of Enterprise Architecture – An Exploratory Study (Reprint). In: Journal of Enterprise Architecture 3 (2007) 3, pp. 33-43.
- [WF07] Winter, R.; Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In: Journal of Enterprise Architecture 3 (2007) 2, pp. 7-18.

- [Wi08] Winter, R.: Business Engineering – Betriebswirtschaftliche Konstruktionslehre und ihre Anwendung in der Informationslogistik. In: Dinter, B.; Winter, R. (eds.): Integrierte Informationslogistik. Springer, Berlin, Heidelberg 2008, pp. 17-38.
- [Zw48] Zwicky, F.: Morphological Astronomy. In: The Observatory 68 (1948), pp. 121-143.

Towards simulation-supported enterprise architecture management

Sabine Buckl¹, Florian Matthes¹, Wolfgang Renz²,
Christian M. Schweda¹, Jan Sudeikat^{2,3}

¹Lehrstuhl für Informatik 19 (sebis)
Technische Universität München
Boltzmannstr. 3
85748 Garching
{buckls,matthes,schweda}@in.tum.de

²Multimedia Systems Laboratory (MMLab)
Fakultät Technik und Informatik
Hamburg University of Applied Sciences
Berliner Tor 7
20099 Hamburg
{wr,sudeikat}@informatik.haw-hamburg.de

Abstract: Enterprise architecture management is based on a holistic view on the enterprise addressing business and IT aspects in an integrated manner. EA management is a process to manage the complexity of the overall architecture and to improve the alignment of business and IT. In order to achieve these goals, it is necessary but not sufficient to manage the static complexity that arises from dependencies between the elements of the EA, like goals, organizational units, business processes, business applications, and IT infrastructure elements. Performance, stability, and scalability can only be analyzed, modeled, and controlled, if static EA models are enriched by appropriate abstractions to capture the dynamic complexity of the EA understood as a socio-technical system of interacting (sub-)systems. This article identifies possible techniques to address dynamic complexity in EA. The potential benefits of system simulations are discussed and the derivation of appropriate simulation models is exemplified. A key observation is the fact that EA management is a iterative evolution process, where each iteration only changes a small fraction of the EA. It is therefore possible to automatically derive model parameters required for the simulation of the future architectures from an analysis of the dynamics of the current architecture.

1 Motivation

In large enterprises, the application landscape, as the entirety of the employed business applications [Wit07], is an important asset, which is a critical support factor for business and

³Jan Sudeikat is doctoral candidate at the Distributed Systems and Information Systems (VSIS) group, Department of Informatics, University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

a costly investment constantly demanding maintenance operations. Consequently, managing the application landscape has not only recently become a challenge, current enterprises have to address. Especially in the larger context of enterprise architecture (EA) management, the alignment of the enterprise's business and IT, forms the focal point. In the light of this increased interest from practitioners, different approaches to facilitate EA management have been developed in academic research [FAW07, Fra02, Lan05, Wit07], by practitioners [Der06, EHH⁺08, Kel07, Gro05], and by tool vendors [MBLS08]. Nevertheless, the approaches concentrate on structural aspects of the EA or the landscape respectively – such as interconnections between business applications.

In contrast, the EA and the application landscape in special form highly dynamic systems, with their behavior being far more complex than their topological structure is likely to indicate. Such *dynamic complexity* is a widely known and accepted fact in many management disciplines, of which first references date back to the 60^{ths} of the last century (cf. the *Forrester effect* [For61]). Concerning application landscapes or EAs, such considerations have not yet been undertaken, although some bordering fields from EA management, e. g. business process management, have already established means and methods for addressing and evaluating behavioral complexity, e. g. business process simulation [JVN06, LM04]. Thereby, the simulated concepts are considered on a high level of abstraction, omitting detailed internal descriptions in favor of a more *phenomenological* treatment of the subject. We regard the establishment of complementary high-level simulation methods for EAs a next important step towards a more mature management discipline and present an approach to this area alongside an application example in Section 4 of the article. Preceding Section 2 provides the foundation for the approach. Section 3 supplements some more elaborate considerations on the EA as a dynamic system. In conclusion, Section 5 gives indications on future research topics in this area.

2 Related work

The simulation approach presented in this paper targets two bordering fields – EA management and simulation techniques. An introduction to both areas is subsequently given, and their impact on the presented approach is outlined.

2.1 EA management

The topic EA management is an important issue in academia as well as in practice for several years now. One of the first papers targeting this field dates back to 1992, when Sowa and Zachman developed a framework for information systems architecture [SZ92]. Since that time, the number of publications concerning EA management has grown constantly [LW04], although no commonly accepted definition of the terms EA or EA management exists. Nevertheless, the various definitions available center around a common concept – a holistic view on the enterprise. Thereby, the areas of considerations may be

different, but mostly range from infrastructure to business or strategic aspects. Figure 1 shows the different layers and crossfunctions of an holistic EA management approach according to [MBLS08].

The color coding of Figure 1 indicates the variability concerning the involvement of information from specific layers or crossfunctions according to the different definitions of EA. While it is common, that information from all layers and cross functions is of importance in the context of EA management, different approaches vary widely in their definition to which level of abstraction this information should be detailed (see e.g. the approaches of [Fra02] and [BELM08]).

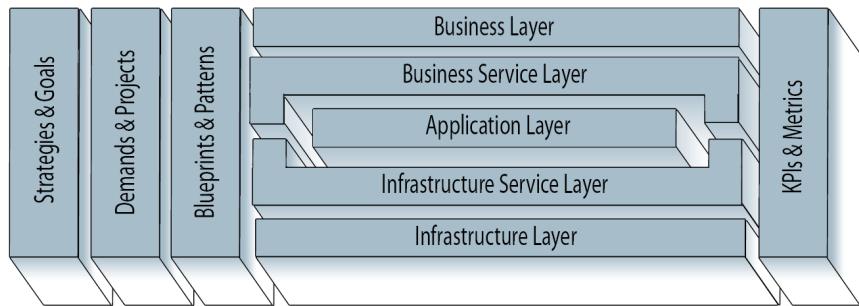


Figure 1: Layers and cross functions of an holistic EA management approach [MBLS08]

The elements of an EA as alluded to above are subject of the process of EA management. According to [MBLS08] EA management can be defined as *a continuous and iterative process controlling and improving the existing and planned IT support for an organization. The process not only considers the information technology (IT) of the enterprise, also business processes, business goals, strategies etc. are considered in order to build a holistic and integrated view on the enterprise. The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business.* One central task of EA management is the management of the application landscape [Lan05, Wit07]. As managing a complex asset as the application landscape is likely to involve a large number of people with different backgrounds (e.g. business process managers, project managers, or business application owners) communication is a major management issue. Therefore, graphical models called *software maps* [LMW05, BELM08] (as the one shown in Figure 2), have proven to be a valuable means of communication in this context. A real-world exemplary visualization¹ focusing on business application interdependencies and their distribution is shown in figure 2.

In addition to visualizations, metrics can be used to bridge the communication gap between the stakeholders from business and IT [AS08, LS07, LS08, Lan08]. Thereby, metrics build a more objective basis for decision making, helping the manager to not simply rely on his *gut feel*. [LS08] and [Lan08] demonstrate the applicability of metrics in the context of application landscape planning through comparing planned landscapes in respect to failure

¹Due to reasons of confidentiality the figure, which is taken from a EA practitioner, is made unreadable.

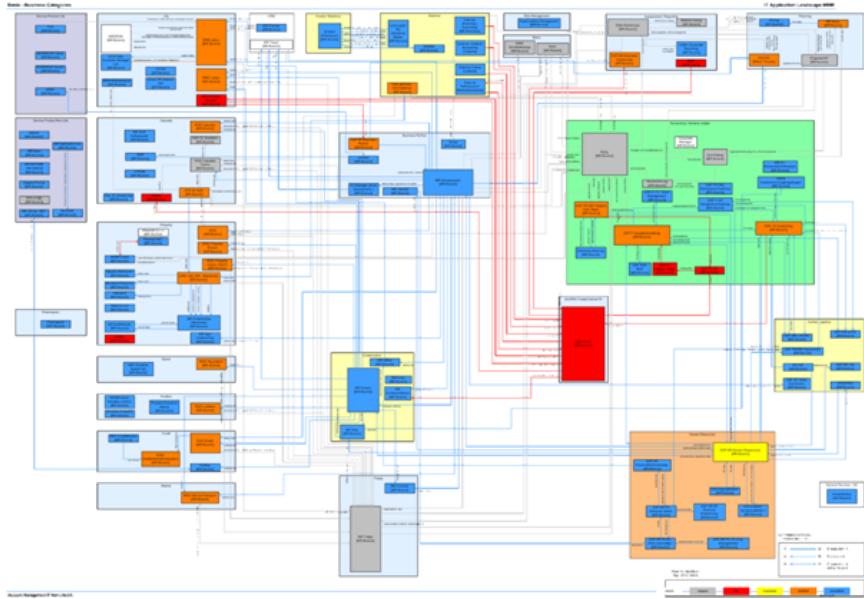


Figure 2: Software map focussing on business application interdependencies

propagation. Notwithstanding the importance of metrics concerning structural dependencies in the application landscape, also measurable properties exist, which arise from the interaction in these highly dynamic systems.

2.2 Simulation techniques

The only means to assess this dynamic complexity are system simulations (cf. [BF04] for an overview of simulation approaches), which therefore is a necessary prerequisite for measuring dynamic system properties. The observation of the timely development of system properties is supported by numerous simulation techniques ranging from purely mathematical approaches, e.g. *equation based modeling* (EBM), to formal modeling approaches, e.g. petri-nets [vdA03] and process algebra [Fok00] as well as object- and agent-oriented simulation methodologies [PSR98]. Mathematical and formal models are simulated by iterative evaluation, while the computational simulation models rely on the emulation of constituent system elements [PSR98]. In the following, we briefly outline the applicability of some of these models for the simulation of EA.

A prominent purely mathematical simulation technique is the *queueing theory* (see e.g. [Tij03]), a widely accepted and used means for deriving performance measures, e.g. processing time, in systems with a continuous random stream of incoming requests [vdA03, BLL⁺94]. Such a situation can be found at multiple occasions in the context of EAs, e.g. concerning business process execution in the back-office. The steps forming these pro-

cesses are thereby executed by business applications – their throughput performance can be computed using queue based simulation models. Such models have been successfully applied on business level (see e.g. [KB03]) or on infrastructure level concepts of the EA as well as on intra-application aspects, like load balancing mechanisms. Contrastingly, up to now, no attempt to use these queueing based simulation techniques on a more holistic model of the EA has been undertaken. This might be due to the fact, that on the level of the whole EA load information is often only known on a rather qualitative level, lacking precise quantitative information, which would be needed for queueing models. Additionally, an embracing, enterprise-wide queueing model is likely to fail for a complexity problem.

Formal simulation techniques, e.g. petri-nets and process algebra, are particularly attractive to model concurrent activities and the resulting macroscopic, dynamic system behavior. Petri-nets [vdA03] describe systems as graph structures, where system states are represented as markings of tokens spread over nodes. This modeling notion complicates the description of component data (*business objects*) and its change during state transitions, quickly calling for more sophisticated methods like colored petri-nets [Jen96]. Process algebra models [Fok00] extend the former approach by supporting compositionality, but as well adopt a purely behavioral stance toward component behaviors. Therefore, both modeling approaches allow to express the dynamic system behaviors, but their abstraction level demands additional effort to interpret the obtained results and relate them to the represented EA concepts.

Object- as well as agent-oriented simulation models enable fine-grained views on system components to resemble the resulting macroscopic system dynamics. In these approaches the simulation model consists of collections of passive or pro-active elements, respectively. Agent-based models are attractive to represent EA aspects as they support modeling systems as organizational structures of autonomous components [MY04] that decide locally on their interactive behavior. This abstraction level supports the representation of intelligent architectural elements that decide themselves to engage in computations, e.g. trigger business processes executions. Therefore, complex local rules, which control interactive computations, i.e. the realization of business processes, can be directly encapsulated to observe their effect on the resulting system behavior. In [PSR98], agent-based simulation models are distinguished from mathematical approaches. It has been argued that the ability to directly represent the element's internal computations in agent models, facilitates the transfer from simulation models to the represented systems. Particularly interesting in the context of EA simulations is the ability to enable experimentation by *what-if* games, e.g. via application landscape scenarios. The ability to directly model system components and their local computations can enable enterprise architects to alter local strategies and observe their impact on the resulting global system behavior. Moreover, the direct mapping between simulation models and the simulated subjects facilitates the implementation of the gained insights, i.e. modifications of the represented EA element. Due to these benefits, agent-based simulations are preferable to rigorous mathematical / formal simulation approaches. We expect that the local decisions of EA components can be mapped to reactive, rule-based reasoning mechanisms, which facilitates the utilization of sophisticated simulation environments [IGN05] that ensure the scalability of the simulation approach for large scale EA.

3 Dynamic system aspects of EAs

Application landscapes, as part of EAs, exhibit a multiplicity of relations within the set of interconnected business applications and components. These relations give not only rise to *structural complexity* but, according to the dynamic properties of the underlying dependencies, cause *dynamic complexity*. In system dynamics, analytic and constructive issues are considered, which are explained here because they will influence the interpretation and validity of statements about dynamic properties of a system.

System dynamics research coined the term *dynamic complexity* to describe the intricate difficulty to assess the long-term behavior of dynamic systems. This notion of complexity is distinguished from the complexity that rises from the combinational number of elements and their relationships. It describes the rise of complex, typically dysfunctional behavior that evolves from subtle cause and effect structures. When the long-term effects of short-time interventions are not obvious, this complexity can be observed in comparatively simple systems with low structural complexity.

In dynamic system analysis it is well known that the behavior of system components can depend on the very details of the imposed conditions. To be specific, constant work load or repeated transient peak load conditions with identical average work load can lead to dramatically different response behavior and throughput. Such effects often are not recognized, if the simulation model does not cover every detail, which at modeling time cannot be identified as crucial. System architects are trained to identify components of concern and the possible appearance of conditions causing unwanted behavior and provide means to stabilize the system properties. Nevertheless, in many cases such unwanted behavior is not discovered at system design time and is often difficult to detect in the running system because of its *dynamic complexity* induced by its complex internal dynamics as a response to simple use-cases.

Since the subtleties of causal relations among system components can lead to dysfunctional system behaviors (e. g. discussed in [Mog05]), the architects of EA face the challenge to infer the dynamic system properties from the structural properties of the conceived EA. We propose the use of system simulations to facilitate the examination of enterprise-wide dynamics and the anticipation of effects of landscape modifications. Architects will use these simulations to examine both the *operational* properties of EAs, e. g. performance measurements, and strategic, non-functional properties, as scalability or robustness, which are influenced by global architectural decisions.

4 Exemplifying our simulation approach

Simulation techniques are commonly accepted means for getting insights into the dynamics of a system in many engineering disciplines. There actually exists a multitude of reasons, why simulation is employed, of which an overview is given e. g. in [Wik08]. Especially the following ones strongly apply in the context of simulating EAs or application landscapes respectively, namely:

- Analyzing the real application landscape's behavior in detail would be very costly and resource-consuming.
- Analyses of the behavior of future application landscape's cannot be performed, as setting up them in an experimental environment is not possible due to the expenses connected.
- Changing execution parameters in the real application landscape for reasons of analyses is not appropriate, as the effects might cause severe business incidents.

Creating a model suitable for simulation of the EA or the application landscape can therefore be considered a sensible way to approach the aspect of dynamic complexity. Such an approach is necessarily complemented with measures to assess certain aspects of the landscape and present them in a condensed form, e. g. by using application landscape metrics [Lan08, LS08].

Subsequently, we present our approach to EA simulation. Thereby, a real world problem of a large bank is used to briefly exemplify the steps to be taken. The large bank operates an application landscape containing a backend, which is purely host-based, i. e. it relies on mainframe systems, on which the core business functions are executed, while a distributed presentation logic exists. With the ever changing business environment on the one hand and the decreasing number of available software engineers skilled in host programming languages on the other hand, a major change in banking application landscapes is at sight. A transition is likely to lead from centralized host architectures to distributed (potentially service oriented) architectures. Nevertheless, this change of paradigm also brings along important uncertainties, especially concerning the latency and failure propagation of the revised application landscape. To illustrate the distribution issue, Figure 3 shows a simplified example of a host-based realization of a business process execution.

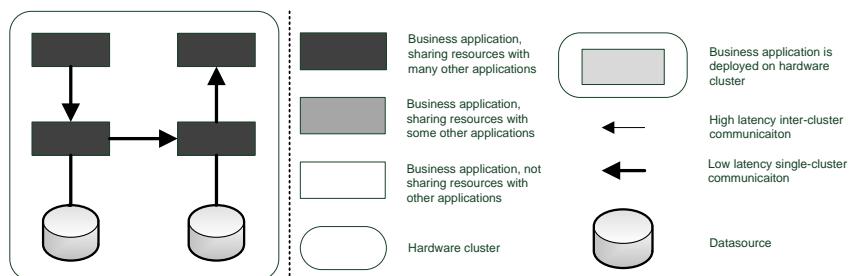


Figure 3: Cutout of the host-based landscape involved in the execution of a business process

Latency and failure propagation on the level of business process execution are clearly dependent on the distribution topology of the new landscape. This topology lays the ground for the complex dynamics, which can lead to reduced throughput and increased latency. Consequently, a company would look to achieve optimal performance properties for the landscape, by choosing the *right* topology. Nevertheless, finding out, what is *right*, must

be considered a complex task, for which the enterprise architects have not had the opportunity to develop a *gut* feel. This lack of experience is addressed by us via metrics indicating e.g. average latency based on simulated dynamics of the application landscape. Thereby, the metrics could be applied to different scenarios for future application landscapes, either created by an enterprise architect or machine-created ones randomly distributing the business applications on systems. After simulating the scenarios' dynamics and evaluating their performance indicators, the architects could choose the future landscape optimally fulfilling their performance requirements.

Alongside this example, we introduce the steps of an continuous and iterative process, supporting the simulation of various aspects of an EA. Thereby, the system of which different elements are simulated, is the EA or respective parts thereof, e.g. the application landscape. Due to reasons of brevity the example concentrates on the issue of latency as introduced above. The steps to be taken are:

1) Gather concerns and objectives: At first the pain points, which should be addressed by the simulation, have to be identified, e.g. by interviewing the respective stakeholders and gathering their concerns² regarding the system. These concerns have to be operationalized to objectives or key performance indicators (KPIs) in order to support the measurement of the target achievement.

Currently, the bank operates a host-based backend system for supporting its backoffice banking services. Driven by the lack of experienced host-programmers, the bank has chosen to migrate its backend to a distributed Java-environment. Thereby, questions like the following have to be answered prior the migration:

- How does the distribution affect the overall system performance, especially concerning the execution time of LUWs (latency)?
- What complex behavior might arise, e.g. regarding failure propagation in partially asynchronous environments? Consider a situation, in which a *fast* system having an incoming queue fails and is brought back to operation after a certain time. May it cause subsequent systems to break down due to too much load in little time?

2) Define objects of observation: In order to determine the parts of the system, which are relevant for the simulation, the objects of observation and their dependencies have to be identified. In addition, KPIs of the system have to be related to the objects of observation.

Business processes are supported by business services, which are realized by business applications utilizing infrastructure services. Currently, the applications supportive for a process are executed on a single host, thus neither distribution of data nor network communication is employed.

The performance analysis in the scenarios mainly targets the processing times for the business processes to be executed. Therefore, the *average processing*

²The terms *concern* and *stakeholder* are used here in accordance to the definitions provided in the IEEE Std 1471-2000 [IEE00].

time per business process request and the *standard deviation* of processing time per business process can be considered to be suitable KPIs.

3) Identify classes and develop respective *information model*: The objects identified in the preceding step have to be classified and their relationships to each other have to be conceptually modeled. This leads to an information model, which represents the influential constituents of the landscape and their relationships to each other according to the simulation scenario addressed.

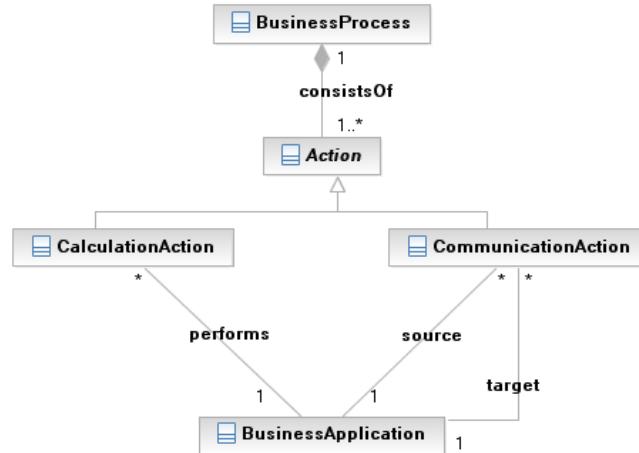


Figure 4: Simplified information model for the banking example

An information model for the banking example has to consider the business processes, the supportive service-providing business applications, and the infrastructure services, which are utilized by the applications. Additionally, the ordering of business service calls in the execution of a business process has to be taken into consideration as well as the information flows between the business applications. Figure 4 provides a simplified information model for the banking example, in order to indicate the level of granularity on which information is needed for simulation. The concepts introduced therein should be self-explanatory and are hence not detailed here.

For supporting simulation, this static aspects have to be complemented with further dynamic information, e. g. concerning the distribution of business process requests over time. Additionally, means to model the resource usage of an application in executing a service request have to be incorporated in the information model, e. g. via in-application cause-effect modeling facilities. Further, the distributed environment calls for ways to model different types of inter-application dependencies:

Communication dependencies two or more business applications communicate in the execution of one business process either synchronously

(mostly) or asynchronously (seldom).

Resource dependencies two or more business applications from different business processes compete for the same resource, e.g. CPU.

Data dependencies one or more business applications are triggered by events raised by data changes via another business application.

4) Choose appropriate simulation technique: Before starting the simulation process, an appropriate simulation technique has to be chosen. As introduced in Section 2, a multitude of different simulation methodologies exists, of which the one best suitable should be taken according to the characteristics of the simulation and the system under consideration.

Purely equation based models could be used to simulate the dynamics of the described system in detail. Nevertheless, analytical solutions may not be easy to find due to the generality of the assumptions and the high level of abstraction. Further, especially resource dependencies cannot be modeled easily in those mathematical models.

Formal simulation tools, as petri-nets or process algebras, may seem appealing in this context, although their main drawback concerning the interpretation of the simulation results cannot be neglected. Further, especially the petri-nets are likely to restrict further refinements of the model especially concerning considerations on business objects exchanged.

Therefore, we decided to choose an agent-based simulation model as agents can more naturally model failure propagation in asynchronous environments. Further agents can be re-used, e.g. if the scenario is extended to adaptive business application relocation strategies to compensate load-peaks or system-failures.

5) Map information model to executable simulation model: The information model, containing all relevant elements for the simulation, e.g. business applications, interfaces, business objects, etc., has to be mapped to an executable simulation model of the application landscape.

Agent-based simulation models can be derived at different levels of granularity that range between two extremes. On the one hand the most detailed representations represent all active parts, e.g. business process and the services that perform individual activities, as individual agents. On the other side it is also possible to represent subsets of active elements, e.g. hardware components, by agents that manage the interactions of logically associated elements.

Following the requirements for the information model given above, the individual *business applications* lend themselves to be described by agent models. Agent instances include the local rules that decide when to engage in the execution of a business service. While several operations, e.g. internal computations in an application, are to be mimiced by agent internal processing,

can communication actions be represented by means of inter-agent communication.

- 6) Identify simulation parameters:** The relevant simulation parameters, complementing the information model defined above, have to be identified.

In the context of the banking example, the parameters necessary for simulating the behavior especially center around the mapping of service requests on business application level to the degree of infrastructure service consumption caused thereby. Additionally, in-application processing times for internal computations make up an important parameter of the simulation model as well as the latency of inter-application communication may be. Finally, the frequency for each of the business processes to be supported can be considered a valuable parameter.

- 7) Determine / estimate simulation parameter values:** For each of the simulation parameters identified before, a value has to be determined. These values can be derived e. g. from estimations from the expertise of business application owners or to be more objective from empirical data on current behavior of the actual systems.

The *frequency of executions per business process* could be determined using a business perspective, e. g. by counting the number of money transfers per day. *Resource consumption* and are not that easy to determine, therefore a simple estimation can be used, assuming that an internal computation utilizes 100% of a CPU core for a certain roughly estimated timespan. *Communication latency* can be determined using static methods on the data contained in log-files of a small-scale experimentation environment. Concerning the values for the simulation parameters, assumptions made have to be explicated, in order to facilitate future validation or adaption of the simulation model.

- 8) Create evolution scenarios of the landscape:** Based on the status quo of the application landscape (current landscape) different evolution scenarios (planned landscapes) have to be created, which represent possible evolutions scenarios. These scenarios can then be compared based on the objectives in order to support the decision process concerning the future evolution of the application landscape via the simulation results.

Potential ways to distribute the service providing business applications to different hardware clusters are made explicit in the evolution scenarios. Thereby, the scenarios are developed by enterprise architects of the bank, taking into consideration similarities between the applications to distribute. Further scenarios are created by combinatorial search algorithms, finding pareto-optimal solutions in the distinct business processes. Optimal scenarios derived by the latter way are subsequently reviewed by architects, as they could be optimized according to performance aspects, but may be senseless in respect to business usage. Potential scenarios conforming to the simple example from Figure 3 are illustrated in the software maps shown in Figure 5.

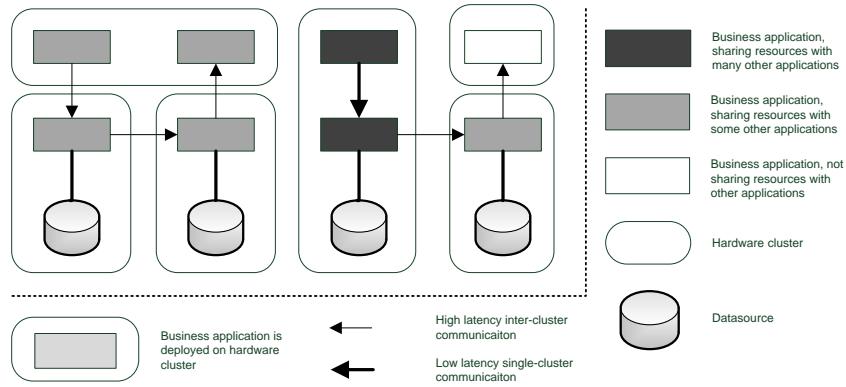


Figure 5: Distribution scenarios for the landscape involved regarding the execution of a business process

The evolution scenario, which achieved the higher outcome is implemented. Thereby the planned landscape is transformed to the current landscape.

9) Validate the simulation results with actual data: Based on the behavior of the new systems representing the previous planned landscape, the simulation results are validated by comparison with the actually measured empirical data of the new system.

A new landscape has been implemented by projects and is finally operated by the IT staff. Performance measures regarding latency can be assessed directly by monitoring request processing time or indirectly by user satisfaction analyses. Especially direct results may be used to adapt simulation parameters, to achieve a better fitting. Indirect observations may at least be helpful to proof or neglect the simulation to be strongly deviating from reality.

In order to improve the simulation process introduced above, the findings of the final step *Validating the simulation results with actual data* should be used as input for the next iteration of the process. Thereby, each step, e. g. *the development of the information model* should be reconsidered in order to improve the outcomes of the simulation process, e. g. by introducing new concepts relevant for the simulation.

5 Critical reflection and outlook

In this paper we have motivated the necessity to enrich today's static EA models that focus on the static complexity arising from dependencies between the elements of the enterprise architecture by appropriate abstractions to capture the dynamic complexity of the EA understood as a socio-technical system of interacting systems. Based on related work on application landscape management, metrics, and simulation techniques established in

other management areas, we outlined an approach to EA simulations. We argued that simulations may provide decision support to develop better performing, more robust and adaptive EAs.

Nevertheless, in this paper we have neither provided a detailed information model nor a full-scale simulation model. To apply the approach presented, both models would be necessary but also dependent on the actual environment, in which simulation supported EA management was applied. Thereby, the steps 6)(identify simulation parameters) and 7)(determine / estimate simulation parameter values) of our approach form a major challenge in a practical environment. To address these issues a validation in the real world would be required. The real world example presented alongside the approach provides a real world example Although a real world example originating from practice is used to detail on the different steps of our simulation approach, case studies have to be accomplished in order to validate the presented approach in a practical setting. We argued that architectural decisions can not only be guided by metrics [Lan08, LS08] but may also be supported by predictive simulations of architecture designs. These designs provide so-called information models of the constituent architecture elements which are the subjects of simulation models. We discussed established simulation techniques and concluded that agent-orientation facilitates the derivation of simulation models. Furthermore agent technology provides a variety of simulation environments and tool support ready for use in the simulation methodology presented here.

Future prospects comprise the integration of application landscape modeling and simulation in an integrated tool chain. System dynamics research established *what-if* simulations as predictive tools to support management decisions (e.g. [Ste00]). Here we propose that EA architects would benefit from a similar approach. Crucial is the seamless integration of modeling and simulation tool support, i. e. being able to automate the derivation of simulation models from landscape descriptions. Since EAs depend on recurring types of architectural elements, companies may maintain a library of reusable simulations elements. These describe common architectural elements parameterized by specific information models. The here advocated agent-based simulation approach supports the required composition of simulation models. Since the complexity of these simulation models grows with the scale of the abstracted EA's, sophisticated simulation environments will be utilized [IGN05]. Particularly challenging in this context is ensuring the proper behavior of the system model, i. e. validating that the derived simulation elements behave as the architectural elements they represent. As this demands manual modeling and simulation effort, it is of interest to enable the reuse of simulation models, e. g. as executable architectural pattern.

References

- [AS08] J.S. Addiks and U. Steffens. Supporting Landscape Dependent Evaluation of Enterprise Applications. In *Multikonferenz Wirtschaftsinformatik (MKWI) 2008*, pages 1815 – 1825, Munich, Germany, 2008.
- [BELM08] S. Buckl, A. Ernst, J. Lankes, and F. Matthes. *Enterprise Architecture Management*

Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19, Technische Universität München, Munich, Germany, 2008.

- [BF04] Andrei Borshev and Alexei Filippov. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasong, Techniques, Tools. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, Oxford, UK, 2004.
- [BLL⁺94] R. Bhaskar, H.S. Lee, A. Levas, R. Petrkian, F. Tsai, and B. Tulskie. Analyzing and re-engineering business processes using simulation. In *Proceedings of the 1994 Winter Simulation Conference*, pages 1206–1213, Lake Buena Vista, FL, USA, 1994.
- [Der06] G. Dern. *Management von IT-Architekturen (Edition CIO)*. Vieweg, Wiesbaden, 2006. ISBN 528158166, (in German).
- [EHH⁺08] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.-P. Richter, M. Voss, and J. Willkomm. *Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten*. dpunkt.verlag, Heidelberg, 2008. ISBN 3898645061, (in German).
- [FAW07] R. Fischer, S. Aier, and R. Winter. A Federated Approach to Enterprise Architecture Model Maintenance. In *Enterprise Modelling and Information Systems Architectures Proceedings of the 2nd International Workshop EMISA 2007*, pages 9–22, St. Goar, Rhine, Germany, 2007.
- [Fok00] Wan Fokkink. *Introduction to Process Algebra*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2000. ISBN: 3-540-66579-X.
- [For61] J.W. Forrest. *Industrial Dynamics*. MIT Press, 1961.
- [Fra02] U. Frank. Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences 35*, Honolulu, 2002.
- [Gro05] *The Open Group*. TOGAF "Enterprise Edition" Version 8.1., 2005.
- [IEE00] IEEE. *IEEE Std 1471-2000: Recommended Practice for Architectural Description of Software-Intensive Systems*, 2000.
- [IGN05] Toru Ishida, Les Gasser, and Hideyuki Nakashima, editors. *Massively Multi-Agent Systems I*, volume 3446 of *LNCS*. Springer, 2005.
- [Jen96] K. Jensen. *Coloured Petri nets: basic concepts, analysis methods and practical use: volume 1*. Springer-Verlag, London, UK, 2nd edition, 1996. ISBN 3-540-60943-1.
- [JVN06] M. Jansen-Vullers and M. Netjes. Business process simulation – a tool survey. In *Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Denmark, 2006.
- [KB03] S. Kounev and A. Buchmann. Performance Modelling of Distributed E-Business Applications using Queuing Petri Nets. In *Proc. of the 2003 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 143–155, Austin, Texas, 2003.
- [Kel07] W. Keller. *IT-Unternehmensarchitektur*. dpunkt.verlag, 2007. ISBN 3898644197 (in German).
- [Lan05] M. Lankhorst. *Enterprise Architecture at Work*. Springer, Berlin, Heidelberg, 2005. ISBN 3540243712.

- [Lan08] J. Lankes. *Metrics for Application Landscapes – Status Quo, Development, and a Case Study*. PhD thesis, Technische Universität München, Fakultät für Informatik, Munich, Germany, 2008. (submitted).
- [LM04] M. Laguna and J. Marklund. *Business Process Modeling, Simulation, and Design*. Prentice Hall, 2004.
- [LMW05] J. Lankes, M. Matthes, and A. Wittenburg. Softwarekartographie: Systematische Darstellung von Anwendungslandschaften. In *Wirtschaftsinformatik 2005*, Bamberg, Germany, 2005. (in German).
- [LS07] J. Lankes and C.M. Schweda. *Constructing Application Landscape Metrics: Why & How*. Technical Report TB0701, Technische Universität München, Institut für Informatik, Lehrstuhl für Informatik 19, Munich, Germany, 2007.
- [LS08] J. Lankes and C.M. Schweda. Using Metrics to Evaluate Failure Propagation in Application Landscapes. In *Multikonferenz Wirtschaftsinformatik (MKWI) 2008*, pages 1827 – 1838, Munich, Germany, 2008.
- [LW04] K. Langenberg and A. Wegmann. *Enterprise Architecture: What Aspects is Current Research Targeting*. Technical Report C/2004/77, Ecole Polytechnique Fédérale de Lausanne (EPFL), France, 2004.
- [MBLS08] F. Matthes, S. Buckl, J. Leitel, and C.M. Schweda. *Enterprise Architecture Management Tool Survey 2008*. TU München, Chair for Informatics 19 (sebis), Munich, Germany, 2008. ISBN 3-00-024520-0.
- [Mog05] Jeffrey C. Mogul. *Emergent (Mis)behavior vs. Complex Software Systems*. Technical Report HPL-2006-2, HP Laboratories Palo Alto, 2005.
- [MY04] XinJun Mao and Eric Yu. Organizational and Social Concepts in Agent Oriented Software Engineering. In *Agent-Oriented Software Engineering V, 5th International Workshop, AOSE 2004, Revised Selected Papers*, volume 3382 of *LNCS*, pages 1–15, New York, USA, 2004. Springer.
- [PSR98] H. V. D. Parunak, Robert Savit, and Rick L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 10–25, London, UK, 1998. Springer-Verlag.
- [Ste00] John D. Sterman. *Business Dynamics – Systems Thinking an Modeling for a Complex World*. McGraw-Hill, 2000.
- [SZ92] J. F. Sowa and J. A. Zachman. Extending and formalizing the framework for information system architecture. *IBM System Journal*, 31 (3), 1992.
- [Tij03] H.C. Tijms. *A First Course in Stochastic Models*. Wiley & Sons, 2003.
- [vdA03] W.M.P. van der Aalst. *Application and Theory of Petri Nets*, pages 1–22. Springer Verlag, Eindhoven, Netherlands, 2003.
- [Wik08] Wikipedia. Wikipedia – Wikipedia, *The Free Encyclopedia*, 2008. [Online; accessed 22-July-2004], (in German).
- [Wit07] A. Wittenburg. *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften*. Phd thesis, Fakultät für Informatik, Technische Universität München, Munich, Germany, 2007. (in German).

Prozessmusterbezogene Kommunikationsmaßnahmen im Requirements Engineering

- Eine Fallstudie

Markus Schäfermeyer, Marianne Corvera

Goethe-Universität Frankfurt
Professur für Information Systems Engineering
Mertonstraße 17
60325 Frankfurt am Main
schaefermeyer|corvera@wiwi.uni-frankfurt.de

Abstract: Erfolgreiches Requirements Engineering (RE) kann nicht nur als Sammelbegriff für praxiserprobte Methoden und Vorgehensweisen der Anforderungsspezifikation verstanden werden, sondern auch als soziale Interaktionsform, die durch effektiven Sprach- und Kommunikationseinsatz charakterisiert ist. Diese Arbeit stellt Ergebnisse einer explorativen Fallstudie vor, die in der Softwareentwicklungsabteilung eines deutschen Telekommunikationsunternehmens durchgeführt wurde. Ziel des Beitrages ist es zunächst, das RE-Vorgehen dieses Unternehmens als Prozessmuster zu skizzieren. Danach sollen innerhalb des betrachteten RE-Prozesses verwendete Kommunikationsmaßnahmen identifiziert werden, um sie anhand geeigneter Faktoren im Hinblick auf ihre Fähigkeiten, Kommunikation und Interaktion zu ermöglichen, analysieren zu können.

1 Einleitung

Unter dem Begriff Requirements Engineering werden allgemein Methoden und Vorgehensweisen zusammengefasst, die den Prozess der Konsolidierung und Legitimierung von Anforderungen sowie deren Überführung in eine adäquate Systemspezifikation beschreiben. Im Fokus dieser Arbeit steht die Gestaltung der sprachlichen und kommunikativen Dimension innerhalb dieses Prozesses. Der effiziente Einsatz von adäquaten Kommunikationsmaßnahmen zur Unterstützung des Wissenstransfers zwischen Stakeholder und Requirements Engineer ist kritisch für erfolgreiches RE und die spätere Implementierung einer mit den Anforderungen übereinstimmenden Software. Nach Studien der Standish Group werden nur 28 % der Softwareentwicklungsprojekte planmäßig abgeschlossen. Das bedeutet, dass es lediglich bei gut einem Viertel aller Projekte im Bereich der Softwareentwicklung gelingt, diese innerhalb des geschätzten Zeitplans, mit dem zur Verfügung stehenden Budget und den bereitgestellten Ressourcen durchzuführen. 23 % der Projekte schlagen fehl und die restlichen 49 % überschreiten mindestens

eines der für ein erfolgreiches Projekt zuvor genannten maßgeblichen Kriterien [St01]. Häufig sind derartige Probleme auf mangelhaftes RE und damit eng verbundene Sprach- und Kommunikationsprobleme zurückzuführen.

Dieser Beitrag stellt die Ergebnisse einer explorativen Fallstudie vor, die in der Entwicklungsabteilung eines deutschen Telekommunikationsunternehmens durchgeführt wurde, welches in diesem Beitrag anonym bleibt und im Folgenden TK genannt wird. Ziel der Fallstudie ist es zunächst, das RE-Vorgehen dieses Unternehmens als Prozessmuster zu skizzieren. Danach sollen innerhalb des betrachteten RE-Prozesses verwendete Kommunikationsmaßnahmen identifiziert werden, um sie anhand geeigneter Faktoren im Hinblick auf ihre Fähigkeiten, Kommunikation und Interaktion zu ermöglichen, analysieren zu können. Erfolgreiche und direkte Kommunikation ist von entscheidender Bedeutung wenn es darum geht, den RE-Prozess so effizient wie möglich durchzuführen. Im Zuge dessen konzentrieren sich die Autoren aus kommunikativer und sprachlicher Perspektive auf folgende Forschungsfragen: Welches Prozessmuster lässt sich im RE-Vorgehen des Unternehmens TK erkennen? Welche Auswirkung haben die in den einzelnen Phasen des Prozessmusters eingesetzten Kommunikationsmaßnahmen auf Kommunikation und Konsensbildung? Wie hat sich der Einsatz einzelner Maßnahmen im Laufe der Zeit verändert? Welche Rückschlüsse lassen sich daraus für den RE-Prozess aus kommunikativer und sprachlicher Perspektive ableiten?

Hierzu werden in Abschnitt 2 zunächst grundlegende Konzepte aus mit dieser Arbeit verwandter Literatur dargestellt, um die theoretische Basis für die nachfolgende Analyse der durchgeführten Fallstudie zu bilden. Anschließend wird in Abschnitt 3 sowohl das methodische Vorgehen erläutert als auch die Fallstudie dargestellt. Des Weiteren werden in Abschnitt 4 Ergebnisse analysiert, qualitativ interpretiert und kritisch diskutiert. Schließlich fasst Abschnitt 5 die entscheidenden Erkenntnisse unter Berücksichtigung der genannten Forschungsfragen zusammen.

2 Requirements Engineering

2.1 Der effiziente Prozess

RE ist als Teilgebiet des Software Engineerings definiert, das sich mit aus der realen Welt stammenden Zielen, Funktionen und Einschränkungen von Softwaresystemen befasst. Ebenso betrachtet es die Beziehungen dieser Faktoren untereinander, um das Softwareverhalten spezifizieren und dessen Entwicklung über die Zeit verfolgen zu können [Za97]. RE wird als Prozess beschrieben, der den Zweck einer Software ergründet, indem er Stakeholder und deren Bedürfnisse identifiziert, diese dokumentiert und der Analyse zugänglich macht, um sie kommunizieren, diskutieren und abschließend implementieren zu können [NE00]. Das Ziel von RE ist es, qualitativ hochwertige Anforderungen zu generieren. Dabei ist eine Anforderung als Eigenschaft, Bedingung oder Fähigkeit zu verstehen, die ein Benutzer benötigt, um ein Ziel zu erreichen oder ein Problem zu lösen [Eb05]. Will eine Software Benutzer zufrieden stellen, so muss sie deren Anforderungen erfüllen.

Im Verlauf eines allgemein formulierten RE-Prozesses werden zunächst Anforderungen der Stakeholder analysiert (Anforderungsanalyse), danach in Dokumenten niedergeschrieben (Anforderungsdefinition) und abschließend spezifiziert (Anforderungsspezifikation) [NE00]. Dabei stellt die Anforderungsspezifikation eine präzise und detaillierte Beschreibung der aus den Benutzeranforderungen hergeleiteten Systemanforderungen dar. RE kann demnach als Prozess verstanden werden, bei dem problembeschreibende Benutzeranforderungen zunächst extrahiert, analysiert sowie aggregiert und danach in ein adäquates Lösungskonzept transformiert werden.

Um den im Rahmen dieser Arbeit verwendeten Softwarebegriff zu konkretisieren, wird auf die von Lehman entwickelte Software-Kategorie „E-Typ“ (Evolution) verwiesen [Le80]. Dieser Software-Typ beschreibt alle Programme bzw. Anwendungen, die ein Problem aus der realen Welt behandeln, steuern und innerhalb einer Computeranwendung implementieren [LR01]. Derartige Programme mechanisieren menschliche Aktivitäten und unterliegen gerade daher häufigen Veränderungen. Einmal eingeführt wird das E-Typ-System selbst Teil des Anwendungsbereiches, wodurch E-Typ-Software die eigenen Anforderungen beeinflusst. E-Typ-Software ist demnach als ein Programm zu verstehen, das sich ständig weiterentwickelt, solange es aktiv verwendet wird [Le80]. Neu wahrgenommenen Veränderungen müssen entsprechend analysiert werden. Sie führen zu neuen Anforderungen, die definiert, spezifiziert und schließlich im Programm umgesetzt werden müssen. Als Maß für den Erfolg dieser Anstrengungen dient speziell bei E-Typ-Software die Zufriedenheit der Benutzer und Entwickler. Zur Gewährleistung dieser Zufriedenheit muss Software ständig verändert und verbessert werden [LR01]. Das effiziente Management dieses kontinuierlichen Veränderungsprozesses, der gerade im Hinblick auf die Weiterentwicklung von Software-Systemen bestimmten Gesetzen [Le96] zu folgen scheint, ist hierbei von entscheidender Bedeutung. Als fundamentale Aussage dieser Gesetze kann festgehalten werden, dass die notwendige Umsetzung neuer Anforderungen die Komplexität des bestehenden Systems fortlaufend steigert und sich somit negativ auf die bisherige Struktur eines Softwaresystems auswirken kann [Be02].

In Ahnlehnung an Brooks ist Komplexität in Essenz (Essence) und Akzidenz (Accidents) zu untergliedern [Br87; Br95]. Dabei wird Essenz als probleminhärerente Komplexität definiert, die von Natur aus Teil der Software ist. Essentielle Komplexität ist demnach nicht reduzierbar, wenn das ursprüngliche Problem eines realen Sachverhalts mittels Software modelliert bzw. vollständig erfasst werden soll. Während der Softwareproduktion entstehende und demnach nicht probleminhärerente Komplexität ist nach Brooks hingegen als Akzidenz zu verstehen [Br87]. Folglich ist die auf Akzidenz beruhende Komplexität auf die Methodik zurückzuführen und daher per Definition vermeidbar. Berry verfeinert die Kategorisierung und definiert die Essenz als Anforderungen, die das Verhalten einer Software charakterisieren [Be02]. Probleme mit der Technik, mit deren Hilfe Anforderungen definiert und in eine Software überführt werden, sind jedoch der Akzidenz zuzuordnen. Ziel effizienten REs muss es demnach sein, die Essenz einer Software mittels geeigneter Vorgehensweisen unter minimalem Aufwand vollständig zu erfassen und dabei gleichzeitig die Akzidenz zu minimieren.

Bedingt durch den Evolutionscharakter des E-Typs muss allerdings berücksichtigt werden, dass die vollständige Erfassung der Essenz nur zu einem bestimmten Zeitpunkt erreicht werden kann. Periodisch betrachtet können Anforderungen bereits zum Zeitpunkt der Implementierung nicht mehr aktuell und die Essenz somit nicht mehr vollständig sein. Resultat dieser Überlegungen ist die Notwendigkeit eines kontinuierlichen und systematischen RE, um eine Software an die sich ständig verändernden Anforderungen der Umgebung und Benutzer anzupassen. Hierbei spielt der adäquate Einsatz von Kommunikationsmaßnahmen eine entscheidende Rolle.

2.2 Die Sprachperspektive des RE-Prozesses

Im Umgang mit Anforderungen gilt es eine weitere grundlegende Erkenntnis zu beachten: Eine falsche Erhebung der Anforderungen kann zu Eskalationen bezüglich geplantem Ressourceneinsatz oder zum Scheitern der Implementierung von Software führen [Ja94]. Aus kommunikativer und sprachlicher Perspektive wären Gründe hierfür eine lückenhafte Anforderungsspezifikation [WO92], eine Missdeutung der Anforderungen [Ly98] oder eine mangelnde Einbindung der Nutzer [BEJ06; KMR00; Ly98; Su95]. Diese vorwiegend als Akzidenz zu bewertenden Probleme ergeben sich wiederum aus Fehlern in der Konzeption von Art und Umfang der Kommunikationsmaßnahmen sowie aus Problemen in der Konsensfindung. Um dies zu vermeiden, ist die Gestaltung effektiver Kommunikationsmaßnahmen in der Softwareentwicklung von großer Bedeutung, denn sie entscheidet sowohl über die Wirtschaftlichkeit der Entwicklung als auch über den Erfolg des Anwendungssystems an sich [We81].

Grundlage für die Gestaltung von Software ist eine modellhafte Spezifikation der vom System zu leistenden Aufgaben anhand von Modellsprachen unter Verwendung von standardisierten Objekt- bzw. Prozessbezeichnungen. Dies bedeutet, dass eine erfolgreiche Konsolidierung von sinnvollen Spezifikationen mit der Kohärenz der modellhaften Spezifikation auf einer Sprach- und Wissensebene einhergeht [Bo79; Ly85]. Um eine Konsolidierung auf sprachlicher Ebene zu erreichen, müssen die einzelnen Stakeholder einen sprachlichen Konsens über die einzelnen Konstrukte bilden, d.h., sie müssen in Bezug auf Konzepte als auch auf repräsentierende Symbole eine Sprachgemeinschaft bilden [Ho07; KL96]. Dieser Ansatz greift auf semiotische Konzepte von De Saussure [Sa74] zurück. Demnach ist ein Zeichen das Ergebnis aus der Verbindung eines Konzepts (Signifikat bzw. mentale Fakten) mit einem das Konzept repräsentierenden Symbol (Signifikant). Wenn bei allen Stakeholdern die gleichen Verknüpfungen zwischen einzelnen Konzepten und jeweiligen Symbolen bezüglich einem Thema bestehen, gehören sie einer Sprachgemeinschaft zu eben diesem Thema an [KL96]. In diesem Sinne wird sprachliche Konsensbildung als der durch die Nutzung von Kommunikationsmedien und Kommunikationsmaßnahmen bedingte Einigungsprozess über Konzepte und Symbole verstanden.

Dabei gilt es allerdings zu beachten, dass im RE-Prozess Menschen mit heterogenen Wissensgebieten und (fach-)sprachlichen Hintergründen involviert sind [KL03]. Hier sind beispielsweise Key-Stakeholder (KS) als Vertreter der unterschiedlichen Geschäfts-

bereiche des Unternehmens und des Managements sowie Prozessmanager (PM) als Requirement Engineers als auch Entwickler bzw. Programmierer (EN) zu nennen.

2.3 Die Kommunikationsperspektive des RE-Prozesses

Sprachliche Konsolidierung erfolgt in einem sozialen Handlungs- und Kommunikationsprozess [HKN91]. Daher greifen wir auf zirkulare Kommunikationsmodelle zurück [SW95]. Hier wird Kommunikation als die Übermittlung und Interpretation von Information zwischen Sender und Empfänger in einem gegebenen Kontext verstanden. Diese erfolgt mithilfe von Kommunikationsmedien in einem spezifischen Kommunikationssetting. Zudem wird Reziprozität im Kommunikationsakt impliziert.

Der Einfluss von Kommunikationsmedien wird durch den Einsatz von Medientheorien untersucht, welche die Kommunikationscharakteristiken unterschiedlicher Medien in einem spezifischen Kommunikationskontext bewerten. Um die Effektivität von Kommunikation aus der angemessenen Verknüpfung der Fähigkeiten von Kommunikationsmedien entlang den Anforderungen des Kommunikationsprozesses abzuleiten, wird Bezug zur Theorie der Mediensynchronizität nach Dennis und Valachich [DV99] genommen. Demnach haben Medien unterschiedliche Fähigkeiten, Kommunikation zu ermöglichen und können anhand der folgenden fünf Medienfaktoren analysiert werden [DV99]:

- *Geschwindigkeit des Feedbacks* (immediacy of feedback) bewertet die Unmittelbarkeit der Rückmeldung,
- *Symbolvarietät* (symbol variety) bezieht sich auf die Anzahl der Möglichkeiten, in denen Informationen übermittelt werden können,
- *Parallelität* (parallelism) wird durch die Anzahl gleichzeitig ablaufender Kommunikationsvorgänge definiert,
- *Überarbeitbarkeit* (rehearsability) beschreibt die Möglichkeiten der Überarbeitung einer Nachricht vor dem Senden und
- *Wiederverwendbarkeit* (reprocessability) bezieht sich auf die Möglichkeit, eine Nachricht ohne Medienbrüche zu späteren Zeitpunkten wiederverwenden zu können.

Kommunikationsmedien sind z.B. Email, Telefonat oder Brief. Dabei spielt Sprache eine besondere Rolle: Weil sie als Instrument der erwähnten Kommunikationsmedien verwendet wird; weil sie grundsätzlich in zwei verschiedenen Formen auftreten kann – mündlich oder schriftlich – und schließlich weil sich Sprache als Medium in durch den Gebrauch bedingter, ständiger Veränderung befindet [Ro89].

In dieser Betrachtung wird der Kommunikationskontext insofern berücksichtigt, als dass nicht nur die Kommunikationsmedien sondern die Konzeption von Kommunikationsmaßnahmen betrachtet werden. Grund dafür ist insbesondere, dass für die Erzielung

sprachlichen Konsens eine zusätzliche Perspektive notwendig ist, die es ermöglicht, Kommunikation als Form der Interaktion zu analysieren. Eine Kommunikationsmaßnahme wird in dieser Arbeit als eine durch den Einsatz von Kommunikationsmedien hergestellte Interaktionssituation definiert, die in einem räumlichen und zeitlichen Kontext unter Einbeziehung von unterschiedlichen KS stattfindet. Insofern werden die Kontextfaktoren Raum, Zeit und Anzahl der KS ebenfalls berücksichtigt. Um die Konsolidierung sprachlichen Konsenses innerhalb der Fallstudie besser verstehen zu können, werden die im RE-Prozess eingesetzten Kommunikationsmaßnahmen bezüglich ihrer Charakteristika anhand der unterschiedlichen Medienfaktoren untersucht und entsprechend der anfangs genannten Fragestellung analysiert.

3 Explorative Fallstudie

Im Folgenden wird der RE-Prozess des Unternehmens TK mit Hilfe einer explorativen Fallstudie untersucht. Im Sinne von Yin [Yi03] fokussieren die Autoren das Phänomen der Gestaltung von Kommunikation im Kontext eines unternehmensspezifischen RE-Prozesses. Ziel dieser Fallstudie ist es, ein kommunikationsbezogenes Prozessmuster darzustellen, das eine in einem Unternehmen mit der Zeit entwickelte, praktisch erprobte, erfolgreiche Herangehensweise zur Anforderungspezifikation aufzeigt. Hierfür werden zunächst die einzelnen Phasen des Prozesses beschrieben und anschließend die jeweils verwendeten Kommunikationsmaßnahmen identifiziert sowie in ihrer zeitlichen Entwicklung analysiert. Parameter der Fallstudie sind somit der Medieneinsatz im Kommunikationsprozess und die Bildung sprachlichen Konsenses. Im Hinblick auf die von Lehman definierte Softwarekategorie „E-Typ“ wird aus evolutionärer Perspektive untersucht, welche Kommunikationsmaßnahmen sich aus welchen Gründen als dauerhafter Bestandteil im RE-Prozess des Unternehmens TK etabliert haben.

3.1 Methodisches Vorgehen

Die im Folgenden dargestellte explorative Fallstudie legt den Grundstein der Untersuchung und beschreibt das subjektive Vorwissen bzw. das interpretative Verständnis der Autoren [Ei89]. Die hierfür verwendete Methode der Fallstudie eignet sich besonders um Forschungsbereiche zu untersuchen, in denen sich die Theorie fortlaufend weiterentwickelt [Yi03]. Um die notwendige Stringenz zu gewährleisten, erfolgt die Betrachtung des Untersuchungsfeld aus unterschiedlichen Blickwinkeln, die alle als Quellen der Evidenz dienen [Yi03].

Aufgrund der einjährigen Tätigkeit eines der Autoren innerhalb des Unternehmens TK liegt ein vollständiger Einblick in alle operativen Prozesse, Informationssysteme, Dokumente und Berichte vor. Neben teilnehmenden Beobachtungen konnten Erkenntnisse aus Interviews gewonnen werden. Befragt wurden zwei Mitarbeiter (PM) des Unternehmens, die für den RE-Prozess der betrachteten Software hauptverantwortlich sind. Die Interviews wurden dokumentiert. Ergänzend konnten betriebsinterne Dokumente miteinbezogen werden, die im Zuge der Planung und Umsetzung der einzelnen Entwicklungsphasen der Software erstellt wurden. Dementsprechend decken die Interviews und Dokumente den kompletten Entwicklungszeitraum der Software ab. Diese Daten werden durch persönliche Beobachtungen des Autors ergänzt, der von August 2006 bis Juli 2007 aktiv am Softwareentwicklungsprozess teilnahm und Beobachtungen notierte.

3.2 Business Overview

Untersuchungsgegenstand der Fallstudie ist die Softwareentwicklungsabteilung des deutschen Telekommunikationsunternehmens TK. TK ist an einem Markt tätig, dessen Umsatzvolumen im Jahr 2005 bei 29 Mrd. € lag. Im Geschäftsjahr 2006/07 erzielte das Unternehmen Umsatzerlöse in Höhe von ca. 2 Mrd. € sowie einen Periodenüberschuss von 100 Mio. €. TK agiert deutschlandweit und beschäftigt derzeit etwa 3.800 Mitarbeiter. Das Produktpotfolio setzt sich aus verschiedenen Sprach- und Datenprodukten zusammen, welche in Abstimmung auf individuelle Kundenbedürfnisse sowohl für Privat- als auch für Geschäftskunden angeboten werden.

Allgemein betrachtet liegen die Aufgaben der Entwicklungsabteilung, im speziellen die der zwei interviewten PM, einerseits in der Mitgestaltung und der Optimierung der TK-Geschäftsprozesse, andererseits in der Sicherstellung einer angemessenen IT-Unterstützung dieser Prozesse. Zu ihrem Aufgabenbereich gehört es, alle operativen Geschäftsprozesse für den effizienten Betrieb des Telekommunikationsnetzes fortwährend zu optimieren, zu gestalten und mit adäquaten Softwaretools zu unterstützen. Im Zuge dessen sind die interviewten PM für die Annahme, Überprüfung und Klärung von Neu- bzw. Änderungsanforderungen verantwortlich. Die Schnittstellenfunktion der PM zwischen unterschiedlichen Fachbereichen und den EN erfordert intensive und sorgfältige Abstimmung, da Geschäftsprozesse unterschiedlicher Organisationseinheiten verstanden und verbessert werden müssen. Zusätzlich erschwert eine stark heterogene Anforderungslandschaft den Prozess des RE maßgeblich. Der RE-Prozess wird deutlich verlangsamt, weil ein Einigungsprozess bezüglich der Anforderungen seitens der Mitarbeiter der einzelnen Fachbereiche (KS) durchlaufen werden muss. Das Fehlen von einheitlichen Vorgehensweisen innerhalb des operativen Geschäftes erhöht somit den Abstimmungsaufwand und führt zu Akzidenz. Ziel der PM ist es daher, eindeutig definierte und abgestimmte Prozesse innerhalb der eigens entwickelten Software zu implementieren. Für die im Rahmen dieser Fallstudie erfolgende Darstellung des in der Abteilung praktizierten RE-Prozesses wird die Hauptaufgabe der PM fokussiert, welche als kontinuierliche Weiterentwicklung der Software IDEFIX und dem damit verbundenen RE definiert werden kann.

Die im Mittelpunkt der Betrachtung stehende Software IDEFIX ist eine Anwendung zur abteilungsübergreifenden Abwicklung aller Geschäftskundenprojekte. Mit dem Einsatz und der fortschreitenden Entwicklung der Software werden heute folgende Ziele verfolgt: (1) Dokumentation aller kundenrelevanten Informationen, die für die Projektabwicklung erforderlich sind, (2) Aufteilung eines Projektes in Teilprojekte, (3) Terminverfolgung/Wiedervorlage, (4) Reportingfunktionen und (5) ein bundesweiter Zugriff auf eine zentrale Datenbank. Jeder Geschäftskundenauftrag wird über die Software ausgeführt, was erkennen lässt, dass sich die Software als Standard innerhalb des Unternehmens TK etabliert hat. Am 6. Februar 2008 waren insgesamt 957 Mitarbeiter als aktive IDEFIX-Anwender registriert.

3.3 Der RE-Prozess des Unternehmens TK

IDEFIX kann als Anwendung charakterisiert werden, die in den Jahren seit ihrer Entstehung kontinuierlich weiterentwickelt wurde (E-Typ-Software). Seit dem Pilotprojekt im Jahr 2002 entstanden fünf weitere Major-Releases, in deren Rahmen IDEFIX um grundlegende Funktionalitäten in Form der Module Projektabwicklung, Lagerverwaltung, Auftragserfassung sowie Reportgenerator für das Management erweitert wurde. Aktuell befindet sich das sechste Major-Release in der Entwicklung, mit dem die Subbeauftragung aller an einem Geschäftskundenprojekt beteiligten Organisationseinheiten implementiert werden soll. Hierdurch wird eine einheitliche bzw. zentrale Organisation und Steuerung eines Großkundenprojektes durch die Projektleitung ermöglicht. Folglich fanden bislang sechs Iterationen des RE-Prozesses statt. Aus heutiger Perspektive betrachtet bestätigen beide PM, dass sich aufgrund der Iterationen ein RE-Vorgehen manifestiert hat, bei dem Erfahrungswerte zu einer Verminderung von Akzidenz geführt haben:

„Über die Jahre hinweg hat sich im Zuge der Weiterentwicklung von IDEFIX eine standardisierte Vorgehensweise bzw. Methode entwickelt und etabliert, die uns trotz unserer stark heterogenen Anforderungslandschaft zunehmend erfolgreiches RE ermöglicht.“ (Prozessmanager 1)

Folglich stellt die Beschreibung dieser Methode ein Prozessmuster im Sinne der von Ambler vorgeschlagenen Definition dar, die ein Prozessmuster als Muster charakterisiert, das eine bewährte Vorgehensweise und/oder Folge von Aktivitäten zur erfolgreichen Softwareentwicklung darstellt [Am98]. Tabelle 1 fasst den typischen RE-Prozess zusammen, der für jedes Release iterativ durchlaufen wird, um Benutzeranforderungen „aus den Köpfen der Stakeholder“ (Prozessmanager 2) in ein für die Programmierung adäquates Pflichtenheft zu überführen. Zusätzlich ist jede einzelne Phase anhand der von den PM verwendeten Kommunikationsmaßnahmen gekennzeichnet.

<i>Phase</i>	<i>Beschreibung/Ergebnis</i>	<i>Kommunikationsmaßnahme(n)</i>
(1) Release-definition	<ul style="list-style-type: none"> - Informale und abstrakte Definition der im folgenden Release zu implementierenden Funktionalitäten. (PM und KS) - Anforderungen unvollständig und in rudimentärer Form. - Grobe Schätzung des Anforderungsumfangs (Zeit- u. Personalaufwand). - <i>Ergebnis: Vorstudie</i> 	Face-to-Face (FtF) Interview, Projektmeeting, Email, Telefonat
(2) Analyse Ist-Prozess	<ul style="list-style-type: none"> - Detaillierte Analyse des “Status Quo” im Hinblick auf die neu zu implementierenden Anforderungen. (PM, KS und EN) - <i>Ergebnis: Modell des Ist-Prozesses</i> 	Workshop, FtF-Interview, Projektmeeting, Email, Telefonat, Austausch von Projektdokumentation
(3) Anforderungs-analyse	<ul style="list-style-type: none"> - Diskussion und Definition des Soll-Prozesses (PM, EN und KS). - Vergleich Ist- mit Soll-Prozess und Ableitung von Anforderungen an das Soll-System. - Weitere Anforderungen aus bereits vorliegenden, thematisch kongruenten Benutzeranforderungen, die angepasst, verfeinert und in das Gesamtkonzept integriert werden. - <i>Ergebnis: Detailliert beschriebener Soll-Prozess inklusive der Anforderungen (1. Teil Lastenheft)</i> 	Workshop, Email, Telefonat, Austausch von Projektdokumentation, schriftliche Kommunikation über Lastenhefterstellung
(4) System-analyse	<ul style="list-style-type: none"> - Ableitung der zu implementierenden IT-Anforderungen aus dem zuvor definierten Soll-Zustand. (PM und EN) - Erstellung des fachlichen Datenmodells. - Spezifizierung weiterer Anforderungen, um Lücken bei bisherigen Benutzeranforderungen zu füllen. - Identifizierung notwendiger Softwareanpassungen am Ist-System, um neue Anforderungen in die Systemlandschaft integrieren zu können. - <i>Ergebnis: Detailliert beschriebene Anforderungs-spezifikation (2. Teil Lastenheft)</i> 	Email, Telefonat, schriftliche Kommunikation über Lastenhefterstellung

(5)	<ul style="list-style-type: none"> - Das Lastenheft ist von allen am RE-Prozess beteiligten KS freizugeben. 	Projektmeeting, Email,
Review des Lastenhefts	<ul style="list-style-type: none"> - Vollständigkeit des Lastenheftes, adäquate Umsetzung und Funktionalität werden seitens aller Beteiligten geprüft und bestätigt. - Bei Fehlern sind Rücksprünge bis Phase 3 möglich. - <i>Ergebnis: Abgestimmtes und von allen KS freigegebenes Lastenheft</i> 	Telefonat, schriftliche Kommunikation über Lastenhefterstellung
(6)	<ul style="list-style-type: none"> - Erstellung von Pflichtenheft und komplettem Systementwurf auf Basis der durch das Lastenheft vorliegenden Systemmodellbeschreibung. (EN) - Transformation im engen Dialog mit den PM, um auftretende Verständnis- oder Implementierungsprobleme umgehend beseitigen zu können. - In Abhängigkeit von der Schwere eines in diesem Prozess auftretenden Problems sind Rücksprünge bis Phase 4 möglich. - <i>Ergebnis: Konsistentes Pflichtenheft</i> 	Projektmeeting, Email, Telefonat

Tabelle 1: Das RE-Prozessmuster

Eine detaillierte Beschreibung der innerhalb des RE-Prozesses angewendeten Kommunikationsmaßnahmen sowie die jeweilige Verwendungshäufigkeit werden in Tabelle 2 abgebildet. Die Darstellung erlaubt die Untersuchung der Kommunikationsmaßnahmen für jedes der durchgeführten Releases im Hinblick auf die Frequenz ihrer Verwendung und veranschaulicht im Zeitablauf stattfindende Veränderungen.

Kommunikationsmaßnahme	Jahr/Release-Nr.		2002	2003	2004	2005	2006	2008
	R1	R2	R3	R4	R5	R6		
FtF-Interview								
- Teilnehmerkreis: PM und KSH; klein, vorwiegend <2 zu 1.	+	+	+	+	+	+		
- Räumliche Nähe, d.h. FtF-Situation, zeitliche Unmittelbarkeit.								
- Vorwiegend mündlich, teilw. schriftliche Dokumentation.								
Projektmeeting								
- Teilnehmerkreis: PM und DE; groß, vorwiegend >4 zu >4.	o	o	o	o	o	o		
- Räumliche Nähe, d.h. FtF, zeitliche Unmittelbarkeit.								
- Vorwiegend mündlich, teilw. schriftliche Dokumentation.								
Workshop								
- Teilnehmerkreis: PM und KSH; groß, vorwiegend >4 zu >10.	-	+	+	+	+	+	+	
- Räumliche Nähe/FtF-Situation, zeitliche Unmittelbarkeit.								
- Vorwiegend mündlich, teilw. schriftliche Dokumentation.								
Telefonat								
- Teilnehmerkreis: paarweise PM, KSH und DE; klein, vorwiegend 1 zu 1.	+	+	+	+	+	+	+	
- Räumliche Entfernung, zeitliche Unmittelbarkeit.								
- Vorwiegend mündlich, teilw. schriftliche Dokumentation.								
Email								
- Teilnehmerkreis: PM, KS und EN; flexibel, 1 zu n.	+	+	+	+	+	+	+	
- Räumliche Entfernung und zeitlicher Abstand.								
- Schriftlich.								
Austausch von Projektdokumentation								
- Teilnehmerkreis: PM, KSH und DE; flexibel, 1 zu n.	+	+	+	+	+	+	+	
- Räumliche Entfernung und zeitlicher Abstand.								
- Schriftlich.								
Kommunikation über Lastenhefterstellung								
- Teilnehmerkreis: PM, KSH und DE; flexibel, 1 zu n.	-	o	+	+	+	+	+	
- Räumliche Entfernung und zeitlicher Abstand.								
- Schriftlich.								

- = nie; o = selten; + = oft

Tabelle 2: Beschreibung und Verwendungshäufigkeit der Kommunikationsmaßnahmen

Um die Charakteristika der verschiedenen Kommunikationsmaßnahmen zu identifizieren und somit den mit ihrem Einsatz verbundenen Nutzen innerhalb des RE-Prozesses im Hinblick auf effiziente Kommunikation und die Bildung von sprachlichem Konsens zu begutachten, fasst Tabelle 3 die Bewertung aller eingesetzten Kommunikationsmaßnahmen durch die PM zusammen. Die Bewertung erfolgt anhand der zuvor genannten fünf Medienfaktoren.

Kommunikationsmaßnahme	Geschwindigk. des Feedbacks	Symbolvarietät	Parallelität	Überarbeitbarkeit	Wieder verwendbarkeit
FtF-Interview	hoch	niedrig - hoch	niedrig	niedrig	niedrig
Workshop	hoch	niedrig - hoch	hoch	niedrig - mittel	niedrig - mittel
Projektmeeting	hoch	niedrig - hoch	hoch	niedrig - mittel	niedrig - mittel
Telefonat	hoch	niedrig	niedrig	niedrig	niedrig
Email	mittel	niedrig - hoch	hoch	hoch	hoch
Projekt-dokumentation	niedrig	hoch	hoch	hoch	hoch
Lastenheft	niedrig	hoch	mittel	hoch	hoch

Tabelle 3: Bewertung der im TK-RE-Prozess eingesetzten Kommunikationsmaßnahmen anhand der Medienfaktoren durch die PM

4 Analyse und Interpretation der Ergebnisse

4.1 Analyse der Kommunikationsmaßnahmen

Die eingesetzten Maßnahmen wurden bezüglich ihrer Auswirkungen auf die Kommunikation und des Erreichens sprachlichen Konsenses bewertet. Tabelle 1 lässt eine Intensivierung der verwendeten Kommunikationsmaßnahmen in den Phasen zwei und drei erkennen. Dies kann dadurch erklärt werden, dass in diesen Phasen der kommunikationsintensive Informationsaustausch zwischen PM und KS stattfindet. Zur effizienten und adäquaten Abwicklung des aufgrund heterogener Anforderungen notwendigen und komplexen Einigungsprozesses ist der verstärkte und parallele Einsatz unterschiedlicher Kommunikationsmaßnahmen unerlässlich. Des Weiteren wird deutlich, dass während jeder Phase mittels Email und Telefon kommuniziert wird. Gründe hierfür werden in Abschnitt 4.2 spezifiziert.

Bei Untersuchung der in Tabelle 2 dargestellten Verwendungshäufigkeit der eingesetzten Kommunikationsmaßnahmen ist zu erkennen, dass Email, Telefonat, FtF-Interview und der Austausch von Projektdokumentation seit 2002 feste und häufig verwendete Bestandteile des RE-Prozesses von TK bilden. Der Austausch von Projektdokumentation dient hierbei primär der schriftlichen Fixierung und somit dem Versuch der Konsensbildung bei noch nicht einheitlich definierten Anforderungen. Zusätzlich erfüllt die Projektdokumentation nach Meinung der PM die Funktion einer „Korrekturschleife“ (Processmanager 2). Einerseits dient sie der Konsolidierung und andererseits erlaubt sie allen

am RE-Prozess beteiligten Personen zu überprüfen, ob der erreichte Konsens tatsächlich ihren Vorstellungen entspricht.

Wurden in 2002 noch keine Workshops mit Beteiligung der PM, EN und KS durchgeführt, begriffen die PM bereits während des zweiten Releases, dass die vielfache Verwendung von Workshops vor allem in den Phasen zwei und drei von enormer Wichtigkeit ist. Grund hierfür ist das Vorhandensein unterschiedlicher sprachlicher Konzepte und operativer Vorgehensweisen innerhalb des Unternehmens TK. Nach Meinung der PM lässt sich mit jedem der durchgeführten Workshops ein allmählich entstehender sprachlicher Konsens feststellen, was auch damit zusammenhängt, dass der Konsens „*von oben*“, also von Seiten der PM oder der Abteilungsleiter, herbeigeführt wird. Zudem dienen Workshops als allgemeine Kommunikationsplattform zur Klärung von Erwartungen, Fragen und Konflikten.

Projektmeetings wurden über alle Releases hinweg nur selten durchgeführt. Ihre Ausführung wird von beiden PM als „*schwierig und holprig*“ bewertet. Problematisch waren auch aus Kommunikations- und sprachlicher Perspektive die häufig wechselnden KS zwischen und innerhalb der Releases und somit das Fehlen einer festen Projektmitgliederstruktur. Für solche Situationen stellen Workshops nach Erfahrung der PM die adäquate Kommunikationsmaßnahme dar. Des Weiteren werden Projektmeetings in den Phasen fünf und sechs nur nötig, wenn Probleme auftreten, die Rücksprünge in eine der vorherigen Phasen notwendig machen.

Schriftliche Kommunikation über die Lastenhefterstellung findet in den Phasen drei, vier und fünf statt. Das Lastenheft wird hierbei als Kommunikationsmaßnahme verstanden, über das vor allem während des Review- und Freigabevorgangs die finale Abstimmung aller Beteiligten erfolgt. Dieses fand erstmals im zweiten Release statt. Die Tatsache, dass dies in 2003 nur selten, ab 2004 allerdings häufig geschah, ist auf das zu diesem Zeitpunkt noch zu schwach ausgeprägte gemeinsame Vokabular aller beteiligten Akteure zurückzuführen. Es war nur schwer möglich, sich in schriftlicher und derart formeller Weise bezüglich der Inhalte des Lastenheftes umgehend abzustimmen. Unkomplizierte Telefonate oder Emails erscheinen hierfür gerade bei Zeitdruck als die besseren Alternativen.

Ergänzend zu den genannten Maßnahmen wird von beiden PM die Relevanz von informellen Kommunikationsmaßnahmen betont. Unter informellen Kommunikationsmaßnahmen ist ungeplante bzw. zufällige Kommunikation in Form von Zwischengesprächen am Rande von Meetings, Gespräche bei der Kaffeepause oder in der S-Bahn usw. zu verstehen. Über alle Phasen des RE-Prozesses hinweg, aber insbesondere zwischen Phase 2 und 5 haben informelle Kommunikationsmaßnahmen entscheidend dazu beigetragen, die Konsolidierung der Anforderung herbeizuführen.

4.2 Interpretation anhand der Medienfaktoren

Anhand der fünf Medienfaktoren (Tabelle 3) werden im Folgenden die in den einzelnen Phasen des RE-Prozesses eingesetzten Kommunikationsmaßnahmen im Hinblick auf ihre Fähigkeit interpretiert, effiziente Kommunikation zu ermöglichen.

Geschwindigkeit des Feedbacks: Die PM bewerten Email, Telefonate und FtF-Interviews in Bezug auf ihre zeitliche Unmittelbarkeit (mittel bis hohe Geschwindigkeit des Feedbacks) vorwiegend positiv. Die Vorteile der Kommunikationsmaßnahme Email sind ihre kurze und kompakte Darstellung und der flexible Verteilerkreis. FtF-Interviews und Telefonate zeichnen sich insbesondere aufgrund der direkten Möglichkeit der Rückmeldung, der dadurch erleichterten Findung sprachlichen Konsenses und der Vermeidung von Missverständnissen aus. Eine hohe Feedbackgeschwindigkeit, insbesondere bei den Maßnahmen Telefon, Projektmeeting und FtF-Interview, kann zunächst als relevant für die sprachliche Konsensbildung bewertet werden, denn hierdurch können alle am RE-Prozess beteiligten Personen direkt auf die Äußerungen der anderen reagieren und Missverständnisse oder Unklarheiten in Bezug auf Konzepte oder Symbole sofort klären.

Dennoch kann das Vorhandensein einer hohen Feedbackgeschwindigkeit nicht automatisch als alleiniger Erfolgsfaktor bewertet werden, denn gerade bei Kommunikationsmaßnahmen, bei deren Durchführung mehrere Personen beteiligt sind (Workshops, Projektmeetings), erhöht sich nach Meinung der PM die Komplexität der Konsensfindung. Die Ursache hierfür liegt in hoher *Parallelität*. Zusätzlich kann sich eine hohe Feedbackgeschwindigkeit negativ auf die Bildung von Kohärenz in der Verwendung von Sprache auswirken, da schnelle Rückmeldung eher die Verwendung eigener Sprache fördert und somit die Einhaltung von in der Sprachgemeinschaft vereinbarten Konventionen erschwert.

Parallelität: Wie im Abschnitt *Geschwindigkeit des Feedbacks* bereits angesprochen, erscheint eine hohe Parallelität in Interaktionssituationen mit einem großen Teilnehmerkreis als problematisch bezüglich der Konsensfindung. Die Anzahl an gleichzeitig ablaufenden Kommunikationsvorgängen ist bei Workshops und Projektmeetings hoch. Mehrere Personen haben die Möglichkeit, sich in das Gespräch einzubringen. Jedoch gelingt es nicht jedem, die seiner Meinung entsprechende Rückmeldung zu platzieren. Daher ist eine ausgewogene Moderation der Gespräche zur Sicherstellung eines effektiven Gesprächverlaufs entscheidend, da somit alle Personen an der Konsensfindung beteiligt werden können.

Überarbeitbarkeit: Die Überarbeitbarkeit spielt bei der Etablierung von Kohärenz bezüglich sprachlicher Konzepte und Symbolen eine entscheidende Rolle, da hierdurch der Anpassungsprozess an bereits erreichten Konsens unterstützt werden kann. In diesem Zusammenhang ist die Überarbeitung von Projektdokumentationen und des Pflichtenheftes zu betonen, die dabei hilft, Missverständnisse in der Kommunikation zu vermeiden, Inkohärenzen zu identifizieren und zu klären. Aufgrund der niedrigeren Feedbackgeschwindigkeit kann eine hohe Überarbeitbarkeit vor allem bei den schriftlichen Kommunikationsmaßnahmen Projektdokumentation, Lastenheft und Email beobachtet werden.

Wiederverwendbarkeit: Die Wiederverwendbarkeit schriftlicher Unterlagen hilft dabei, sich bereits erreichten Konsens gegenwärtig zu machen. Zwar wurde die Dokumentation insbesondere als zeitintensive Pflicht angesehen, im Falle von Missverständnissen wurde jedoch die Relevanz der Wiederverwendbarkeit als konsensbewahrendes Mittel vor allem bei Email, Projektdokumentation und Lastenheft deutlich.

Symbolvarietät: Eine hohe Symbolvarietät bei Email, beim Dokumenta tionsaustausch und beim Lastenheft wird als positiv bewertet, da diese es ermöglicht, Inhalte in unterschiedlicher Form zu illustrieren. Insbesondere bei Projektdokumentation und Lastenheft erstellung lässt sich eine hohe Symbolvarietät beobachten.

4.3 Abschließende Interpretation

Aufgrund der Analyse der Medienfaktoren in Zusammenhang mit den Konzept der Konsensbildung wurde für den Einsatz von Kommunikationsmaßnahmen im RE-Bereich insbesondere deutlich, dass die Stärke schriftlicher Kommunikationsmaßnahmen (Projektdokumentation, Lastenheft) in der Gewährleistung einer hohen Überarbeitbarkeit und Wiederverwendbarkeit innerhalb des Abstimmungsprozesses liegt. Sie ermöglichen die Errichtung sprachlichen Konsenses über die unterschiedlichen RE-Phasen hinweg und bieten die Chance, Kohärenz im Prozess zu gewährleisten. Zudem sind mündliche Kommunikationsmaßnahmen (Interview, Workshops, Telefonate) in der Lage, eine hohe Feedbackgeschwindigkeit sicherzustellen. Dies ist für eine zügige Klärung von Missverständnissen entscheidend, um Konsensprobleme zu erkennen, zu lösen oder zu vermeiden.

Die Kommunikationsmaßnahme Email stellt hierbei eine Art Mischform dar, da sie einerseits eine hohe Wiederverwendbarkeit und Überarbeitbarkeit, andererseits eine mittlere Feedbackgeschwindigkeit ermöglicht und sich somit von den anderen schriftlichen Kommunikationsmaßnahmen unterscheidet, indem sie sich sowohl auf die Konsenserhaltung als auch auf die Klärung von Missverständnissen positiv auswirkt.

Insgesamt wurden über die Zeit die angewendeten Kommunikationsmaßnahmen anhand ihres speziellen Nutzens eingesetzt und geprägt, wodurch Akzidenz innerhalb des RE-Prozesses weitestgehend minimiert werden konnte.

5 Fazit und Ausblick

Die erfolgreiche Überführung des TK-RE-Prozesses in ein Prozessmuster und die Analyse der in den Phasen des Prozessmusters eingesetzten Kommunikationsmaßnahmen haben gezeigt, dass das Erreichen sprachlichen Konsenses und die damit verbundene Sicherstellung einer erfolgreichen Kommunikation maßgeblich zur Reduktion der durch Akzidenz verursachten Komplexität beitragen können. Innerhalb des Unternehmens TK lässt sich eine über die Zeit stattfindende Entwicklung beobachten, während der Kommunikation und sprachliche Konsensfindung kontinuierlich ausgebaut, verbessert und professionalisiert werden konnten, was im Wesentlichen mit der erfolgreichen Anpassung und Abstimmung der innerhalb des RE-Prozesses verwendeten Kommunikationsmaßnahmen zusammenhängt.

Dabei ist schnelle Kommunikation bzw. sofortiges Feedback in Form von Email, Telefonaten und Workshops als besonders wichtig zu bewerten. Hierdurch lassen sich Unklarheiten bezüglich sprachlicher Konzepte oder Symbole schnell und direkt zwischen

den am RE-Prozess beteiligten Personen ohne Umwege klären. Zusätzlich müssen diese Maßnahmen durch ausführlichen Austausch von Projektdokumentation und einer in Form eines Lastenheftes detailliert erfolgende Beschreibung der Ergebnisse bzw. der Anforderungen ergänzt werden. Wichtige Voraussetzung hierfür ist, dass die verwendeten Kommunikationsmittel im Unternehmen etabliert und von allen Beteiligten akzeptiert und eingesetzt werden. Zusätzlich ist die enorme Relevanz der informalen Kommunikationsmaßnahmen für ein erfolgreiches RE zu betonen. Diese sowie auch die Kommunikationsmaßnahmen mit schneller Rückmeldungsmöglichkeit erlauben es den PM, ein der Essenz der IDEFIX-Software angemessenes, effektives RE durchzuführen und gleichzeitig die Akzidenz zu minimieren.

Aufgrund der vorliegenden Ergebnisse lässt sich abschließend festhalten, dass eine optimale Gestaltung des Kommunikationssettings und eine Sensibilisierung der Mitarbeiter für die konsensbezogenen Herausforderungen, welche durch die unterschiedlichen Kommunikationsmaßnahmen angesprochen werden können, auch in anderen RE-Settings Anwendung finden können. Es bleibt festzustellen, dass die Erkenntnisse vor allem für Unternehmen oder RE-Settings Gültigkeit besitzen, die durch eine stark heterogene Anforderungslandschaft in Folge fehlender Prozessstandardisierung gekennzeichnet sind. Ferner wird deutlich, welche Faktoren der Kommunikation durch bestimmte Kommunikationsmaßnahmen unterstützt werden und wie in einer konkreten Kommunikationslandschaft Kommunikationsmaßnahmen verwendet werden können, um den Anforderungen des RE gerecht zu werden. Inwiefern die Ergebnisse auf weitere Individualsoftwareentwicklungsprojekte übertragen werden können, wird anhand weiterer Fallstudien zu untersuchen sein.

Literaturverzeichnis

- [Am98] Ambler, S. W.: Process Patterns: Building Large-Scale Systems Using Object Technology. Cambridge University Press, 1998.
- [Be02] Berry, D. M.: The Inevitable Pain of Software Development: Why There Is No Silver Bullet. In: Proceedings of the 9th International Workshop on Radical Innovations of Software and Systems Engineering in the Future (RISSEF 2002) Venice, Italy, 2002, S. 50-74.
- [BEJ06] Buschermöhle, R.; Eekhoff, H.; Josko, B.: SUCCESS Erfolgs- und Misserfolgsfaktoren bei der Durchführung von Hard- und Software-Entwicklungsprojekten in Deutschland. BIS-Verlag - Carl von Ossietzky Universität: Oldenburg, 2006.
- [Bo79] Boland, R. J. J.: Control, Causality, and Information System Requirements. Accounting, Organization and Society 4(4), 1979: S. 259-272.
- [Br87] Brooks, F. P.: No Silver Bullet : Essence and Accident in Software Engineering. IEEE Computer 20(4), 1987: S. 10-19.
- [Br95] Brooks, F. P.: The Mythical Man-Month: Essays on Software Engineering. Anniversary Edition. Auflage Addison-Wesley: Reading, MA, 1995.
- [DV99] Dennis, A. R.; Valacich, J. S.: Rethinking Media Richness: Towards a Theory of Media Synchronicity. In: Proceedings of the Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.
- [Eb05] Ebert, C.: Systematisches Requirements Management: Anforderungen ermitteln, spezifizieren, analysieren und verfolgen. Heidelberg, 2005.
- [Ei89] Eisenhardt, K. M.: Building Theories from Case Study Research. Academy of Management Review 14(4), 1989: S. 532-550.
- [HKN91] Hirschheim, R.; Klein, H.; Newman, M.: Information Systems Development as Social Action: Theoretical Perspective and Practice. OMEGA 19(6), 1991: S. 587-608.
- [Ho07] Holten, R.: Deriving an IS-Theory from an Epistemological Position. In: Proceedings of the 18th Australasian Conference on Information Systems (ACIS) Toowoomba, Queensland, Australia, 2007.
- [Ja94] Jackson, M. A.: The Role of Architecture in Requirements Engineering. In: Proceedings of the Proceedings of the IEEE International Conference on Requirements Engineering. IEEE Computer Society Press: Colorado Springs, 1994, S. 241.
- [KL03] Kavakli, E.; Loucopoulos, P.: Goal Driven Requirements Engineering: Evaluation of Current Methods. In: Proceedings of the EMMSAD '03 Velden, Austria, 2003.
- [KL96] Kamlah, W.; Lorenzen, P.: Logische Propädeutik. Vorschule des vernünftigen Redens. 3. Auflage Verlag J.B. Metzler: Stuttgart, Weimar, 1996.

- [KMR00] Keil, M.; Mann, J.; Rai, A.: Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models. *MIS Quarterly* 24(4), 2000: S. 631-664.
- [Le80] Lehman, M. M.: Programs, Life Cycles and Laws of Software Evolution. In: Proceedings of the IEEE, 1980, S. 1060-1076.
- [Le96] Lehman, M. M.: Laws of Software Evolution Revisited. In: Proceedings of the Software Process Technology - Proceedings of the 5th European Workshop. Springer Verlag: Nancy, France, 1996, S. 108–124.
- [LR01] Lehman, M. M.; Ramil, J. F.: Evolution in Software and Related Areas. In: Proceedings of the Proceedings of the 4th International Workshop on Principles of Software Evolution. International Conference on Software Engineering.Vienna, Austria, 2001, S. 1-16.
- [Ly85] Lyytinen, K. J.: Implications of Theories of Language for Information Systems. *MIS Quarterly* 9(1), 1985: S. 61-76.
- [Ly98] Lyytinen, K.; Keil, M.; Cule, P.; Schmidt, R.: A framework for identifying software project risks. *Communications of the ACM* 41(11), 1998: S. 76 - 83.
- [NE00] Nuseibeh, B.; Easterbrook, S.: Requirements engineering: A roadmap. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000). ACM Press: Limerick, Ireland, 2000, S. 35-46.
- [Ro89] Rorty, R.: Contingency, Irony, and Solidarity. Cambridge University Press, 1989.
- [Sa74] de Saussure, F.: Course in General Linguistics. Peter Owen Ltd.: London, UK, 1974.
- [St01] Standish Group International, I.: Extreme CHAOS. Research report, ordering information available at www.standishgroup.com, 2001.
- [Su95] Suchman, L.: Making work visible. *Communications of the ACM* 38(9), 1995: S. 56-64.
- [SW95] Sperber, D.; Wilson, D.: Relevance: communication and cognition. 2. Auflage Blackwell: Oxford, 1995.
- [We81] Wedekind, H.: Datenbanksysteme I. Eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung. 2. Auflage Mannheim, Germany et al., 1981.
- [WO92] Weltz, F.; Ortmann, R.: Das Softwareprojekt: Projektmanagement in der Praxis. Frankfurt, 1992.
- [Yi03] Yin, R. K.: Case Study Research: Design and Methods. 3. Auflage SAGE Publications: Thousand Oaks, CA, USA et al., 2003.
- [Za97] Zave, P.: Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys* 29(4), 1997: S. 315-321.

Using Fuzzy Process Models to Improve Technical Customer Services: A Case Study for Heating Facilities

Oliver Thomas¹, Philipp Walter¹, Thorsten Dollmann¹,
Peter Loos¹, Michael Schlicker²

¹Institute for Information Systems (IWi)
at the German Research Center for Artificial Intelligence (DFKI),
Saarland University, Saarbruecken (Germany)
{oliver.thomas|philipp.walter|thorsten.dollmann|peter.loos}@iwi.dfki.de

²Interactive Software Solutions GmbH,
Saarbruecken (Germany)
michael.schlicker@interactive-software.de

Abstract: In this article, a mobile, internet-based service tool for the support of technical customer service processes is described. The tool uses a fuzzy component structuring repair information formatted for multimedia use and makes this information available to customer service technicians through the process-oriented use of mobile end devices. Through the use of mobile technologies in connection with the knowledge of experienced customer service technicians with respect to inspection, maintenance and servicing work and through the processing of this knowledge in accordance with fuzzy rules, the efficiency of the service process can be increased. The potential use of this system is exemplified by means of a real-life application from the heating, air conditioning and sanitary engineering (HAS) branch.

1 Introduction

1.1 Objective and Approach

Under the heading of business process management, the planning, controlling and monitoring of the processes carried out when rendering services within the structure of a company are acknowledged as being significant instruments for corporate management both in theory and in practice. Through the use of standardized procedures and integrated software tools in the IT-oriented structuring, implementation and monitoring of business processes, the knowledge contained within an organization can, in many cases, be made available successfully.

One major factor that causes an increase in the complexity of processes within companies is the integrated examination of an rising number of process steps, available customer data and wide-ranging product variants, all of which need to be taken into consideration. This calls for appropriate management methods and supportive software tools. The rising proportion of intangible resources also causes changes in the quality of business processes. The role played by knowledge-related work and services in the process of value creation is constantly increasing. For over 30 years now, the process of

value creation in “industrialized nations” has been attributable more to the supply of services, than to the production of material goods and the percentage is currently around 70% [WR07].

The machinery and plant construction sector is Germany’s largest industrial sector [VDMA08]. In response to an increase in competition, manufacturers of machinery and plants are expanding and improving the services they provide, particularly in the field of technical customer services (TCS), the interface between the manufacturing of products and their actual use [LaLo76; Czep80; Peel87; StLa89; BoKL06; Harr07].

In order to adequately perform the tasks associated with these services, a technical customer services unit must be provided with the correct “information mix”. In this respect, a central problem is determining the scope and point in time of the information relevant to the decision-making process, together with the degree to which it has been compressed [SaBo03; Timm05]. Current approaches to providing support for TCS frequently fail because of the increased complexity of the machinery and the associated increase in the requirement for representation of the service supply processes. The consequences are defective start-up, maintenance and repair work, which results in prolonged down times for machinery leading to higher prices for customers and a loss of market shares for the manufacturer.

The circumstance described above is, initially, counteracted by integrating the manner in which physical products and service-related modules of information are taken into consideration and by bringing together these two production factors to create efficient and mobile service processes that are made available to the TCS. Thanks to the creation of a new bundle as a result of this integrated study, consisting of one material product plus services, it is possible to provide the TCS unit with all that is necessary to ensure that the start-up, servicing, maintenance and repair of machinery and plant are carried out in a customer-friendly manner.

1.2 Our Contribution

The contrast between the construction of models for human behavior and the mechanical processing of information clashes with requirements for the continuous support of planning functions, from managerial planning to the implementation on an information technology level of business processes. The depiction of the resources available during both thinking and acting and the provision of IT-support for decision-making processes and interaction, call for an extension of the range of instruments available for the modeling process, which incorporates the specific characteristics of human thought. In this article, the Event-driven Process Chain (EPC) [KeNS92] is used as a representative of the many available specialist conceptual modeling methods.

One strategy for resolving the issue of reducing the complexity occasioned by the characteristics of human ways of thinking and for increasing the extent to which computer-assisted systems can be controlled, albeit one that has to date been used only to a limited extent, is provided by technologies from the field of Artificial Intelligence (AI). One sub-category of AI research that has been highly successful in providing explanations of human ways of thinking for use in computer systems can be found in the

methods from the fuzzy set theory, artificial neural networks and evolutionary algorithms, all of which come under the general heading of soft computing [Zade94; Zade97; Boni97]. The central finding of soft computing technologies is the transfer of the human capacity for processing statements that are imprecise, uncertain and endowed with truth values on a sliding scale. Here, the explication of this vagueness takes place on the basis of precise mathematical models, which makes it possible for this to be reproduced within information systems. Using the fuzzy set theory, conclusions can be reached on the basis of unclear or incomplete data. Decision-making patterns can, with the assistance of artificial neural networks, be learned from data and emulated by means of applying them to fresh data. New and partly creative problem-solving techniques can be found by evolutionary algorithms.

In this article, we introduce the idea of a process-oriented service tool for the support of technical customer service processes in the machinery and plant construction sector. This tool makes use of a fuzzy component for structuring repair information formatted for multimedia use and makes this information available to customer service technicians through the use of mobile end devices. Through the use of mobile technologies in connection with the explication of knowledge from experienced customer service technicians with respect to inspection, maintenance and servicing work and through the processing of this knowledge in accordance with fuzzy rules, the efficiency of the service process can be increased. The potential use of this system is exemplified on a use case from the heating, air conditioning and sanitary engineering (HAS) branch.

2 Technical Service Processes in Technical Customer Services

2.1 The Information Basis for Service Processes

The core idea for the service process modeling that forms the basis of the approach pursued here is the differentiation between the function, structure and service processes in technical facilities [TWLN07]. Both the functions of a technical facility and its service processes are directly connected with its structure. Figure 1 shows how this interrelationship is reproduced and used with the function structure being used as the basis for defining the fault, identifying the groups of components relevant for the performance of that function and suggesting appropriate service processes.

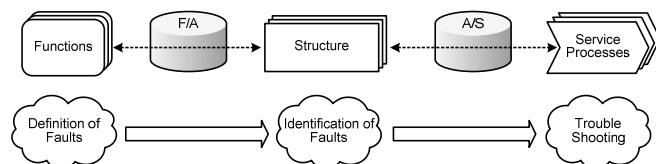


Figure 1: Functions, structure and service processes of technical facilities [TWLN07]

Functions The modeling of the functions in a technical facility forms the basis for the definition of a problem, which can be seen as the “breakdown of a function”. As opposed to definitions of potential problems, the functions of a technical facility can be

listed and classified into technical groups of components, which will eventually serve as points of reference for the work of the TCS unit.

Groups of components The technical classification of the facility into various groups of components stands in contrast to the functional classification process. This technical classification arises out of technical product development. The interlinking of groups of components with functions takes place on the basis of the technical sharing of tasks on a technical level – a group of components is assigned to the functions for which it is needed. The objective here is to identify groups of components that may potentially be the cause of any given problem.

Service processes The objective of the diagnosis of a problem is the rectification of that problem, which in turn is achieved by means of carrying out service processes on the groups of components of the technical facility. This is made possible by interlinking service processes with groups of components on a range of different levels of detail. Through the composition of multiple groups of components, functions that go beyond the sum of the individual functions of these groups of components emerge. A heater, for example, can heat water, which is something none of its groups of components can accomplish on their own.

Once particular groups of components have been identified as the possible sources of the problem using the functional classification described above, measures for the rectification of this problem can then be selected and implemented via the link with the service processes.

2.2 An Example of a Service Process Model

A realistic application scenario follows as an illustration of the approach described above. This scenario describes a fault in an appliance for heating water. This appliance is found to be defective by the customer service technician at the customer's premises. The starting point for the scenario is the fault "Hot water is not hot" – in other words, a problem has occurred within the appliance that interferes with its function and as a result the water inside the appliance is no longer being heated as intended. The handling of this repair process places great demands upon the TCS unit, because almost every component of this heating installation can be a potential cause for this fault. This real-life scenario is based upon a heating high efficiency system boiler, ecoTec plus 196/3-5 type, from Vaillant Deutschland GmbH & Co.KG.

In preparation for repairing the defect, the manufacturer must start by making the necessary service process information available for the product that has become defective. This service information includes the functional, product and service process structure, together with the service process models, which should be accompanied by the relevant technical documentation. Within the framework of this scenario, a service process emerges that is portrayed in part in Figure 2. The links between the models are realized partly by means of "process signposts" that are contained within the expanded scope of the language of the EPC. For the sake of simplicity, we have decided not to separate all the various functional refinements into their component parts. They are indicated alongside the corresponding EPC functions by the symbol "■".

It is possible to raise specific criticisms of this service process model of the rectification of the fault, “Hot water does not heat up”, thus calling into question whether it should be adopted without due reflection for the purposes being pursued in this sample application. These points of criticism are analyzed in detail below and combined in the form of a statement of the objectives of any further action.

2.3 A Discussion of the Service Process Model

Figure 2 initially makes the high level of complexity in representing service processes through modeling clear. Using systems theory, it is possible to state that in relation to service process models, this is a consequence of the large number of speech artifacts required to described the process, as well as a result of their variability, which is in turn geared towards the wide variety of machinery and plants available from different manufacturers. For instance, in the present case study a wide variety of activities and decision-making nodes have to be modeled, although only a few components can be identified as the original cause of the fault, “Hot water is not hot”.

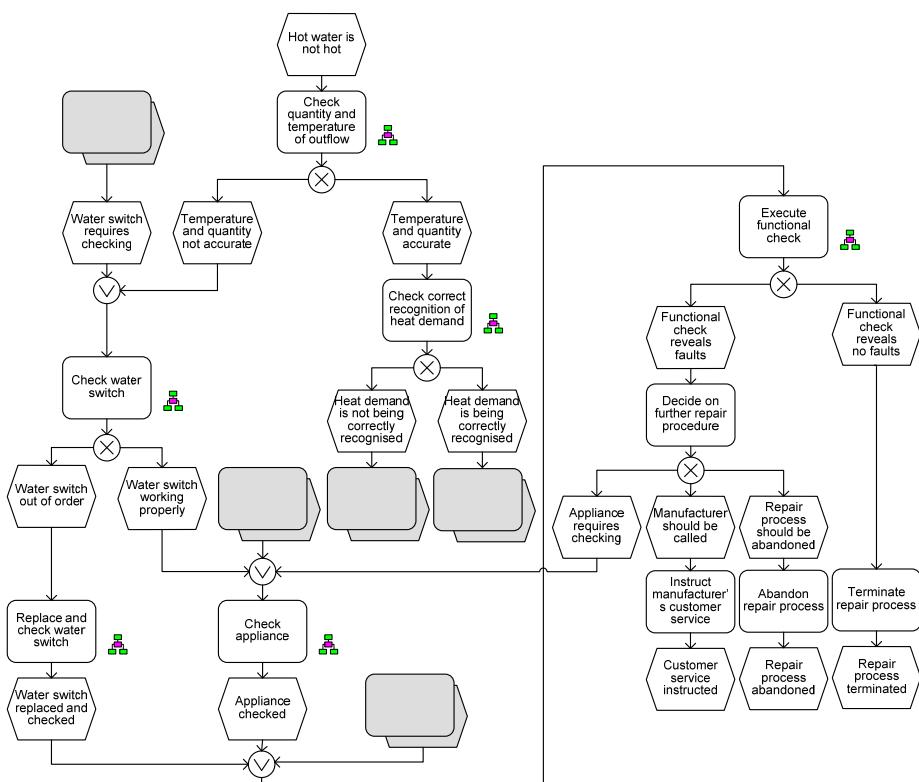


Figure 2: EPC model “Hot water is not hot” (in part)

As is customary in process modeling, the designators for model elements used in the service process model “Hot water is not hot” closely follow technical terminology, in

order to ensure that the model can be used as a starting point for the implementation of the application systems. It is true that this close imitation of existing technical terminology is supposed to reduce the risk of any misinterpretation – however, misunderstandings and false conclusions arising out of the ambiguity inherent in natural speech cannot be wholly excluded. In the present model (c.f. Figure 2), it is part of the run-time that the quantity and temperature of the outflow should be checked at the beginning of the service process. Then further activities on the part of the TCS unit are envisaged in order to rectify this fault depending on the two mutually exclusive possible outcomes of this test. Admittedly, the actual meaning of the word “correct” in relation to quantities of water discharged or temperature value is not directly apparent from the context here. If the model is being consulted solely as a means of recommending a course of action for the provision of services by the customer service department, then this does not present a problem. A customer services technician will, by virtue of his knowledge and experience, generally be in a position to interpret the measured or estimated values as correct or incorrect.

If, however, the representation of the process is intended to serve as the basis for the implementation of a mobile application system then the statements present in the model must be formalized, in order to allow them to be processed automatically.

To ensure the success of a mechanical processing of this kind it is, for example, possible for appliance-specific quantitative threshold values or ranges to be defined for the values to be measured. The decisions to be made during the service, maintenance or repair procedure are then made with reference to these specifications, depending on whether the given threshold values are exceeded or fallen short of, or on whether or not the values measured fall within a prescribed range. For example, in the case of the combination water heater with flue attachment for heat and hot water on which this application is based, the rate of flow is 5.5 liters per minute at an outflow temperature of 60°C (corresponds to 140°F). If the ranges for the determination of a correct combination of outflow quantity and temperature were to be set at 4–5 liters per minute and 55–65°C, then the measurement of 54.9°C would be deemed incorrect, regardless of the outflow quantity measured. This contradicts actual customer service practice where such exclusion criteria are seldom adhered to. In fact, the customer service technician applies implicit compensation mechanisms that balance out any exceedance of the threshold values with better values in other areas. The rules for these interdependent cause-effect relationships are, in general, not documented and are based upon the knowledge and experience of customer service technicians.

Setting aside the problems posed by sharply-defined threshold values, in the EPC model “Hot water is not hot” the sequence of checking functions and replace-and-check functions was determined at the time of the appliance’s construction. However, during the actual run-time of the application system intended to provide support for the provision of these services, this static perspective on the activities required from the TCS unit during a service process is not desirable. Thus, during the process of service, maintenance and repair it is entirely normal to find situations in which the rectification of a defect may become directly apparent to the experienced customer service operative by means of measured or estimated values. If the TCS unit were to orient itself

exclusively on the inflexible specifications provided by the process model this would, on occasion, lead to the implementation of work processes not necessarily conducive to the identification or rectification of the defect in question. To date, the service process model has not included any representation of the dynamic needed in the sequence of checking functions if superfluous activities are to be avoided.

3 Support for the Provision of Services through Fuzzy Process Modeling and Controlling

3.1 Expansion of the Service Process Model through the Inclusion of Fuzzy Data

Figure 3 shows the fuzzy expansion of the service process model “Hot water is not hot”. The process is depicted in the form of a fuzzy event-driven process chain [TALL06; ThDL07]. The fuzzy constructs of the EPC are identified using grey shading. Initially, the designators for model elements used in the fuzzy EPC model “Hot water is not hot” are abstracted from the concrete functions of specific technical installations, their construction and the customer service activities necessary for their servicing and maintenance. To this extent, the fuzzy process model displays a higher level of abstraction than the “sharp” EPC models shown previously. The interdependencies between functions, components and the activities required on these components, as derived in Section 2.1 as the basis of information for service processes, need not be taken into consideration for the purpose of reaching a decision until the actual run-time of the process.

Depending upon the respective function tests, specific tasks must be carried out on the component parts by the customer service operatives – that is to say, the component parts must be maintained, repaired or replaced or the appliance must be set up. The decision with respect to the type of task to be carried out is taken during the actual run-time with the help of the function test in question. In addition, the “termination criteria” set out in the sharp service process model, which relates to the general termination of the repair process or the establishment of contact with the manufacturer’s customer service team are also taken into consideration. If it is not possible to use the inputted data and/or the measured values automatically read in as a basis for the decision-making process, the operator for the decision within the process model plans a return to the function tests, in order to give the customer service technician the opportunity to input further information. In this manner, the sequence of checking functions and replace-and-check functions previously, in the sharp case study, determined while the process of the provision of services was first set up, is broken up and is not defined until the actual run-time of the process represented.

A fuzzy system is being used here to support of the decision-making process. The embedding of the fuzzy information required takes place with the assistance of units from the organization, data objects and services that display fuzzy attributes. In this spirit, these attributes should in each case be interpreted as linguistic variables. In Figure 3, this is factored in by means of the fuzzy data objects *Quantity of outflow* and *Temperature of outflow*. Further model elements such as, for example, the data object

Tool and the organizational unit *Customer service* are annotated as sharp speech constructs relating to the central checking function of the fuzzy EPC model.

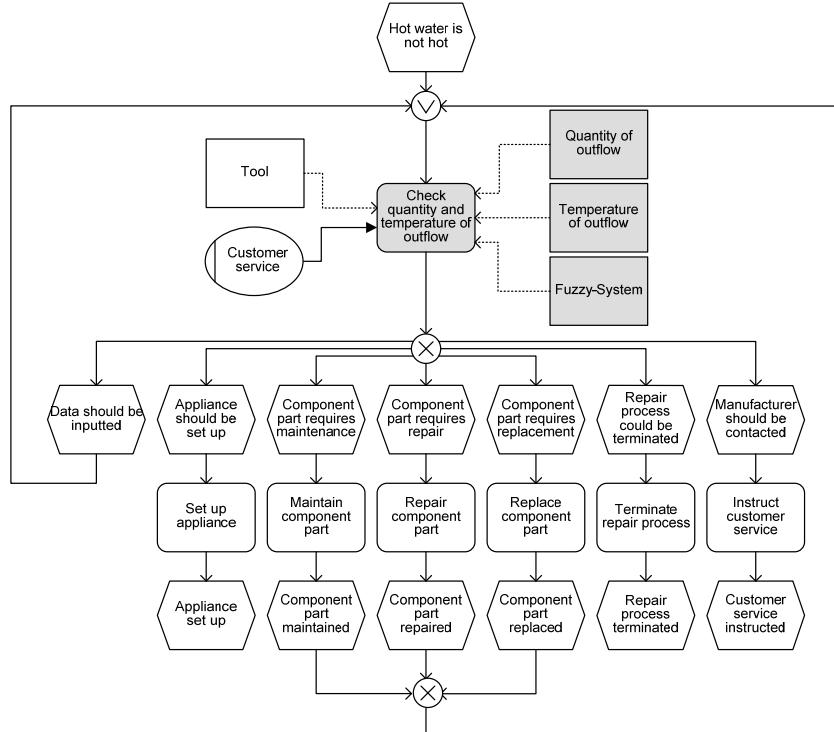


Figure 3: Fuzzy EPC model “Hot water is not hot”

3.2 How the Embedded Fuzzy System Works

The fundamental concept for using fuzzy systems support in the decision-making processes of the sample application under consideration orients itself on the measured values for the quantity and temperature of the outflow. The activities required by the customer service team in the further course of their service provision are dependent upon these values.

The incoming data values for the quantity and temperature of the outflow are implemented internally within the system according to their degree of membership. This fuzzification of the measured values is accompanied by the fuzzy inference process, which involves the use of language to describe, on the basis of the if-then rules determined in a rule basis, the value of the output quantity to be defined, which in the present instance entails giving priority to checking the water switch. Finally, this value is transformed into a sharp value through a process of defuzzification.

Initially, the constituent parts of the specification of the fuzzy system are the definition of the linguistic variables and the membership functions of the linguistic terms that describe in words the continuous input and output quantities.

In the present case for example, three terms, namely “cold”, “warm” and “hot”, are initially defined to describe the continuous quantity “Outflow temperature” by means of a linguistic variable (c.f. Figure 4, left).

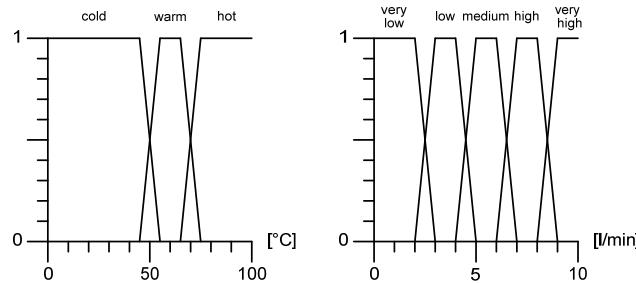


Figure 4: Membership functions of the linguistic variables “Outflow temperature” (left) and “Outflow quantity” (right) for the fuzzification of the input quantities

Depending on the current temperature value hot water appliance’s outflow, these terms describe the measured and/or imported temperature more or less correctly. In a similar fashion, the linguistic variable “Output quantity” is characterized using the terms “very low”, “low”, “medium”, “high” and “very high” and by the mapping these values on a graph (c.f. Figure 4, right). Here, the definitions of the membership functions for the input quantities are based on the combination wall-mounted water heater with flue attachment for heat and hot water on which this application is based. The ideal outflow quantity for it is 5.5 liters per minute and the ideal outflow temperature 60°C. In Figure 4, it can be recognized with respect to both of these linguistic variables that the membership functions are symmetrically constructed around these ideal values.

While the opening situation is described by means of the linguistic variables “outflow quantity” and “outflow temperature”, the level of priority to be accorded to when checking the water switch can be derived with the help of the established rules. In this respect the importance of carrying out this check is given a fuzzy evaluation with the help of a linguistic variable depending on its values. Figure 5 displays the construction of the membership functions of the linguistic variable “Priority to the water switch”. This priority should have a numerical value between 0 and 100 percent.

The behavior of the fuzzy system in the various situations entailed in the service process is determined by the rule basis. Here, each rule has an if-component (antecedent) and a then-component (consequence). In the if-component of a rule, the situation in which the rule is to be applied is described with the help of the input variables, following which the then-component is used to model the corresponding reaction. The evaluation of the rules is in turn influenced by the selection of operators. Here, the minimum operator frequently represents the word “And” and the maximum operator frequently represents the word “Or” [e.g. ZiZy80].

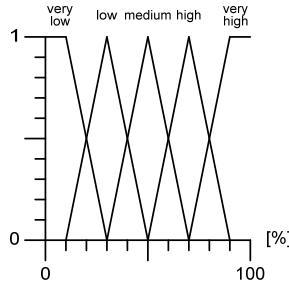


Figure 5: Membership functions of the linguistic variable “Priority accorded to the water switch” for the defuzzification of the output quantities

Table 1 displays the rule basis for determining the level of priority to be accorded to when checking the warm water switch on the basis of the input situations described by the variables “outflow quantity” and “outflow temperature”. If, for example, the available data indicates “cold” hot water and a “low” outflow quantity at the tapping point, then this should lead to a “high” level of priority for checking the water switch. On the other hand, a “warm” outflow temperature coupled with a “medium” outflow quantity should result in a “very low” evaluation of this priority. As a result of the overlapping of the membership functions, it is possible for multiple rules to come into play. The outcomes of the individual rules should, therefore, be drawn together into an overall statement at the conclusion. Then, finally, this overall statement should be used as the source for deriving the sharp values for the output quantity, which will in general be required. A range of methods exist for the defuzzification process and depending upon the circumstances of their application they will, even in the decision-making process concerning the repair procedure, represent the required behavior and the accompanying feasible outcomes.

Priority given to checking the water switch		Outflow quantity				
		very low	low	medium	high	very high
Outflow temperature	cold	high	high	medium	high	high
	warm	medium	low	very low	low	medium
	hot	high	high	medium	high	high

Table 1: Rule basis for determining the priority to be accorded to checking of the water switch

The procedure for determining the priority values can also be carried over to other groups of components and/or types of faults in this application. In order to produce a secure fault diagnosis, these values should be consulted by means of the mobile application system to be implemented. One sample calculation of the priority value of the output quantity at 0.42 (for reasons of space, the requirement for a demonstration of the calculation with the help of the fuzzy system previously specified is waived) does not constitute a definite indication that the water switch is the cause of the fault “Hot water does not heat up” and the element should therefore be replaced and checked directly following the evaluation of the measurements for the outflow quantity and temperature.

Rather, the intention in the present case is that further data should be measured or inputted into the mobile application system by the customer service technician.

4 Discussion

Based on problems posed in service process modeling, the deployment of fuzzy systems described above is justified in three respects. *First*, because natural language is used for modeling the framework of rules and is, at the same time, linked to numerical values by means of the definition of terms. Alongside the automatic processability of this service-related knowledge, complex cause-effect relationships can be documented intuitively and comprehended by the party providing the services. To this extent, the service technician can “understand” how an application system based on fuzzy rules derives its recommendations. Therefore, a system of this type is not seen exclusively as a “black box” and its acceptance among users can be increased significantly. *Second*, one characteristic of the fuzzy system constructed in this application is that the sequence in which the rules are applied doesn’t need to be specified and the database itself does not have to be complete, in order to make a reliable decision. This proceeds in accordance with the requirement formulated in Section 3.1, that the depictions of work processes in a process model should not be provided in an inflexible form. Thus, by and by, measured and/or estimated values can flow into the result of the evaluation in accordance with the evaluation of the rules, without having to run through a rigidly prescribed decision tree in sequence. Once sufficient data is available for a secure decision-making process then, with the help of the fuzzy controller, a decision can be made with respect to subsequent procedures during the service process. *Third*, the re-use of frameworks of rules and data already collected during the run-time of the service processes is a further key benefit in using such a fuzzified procedure as described above in modeling service process models. When for example, making a minor adjustment to the service process model only, such as an adjustment based on new findings with regard to a fault in a component – then the old rules need only be modified to a minor extent or it may even be possible to use them directly, in part at least. This means that the rules constitute a key knowledge component that can be re-used by the technical customer service team within individual service models across the board with respect to different types of faults and for different groups of components and even at different locations. In addition, the measured values that have already been read in for the outflow quantity and temperature can be taken into consideration directly as input data in subsequent functions of the service process, without having to be re-collected.

It could, for example, happen that, in the course of the service process, the gate valve at the front of the heating appliance is at a position that prevents the appliance from producing hot water in the correct manner. If the position of the gate valve needs to be corrected, then the outflow quantity and the outflow temperature will already have been collected at the tapping point and can be taken into consideration further evaluating the data. Here, the “rules of thumb” for the determining the perfect position of the gate valve can be collected in a rule-based manner, thus also making an evaluation of data by means of a fuzzy system possible.

5 Conclusion

In this article, the concept of a process-oriented service tool for supporting technical customer service process tools in the machinery and plant construction sector has been presented and then expanded by taking fuzzy data into consideration. The tool provides support for customer service technicians in deciding how to proceed with the service process. This decision is reached with the help of the fuzzy system that has been put in place.

Using the fuzzy EPC, findings formulated in natural language can be reproduced within the model in mathematical terms and then be re-used automatically. In terms of the design of the model, it must be pointed out that the adaptation of a process can be confined to only adapt the technical knowledge stored in the rules governing the decision-making process. Since the rule-basis is based on if-then rules, the functional behavior of the systems is comparatively easy to comprehend, system decisions are rendered transparent and it becomes possible to integrate available knowledge relatively simply. In the event of modifications in the process being controlled, old rules can be directly carried over or require minor modifications only. The continual improvement of the process definitions is thus made easier.

The creation of service process models for providing support to technical customer service teams clearly involves an extremely high outlay, which may well exceed any potential savings later achieved as a result of repair decisions that can be executed automatically. In order to make the expansion illustrated here available to the user – that is, to the designer or user of the model – it is necessary for fuzzy aspects to be integrated into the modeling tools. At the same time, in addition to the use of the manual explication techniques presented above, the application of automatic procedures on the basis of historical process instance data also needs to be considered. To this end, in [AdTL06] a proposal based on artificial neural networks was presented that, as an approach from the context of soft computing, directly supports the technology developed here.

References

- [AdTL06] Adam, O.; Thomas, O.; Loos, P.: Soft Business Process Intelligence – Verbesserung von Geschäftsprozessen mit Neuro-Fuzzy-Methoden. In: Lehner, F.; Nösekabel, H.; Kleinschmidt, P. (eds.): *Multikonferenz Wirtschaftsinformatik 2006 : Band 2*. Berlin : GIT, 2006, pp. 57–69
- [BoKL06] Bolumole, Y. A.; Knemeyer, A. M.; Lambert, D. M.: *The customer service management process*. Sarasota, Fla. Supply Chain Management Institute, 2006
- [Boni97] Bonissone, P. P.: Soft computing: the convergence of emerging reasoning technologies. In: *Soft computing* 1 (1997), no. 1, pp. 6–18
- [Czep80] Czepiel, J. A.: *Managing customer satisfaction in consumer service businesses*. Cambridge, Mass. Marketing Science Institute, 1980 (Report / Marketing Science Institute ; 80,109)
- [Harr07] Harris, E. K.: *Customer service : a practical approach*. 4th ed. ed. Upper Saddle River, N.J : Pearson Prentice Hall, 2007

- [KeNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage “Ereignisgesteuerter Prozeßketten (EPK)”. In: Scheer, A.-W. (ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, no. 89, Saarbrücken : Universität des Saarlandes, 1992
- [LaLo76] LaLonde, B. J.: Customer service: meaning and measurement. Chicago, Ill. National Council of Physical Distribution Management, 1976
- [Peel87] Peel, M.: Customer service : how to achieve total customer satisfaction. London : Kogan Page, 1987
- [SaBo03] Sawy, O. A. E.; Bowles, G.: Information technology and customer service. Oxford : Butterworth-Heinemann, 2003
- [StLa89] Sterling, J. U.; Lambert, D. M.: Customer service research : past, present and future. In: International journal of physical distribution & materials management 19 (1989), no. 2, pp. 2–23
- [TALL06] Thomas, O.; Adam, O.; Leyking, K.; Loos, P.: A Fuzzy Paradigm Approach for Business Process Intelligence. In: IEEE Joint Conference on E-Commerce Technology (CEC 2006) and Enterprise Computing, E-Commerce and E-Services (EEE 2006) : June 26–29, 2006, San Francisco, California. Los Alamitos, CA : IEEE Computer Society Press, 2006, pp. 206–213
- [ThDL07] Thomas, O.; Dollmann, T.; Loos, P.: Towards Enhanced Business Process Models Based on Fuzzy Attributes and Rules. In: Proceedings of the 13th Americas Conference on Information Systems : August 09–12, Keystone, Colorado, USA. Atlanta, Georgia, USA : AIS, 2007
- [Timm05] Timm, P. R.: Technology and customer service: profitable relationship building ; loyalty, satisfaction, organizational success. Upper Saddle River, NJ : Pearson Prentice Hall, 2005
- [TWLN07] Thomas, O.; Walter, P.; Loos, P.; Nüttgens, M.; Schlicker, M.: Mobile Technologies for Efficient Service Processes: A Case Study in the German Machine and Plant Construction Industry. In: Proceedings of the 13th Americas Conference on Information Systems : August 09–12, Keystone, Colorado, USA. Atlanta, Georgia, USA : AIS, 2007
- [VDMA08] VDMA (ed.): Maschinenbau in Zahl und Bild 2008. Mühlheim am Main : reuffurth, 2008 (Volkswirtschaft und Statistik)
- [WR07] World Resources InstituteWorld Resources Institute (ed.): Economics, Business, and the Environment – GDP: Percent GDP from services. URL http://earthtrends.wri.org/searchable_db/index.php?step=countries&ccID%5B%5D=0&theme=5&variable_ID=216&action=select_years [20.03.2007]
- [Zade94] Zadeh, L. A.: Fuzzy logic, neural networks, and soft computing. In: Communications of the ACM 37 (1994), no. 3, pp. 77–84
- [Zade97] Zadeh, L. A.: What is soft computing? In: Soft computing 1 (1997), no. 1, pp. 1
- [ZiZy80] Zimmermann, H.-J.; Zysno, P.: Latent Connectives in Human Decision Making. In: Fuzzy Sets and Systems 4 (1980), pp. 37–51

Geschäftsprozessmanagement mit EPK

EPK-Validierung zur Modellierungszeit in der bflow* Toolbox

Volker Gruhn¹, Ralf Laue¹, Heiko Kern² und Stefan Kühne²

¹ Angewandte Telematik / e-Business, Universität Leipzig
Klostergasse 3, 04109 Leipzig
{gruhn, laue}@ebus.informatik.uni-leipzig.de

² Betriebliche Informationssysteme, Universität Leipzig
Johanniskasse 26, 04103 Leipzig
{kern, kuehne}@informatik.uni-leipzig.de

Abstract: Dieser Beitrag stellt den Prototyp eines EPK-Modellierungswerkzeugs vor, der Verfahren zur Suche in Graphen nutzt, um Fehler in EPK-Modellen zu identifizieren. Dieses Werkzeug hat gegenüber bekannten Ansätzen zwei Vorteile: Zum einen ist es nicht notwendig, den (oft sehr großen) Zustandsraum aller möglichen Abläufe in einem Modell zu berechnen. Zum Zweiten kann unser Ansatz auch auf noch nicht vollständig fertiggestellte Modelle angewendet werden. Der Modellierer wird sofort zur Modellierungszeit über mögliche Probleme sowie deren Ursachen informiert und erhält unmittelbare Vorschläge zur Beseitigung der Probleme.

1 Einleitung

Verfahren zur Überprüfung der Korrektheit von EPKs und anderen Geschäftsprozessmodellen sind seit langer Zeit Gegenstand der Forschung. Wynn et al. [WVvE] schreiben in einem kürzlich veröffentlichten Artikel, dass „die Verifizierung von Prozessen so weit ausgereift ist, dass sie in der Praxis eingesetzt werden kann.“ Dem ist zwar zuzustimmen, es ist jedoch auch festzustellen, dass gängige Validierungstechniken den Geschäftsprozessmodellierer noch nicht auf die bestmögliche Weise unterstützen. Der Grund für diese Aussage liegt darin, dass formale Validierungsmethoden in der Regel erst angewendet werden, nachdem ein Modell fertiggestellt wurde. So können verschiedene Ansätze, die eine EPK zum Zwecke der Validierung in ein Petrinetz übertragen, nicht auf noch nicht komplett fertiggestellte Modelle angewendet werden.

In diesem Beitrag stellen wir einen Validierungsansatz vor, der dem Modellierer bereits während der Modellierung Rückmeldungen zu möglichen Problemen im Modell gibt. Wir beschreiben eine prototypische Implementierung dieses Ansatzes im Open-Source-Modellierungswerkzeug *bflow* Toolbox*¹. Mit dieser können nicht nur typische Kontrollflussfehler (wie Deadlocks oder Lifelocks) gefunden werden, sondern auch Probleme, die

¹ <http://www.bflow.org/>

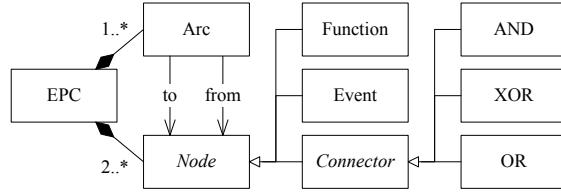


Abbildung 1: EPK-Metamodell

man unter der Bezeichnung „schlechter Modellierungsstil“ zusammenfassen könnte. Fehler werden im Modell durch direktes Markieren der betroffenen Modellelemente angezeigt. Dem Modellierer werden Möglichkeiten aufgezeigt, die gefundenen Fehler zu korrigieren. Mit Hilfe einer erweiterbaren Regelsprache können eigene Validierungsregeln (etwa für organisationsweite Stilregeln) hinzugefügt werden.

In Anlehnung an die aus der modernen Softwareentwicklung bekannten Begriffe Continuous Compilation und Continuous Testing bezeichnen wir unseren Ansatz als *Continuous Validation*. Damit soll hervorgehoben werden, dass eine Überprüfung des Modells ständig während des Modellierungsprozesses im Hintergrund abläuft und der Modellierer eine unmittelbare Rückmeldung über gefundene Probleme erhält.

2 Begriffe und Definitionen

2.1 Ereignisgesteuerte Prozessketten

Wir verwenden in diesem Beitrag ereignisgesteuerte Prozessketten mit den üblichen in der Literatur eingeführten Syntaxregeln [NR02]. Mögliche syntaktische Variationsmöglichkeiten (wie das Erlauben von aufeinanderfolgenden Funktionen ohne ein Ereignis dazwischen) sind ohne Einfluss auf die Grundideen unseres Ansatzes. Dank der Möglichkeit, die Regelbasis selbst anzupassen, können sie je nach Bedarf berücksichtigt werden.

Wir verwenden zur Beschreibung von EPKs das in Abb. 1 gezeigte Metamodell. EPKs sind nach diesem Metamodell endliche gerichtete zusammenhängende Graphen, bestehend aus Knoten (*Node*) und Kanten (*Arc*). Knoten unterteilen sich in Funktionen (*Function*), Ereignisse (*Event*) und die verschiedenen Arten von Konnektoren (*Connector*).

Die Begriffe *Zustand* einer EPK (worunter wir eine Markierung von Kontrollflusspfeilen verstehen), *Start- und Endzustand*, *Folgezustand* und *Zustandsübergangsrelation* verwenden wir wie in [NR02, Kin04, Men07] eingeführt.

2.2 Kontrollflussfehler

In [van99], definiert van der Aalst den Begriff der *Soundness* für EPK-Modelle. Dieses Korrektheitskriterium für Geschäftsprozessmodelle erfordert drei Eigenschaften:

1. In jedem Zustand einer EPK, der von einem Startzustand aus erreicht werden kann, muss es möglich sein, einen Endzustand zu erreichen (also einen Zustand, für den es in der die jeweilige Semantik beschreibenden Zustandsübergangsrelation keine Folgezustände gibt). (*option to complete*)
2. Wenn ein Zustand keinen Folgezustand hat, dürfen in diesem Zustand nur Endereignisse markiert sein. (*proper completion*)
3. Es gibt in der EPK kein Modellelement, für die es keinen Ablauf der EPK von einem Start- zu einem Endzustand gibt, in dem dieses Element markiert wird. (*no needless elements*)

Ist die Soundness-Eigenschaft verletzt, bedeutet dies, dass die EPK fehlerhaft ist. Ein typisches Beispiel ist eine Deadlock-Situation, in der ein XOR-Split mehrere Kontrollflusspfade eröffnet, die später von einem AND-Join zusammengeführt werden.

3 Ansätze zur Validierung des EPK-Kontrollflusses

Das erste Problem, das bei der Validierung von EPK-Modellen zu überwinden ist, besteht darin, dass die Modellierungssprache EPK ursprünglich eingeführt wurde, ohne formell die Semantik eines Modells zu definieren [KNS92]. Daher besteht der erste Schritt gängiger Validierungsansätze darin, ein EPK-Modell in einen Formalismus mit wohldefinierter Semantik zu transformieren. Genaugenommen wird erst durch diese Transformation die Bedeutung des Modells definiert. Verschiedene Autoren zeigen, dass Petrinette ein geeigneter Formalismus sind, um die Semantik von EPK-Modellen geeignet zu beschreiben [van99, vvV05, Men07]².

Nachdem ein EPK-Modell in ein Petrinetz überführt wurde, steht das komplette Arsenal an Petrinetz-Validierungswerkzeugen zur Verfügung, um Eigenschaften des Modells zu überprüfen. Wie verschiedene Autoren zeigen [LSW97, vJVVv07, Wyn06, Men07], sind diese Werkzeuge geeignet, um Fehler in EPKs aufzudecken.

Petrinetz-basierte Ansätze weisen aber häufig zwei Nachteile auf: Zum einen ist das Ergebnis einer Validierung oft nur die Tatsache, dass ein Fehler (z. B. Deadlock) vorliegt, ohne die Modellelemente benennen zu können, die für diesen Fehler „verantwortlich“ sind.

Selbst in den Fällen, in denen das Validierungswerkzeug das im Petrinetz erkannte Problem wieder in die Problemdomäne der analysierten EPK „zurückübersetzt“, können Informationen zu Fehlern verloren gehen.

² Die Liste der zitierten Arbeiten ist bei Weitem nicht vollständig. Eine ausführliche Übersicht findet sich in [vJVVv07] und [Men07].

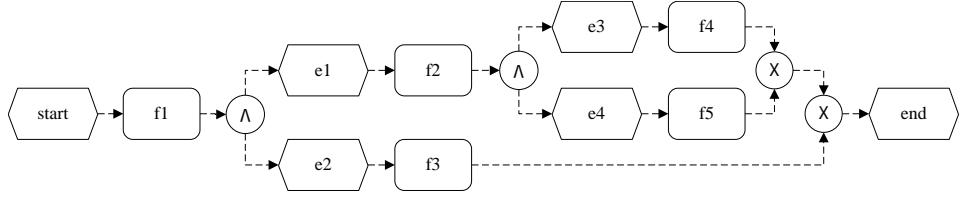


Abbildung 2: Zwei verschachtelte Kontrollflussblöcke mit Modellierungsfehler durch die Paarung AND-Split/XOR-Join

Dies soll in Abb. 2 illustriert werden. In diesem Modell ergeben sich aus einer unsachgemäßen Kombination eines AND-Splits mit einem XOR-Join offensichtlich zwei Fehler – einer im inneren AND-XOR-Kontrollblock und ein weiterer im äußeren AND-XOR-Kontrollblock.

Nehmen wir an (wie dies in der Literatur zur Semantik von EPK-Modellen in der Regel der Fall ist), dass ein XOR-Join „blockiert“, wenn an mehr als einem seiner Eingänge ein Kontrollfluss ankommt. Dann leitet im gezeigten Modell der innere XOR-Join auf Grund dieser Blockade nie einen Kontrollfluss an seinen ausgehenden Pfeil weiter. Daraus folgt aber, dass am äußeren XOR-Join (in der Abbildung rechts) immer nur am unteren eingehenden Pfeil ein Kontrollfluss ankommt. Ein Analysewerkzeug, das den gesamten Zustandsraum aller möglichen Abläufe im Modell untersucht, wird daher für den äußeren AND-Split/XOR-Join-Block nie einen Fehler feststellen.³

Ein zweiter Nachteil von Petrinetzbasierten Validierungsverfahren ist, dass sie häufig komplett modellierte EPKs als Eingabe voraussetzen. Eine während des Modellierungsprozesses im Entstehen begriffene EPK, die aus mehreren noch nicht miteinander verbundenen Teilen besteht, kann in diesem Falle nicht untersucht werden.

Eine weitere in der Literatur beschriebene Methode zur Suche nach Kontrollflussfehlern in EPKs und anderen Modellierungssprachen sind Reduktionsverfahren (erstmals beschrieben in [SO00]). Die Grundidee dieser Reduktionsverfahren besteht darin, dass in einem Modell sukzessive wohlstrukturierte Teile (wie beispielsweise eine Kombination aus öffnendem AND-Split und schließendem AND-Join ohne weitere Konnektoren dazwischen) entfernt werden. Es kann gezeigt werden, dass ein Entfernen dieser wohlstrukturierten Teile die Soundness-Eigenschaft des Modells nicht verändert [Wyn06]. Ist es also möglich, durch wiederholtes Anwenden von Reduktionsregeln das ursprüngliche Modell zu einem einzigen Knoten zu reduzieren, ist nachgewiesen, dass dieses Modell die Soundness-Eigenschaft erfüllt. Ist eine solche Reduktion jedoch nicht möglich, kann die Frage, ob das Modell sound ist, nicht entschieden werden. Somit liefern Reduktionsverfahren auf die Frage „Gibt es Kontrollfluss-Fehler im Modell?“ entweder die Antwort „Ja“ oder aber die Antwort „Das kann nicht entschieden werden“. Wünschenswert wäre jedoch eine Antwort, die entweder „Nein“ oder „Ja, und der erkannte Fehler lässt sich wie folgt beheben...“ heißt.

³ Etwas anderes gilt für Ansätze wie [van98], wo strukturelle Eigenschaften des Petrinetzes untersucht werden, die direkte Rückschlüsse auf Fehler in der EPK ermöglichen.

Einen Schritt in diese Richtung geht Mendling [Men07]. In dieser Arbeit werden nicht nur Reduktionsregeln für korrekte Modellteile betrachtet, sondern auch Modellelemente entfernt, die typische Fehlersituationen aufweisen. Diese Fehler werden dann gemeldet und eindeutige Hinweise auf die Modellelemente, die zum Fehler führen, gegeben. An Hand des SAP-Referenzmodells zeigte Mendling, dass sich auf diese Weise für die meisten EPKs entweder Soundness feststellen lässt oder sich Fehler lokalisieren lassen. In den Fällen, wo sich mit Hilfe von Reduktionsregeln kein Fehler im Modell finden lässt, jedoch das Modell auch nicht auf einen einzelnen Knoten reduziert werden konnte, verwendete Mendling die oben diskutierten Petrinetz-basierten Verfahren, um zumindest zu einer Ja/Nein-Entscheidung zur Soundness-Eigenschaft zu gelangen.

Die in der Arbeit von Mendling [Men07] beschriebenen Reduktionsregeln für typische Fehlersituationen betrachteten wir als einen Ausgangspunkt zum Aufstellen unserer Validierungsregeln.

Unser Beitrag wurde ebenso von dem in [vDMvdA06] veröffentlichten Ansatz beeinflusst. Dieser überträgt bekannte Begriffe und Eigenschaften von Petrinetzen (etwa den Begriff der *Handles* in einem Petrinetz [ES89, van98])) in die Anwendungsdomäne der EPKs. Zwischen EPK-Modellelementen werden Beziehungen der Art „nachdem Modellelement X durchlaufen wurde, muss immer ein Modellelement aus der Menge {Y, Z} durchlaufen werden“ betrachtet. Diese Beziehungen werden in [vDMvdA06] *causal footprints* genannt. Sie dienen als Grundlage zur Erkennung von Fehlermustern (z. B. Deadlocks oder Lifelocks). Mit Hilfe der causal footprints ist es möglich, direkt die Modellelemente zu identifizieren, die für einen Fehler verantwortlich sind. Ebenso lässt sich das Verfahren auch auf noch unvollständige Modelle anwenden. Allerdings ist der in [vDMvdA06] beschriebene Ansatz noch nicht für den Einsatz in der Praxis geeignet. Der erste Grund hierfür, dass keine OR-Konnektoren betrachtet werden, wiegt weniger schwer, da eine Erweiterung des Ansatzes von [vDMvdA06] um OR-Konnektoren ohne größere Schwierigkeiten möglich ist. Eine weit größere Einschränkung ist, dass die Berechnung aller causal footprints deutlich zu zeit- und speicherintensiv ist, da die Zahl der zu berücksichtigenden Beziehungen schnell sehr groß wird.

Ein weiterer heuristischer Ansatz ist in [VVL07] vorgestellt. Hier werden Workflow-Grafen in Single-Entry/Single-Exit-Regionen (also Teilmodelle, die nur einen eingehenden und einen ausgehenden Kontrollflusspfeil haben) unterteilt, was eine effiziente Untersuchung der Soundness-Eigenschaft für diese TeilmODELLE ermöglicht.

Dem in diesem Beitrag verwendeten Ansatz am ähnlichsten ist das von Awad und Puhmann [AP08] beschriebene Verfahren, das eine Abfragesprache namens BPMN-Q dazu nutzt, Fehlermuster in BPMN-Modellen zu finden. Es umfasst jedoch noch deutlich weniger Fehlermuster; insbesondere werden Modelle, die einen OR-Konnektor enthalten, nicht betrachtet.

4 Validierung zur Modellierungszeit

4.1 Validierungsansatz

In modernen Entwicklungsumgebungen für Programmiersprachen, wie beispielsweise der Eclipse-Umgebung, ist es heutzutage eine Standardfunktion, bereits während der Eingabe des Programms, die Syntax auf ihre Richtigkeit zu prüfen und entsprechende Fehler direkt anzuzeigen. Diese Form der Unterstützung wollen wir mit unserem Werkzeug auf die Geschäftsprozessmodellierung übertragen. Die Modellierung wird hierbei im Hintergrund ständig überprüft und dem Modellierer gegebenenfalls sofortige Rückmeldungen über Schwächen und Inkonsistenzen in möglicherweise auch unvollständigen Modellen angezeigt. Dadurch wird der typische Modellierungsprozess mit sequentieller Modellierung, Validierung und evolutionärer Weiterentwicklung soweit wie möglich verkürzt zu einem Modellierungsprozess mit integrierter Validierungsunterstützung. Darüber hinaus wird in dem hier vorgestellten Validierungsansatz nicht nur eine Syntaxüberprüfung vorgenommen, sondern es werden auch semantische und pragmatische Aspekte berücksichtigt.

Die zu entwickelte Validierungsunterstützung soll auf drei Prinzipien basieren.

1. Die Validierungsregeln sollen modular ausdrückbar und leicht lesbar sein, um Anpassbarkeit und Erweiterbarkeit zu gewährleisten. Hierfür eignen sich deklarative Validierungsregeln, weil diese leicht zu beschreiben sind und neue Regeln hinzugefügt werden können, ohne die bereits existierenden zu beeinflussen.
2. Der Validierungsansatz soll direkt auf den Eingabemodellen arbeiten, um möglichst den Nachteil von nicht-lokalierten Fehlermeldungen zu vermeiden.
3. Die Validierungslösung soll nahtlos in ein Modellierungswerkzeug integriert werden können. Fehlerursachen und mögliche Verbesserungsvorschläge sollen direkt im Modell angezeigt werden.

Die Umsetzung dieser drei Prinzipien im Kontext der Geschäftsprozessmodellierung resultiert in prozessspezifischen Validierungsregeln und dazugehörigen Hilfsfunktionen, die von diesen Regeln verwendet werden. Da sich ereignisgesteuerte Prozessketten und andere Geschäftsprozessmodellierungssprachen durch Graphen darstellen lassen, stellen diese Hilfsfunktionen Graph-Berechnungsfunktionen, wie „Gibt es einen Pfad zwischen zwei Knoten?“ oder „Geht jeder Pfad zwischen zwei Knoten durch ein anderen bestimmten Knoten?“ bereit. Weitere grundlegende Funktionen, die typischerweise in diesen Validierungsregeln verwendet werden, beziehen sich auf Modellelemente, wie „Ist dieses Element ein Startelement?“ oder „Ersetze ein Element durch ein anderes Element“, und mengenorientierte Operationen, wie „Selektiere alle Startereignisse einer EPK“ oder „Berechne die Schnittmenge aus allen Nachfolgern eines Elements und den Vorgängern eines anderen Elements“.

4.2 Implementierung

Der vorgestellte Validierungsansatz ist als eine Erweiterung des Open-Source-EPK-Mödellierungswerkzeugs bflow* Toolbox implementiert. Die bflow* Toolbox verwendet als Basistechnologien das Eclipse Modeling Framework (EMF)⁴ und das Graphical Modeling Framework (GMF)⁵. Für die Implementierung der angestrebten Validierungsunterstützung wird eine deklarative Validierungssprache und eine Modell-zu-Modell-Transformationssprache benötigt, um Anfragen auf Modelle auszudrücken und mögliche Fehler in Form einer Modell-zu-Modell-Transformation zu korrigieren. Im technischen Raum EMF stellen die Werkzeuge openArchitectureWare (oAW)⁶ und Epsilon⁷ solche Funktionalitäten zur Verfügung. Da oAW neben seinen Modellverarbeitungsmöglichkeiten auch eine Integration in GMF bietet, wurde oAW als Implementierungstechnologie gewählt.

Zur Beschreibung von Validierungsregeln wird die von oAW zur Verfügung gestellte Sprache *Check* verwendet. Ein einfache Check-Regel, die überprüft, ob ein Join gleichzeitig ein Split ist, ist in Listing 1 als Beispiel dargestellt. Eine Check-Regel beginnt mit der Definition eines Kontexts, der durch ein Element aus dem Metamodell (konkret einer EClass) festgelegt wird. Die Anwendung der Regel wird somit auf das Metamodellelement bzw. dessen Instanzen eingeschränkt. Diese Einschränkung kann durch eine optionale Bedingung in Form einer If-Anweisung verstärkt werden. Das darauf folgende Schlüsselwort ERROR signalisiert, dass es sich bei der verletzten Regel, um einen Fehler handelt. Eine weitere Fehlerkategorie, die eine Warnung an den Modellierer darstellt, wird durch WARNING eingeleitet. Nach diesen Schlüsselwörtern kann eine Nachricht spezifiziert werden, die dann in der Modellierungsumgebung eingeblendet wird. Nach dieser Nachricht muss hinter einem Doppelpunkt ein Ausdruck angegeben werden, der ein Boolean-Wert zurückliefern muss. Dieser Boolean-Ausdruck beschreibt die Zusicherung, die für valide Modelle gelten muss.

Listing 1: Beispiel einer Check-Regel

```
// Connectors are either splits or joins
context epc::Connector
  if this.isJoin()
    ERROR "Connector is a split and \
      a join as well. ..." :
    !this.isSplit();
```

Listing 2: Beispiel einer XTend-Funktion

```
// Is a connector a join-connector
Boolean isJoin(Element e) :
  epc::Connector.getInstance(e)
  && e.incomingArcs().size > 1;
```

Der Restriktions- und Zusicherungsausdruck in Listing 1 bezieht sich auf zwei separate in XTend definierte Funktionen. XTend ist eine weitere in oAW enthaltene funktionale Programmiersprache, um Modell-zu-Modell-Transformationen zu implementieren. Die Definition der Funktion *isJoin()*, die in Listing 1 verwendet wird, ist in Listing 2 gezeigt.

Die definierten Check-Regeln arbeiten auf EMF-Modellen. Um mögliche Fehlernachrichten, die durch die Verletzung bestimmter Regeln erzeugt wurden, im Editor anzuzeigen,

⁴ <http://www.eclipse.org/modeling/emf/>
⁵ <http://www.eclipse.org/modeling/gmf/>
⁶ <http://www.openarchitectureware.org/>
⁷ <http://www.eclipse.org/gmt/epsilon/>

wird der von oAW bereitgestellte GMF-Adapter verwendet. Dadurch ist es möglich, Modellelemente, die an der Verletzung einer Regel beteiligt sind, zu markieren und entsprechende Fehlernachrichten mit Verbesserungsvorschlägen anzuzeigen. Auf diese Weise erhält der Modellierer eine lokalisierte Rückmeldung.

Die verwendeten Technologien sind unabhängig von der EPK-Modellierung in der bflow* Toolbox und können auch zur Validierung von anderen Modelltypen eingesetzt werden. Lediglich die Validierungsregeln sind abhängig vom Modelltyp. Weiterhin sind die zugrunde liegenden Prinzipien des Validierungsansatzes nicht auf die beschriebenen Technologien beschränkt. Sie können auch in anderen EPK-Modellierungswerkzeugen, wie der ARIS Design Platform von IDS Scheer⁸ mit ihrer Makrosprache angewendet werden.

5 Typische Modellierungsfehler

5.1 Syntax-Fehler

Sehr typische Fehler in EPKs sind syntaktische Fehler. Die Überprüfung auf syntaktische Korrektheit bereitet wenig Schwierigkeiten. Da einige syntaktische Einschränkungen, wie das Verbot von Zyklen zwischen Konnektoren, nicht im EPK-EMF-Metamodell festgelegt werden können, erfordert dies zusätzliche Regeln [NR02, GL06]. Mit dem vorgestellten Ansatz lässt sich dies durch die Definition einfacher Regeln leicht realisieren.

Abb. 3(a) zeigt einen Syntax-Fehler, bei dem ein Konnektor gleichzeitig ein Join und Split ist. Dieser Fehler wird durch die Check-Regel in Listing 1 erkannt. Ein weiterer Syntax-Fehler ist, dass ein Ereignis oder eine Funktion mehr als eine ein- oder ausgehende Kante hat (siehe Abb. 3(b)). Solche Fehler sind nicht ungewöhnlich: Sie traten beispielsweise in 14 von 604 Modellen des SAP-Referenzmodells auf.

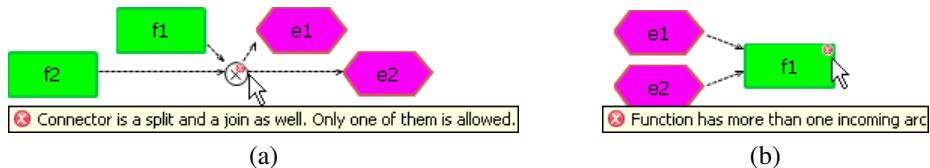


Abbildung 3: Konnektor mit mehr als einem eingehenden und mehr als einem ausgehenden Pfeil (a), Funktion mit mehr als einem eingehenden Pfeil (b)

5.2 Konfigurerende Konnektoren

Deadlocks und Synchronisierungs-Fehler in einer EPK resultieren daraus, dass der Typ eines Split-Konnektors nicht dem Typ des zugehörigen Join-Konnektors entspricht. Wird

⁸ <http://www.ids-scheer.com>

beispielsweise der Kontrollfluss durch ein XOR-Split in zwei alternative Pfade aufgeteilt und später durch ein AND-Join wieder zusammengeführt, führt dies zu einem Deadlock, da der AND-Join auf die Fertigstellung beider Kontrollflusspfade wartet.

Zu beachten ist, dass der oben verwendete Begriff des „zugehörigen Join-Konnektors“ für nicht wohlstrukturierte Modelle zunächst einer Definition bedarf. Zu diesem Zwecke führen wir die Relation *match* wie folgt ein:

Ein Split *s* und ein Join *j* werden als zusammengehörig betrachtet (symbolisiert durch die Relation $\text{match}(s, j)$) genau dann, wenn es zwei gerichtete Pfade von *s* nach *j* gibt, deren einzige gemeinsame Elemente *s* und *j* sind.

Falls es keinen „Einsprung in“ oder „Aussprung aus“ dem mit einem XOR-Split *s* beginnenden und mit einem AND-Join *j* endenden Kontrollflussblock gibt, ist das Modell in jedem Falle fehlerhaft, und zwar unabhängig von weiteren auf dem Pfad von *s* nach *j* liegenden Modellementen. Dabei definieren wir die Begriffe Einsprung und Aussprung wie folgt:

Sei *s* ein Split und *j* ein Join, für den $\text{match}(s, j)$ zutrifft. Der Kontrollflussblock zwischen *s* und *j* weist keine Ein- und Aussprünge auf (*seseMatch*(*s, j*) für *Single-Entry-Single-Exit*), gdw. die folgenden Bedingungen zutreffen:

1. Jeder Pfad von *s* zu einem Endereignis enthält *j*.
2. Jeder Pfad von einem Startereignis zu *j* enthält *s*.
3. Jeder Pfad von *s* zu *s* enthält *j*.
4. Jeder Pfad von *j* zu *j* enthält *s*.

Diese Definitionen ermöglichen das Finden von Kontrollflussfehlern, wo der Typ des Splits sich vom Typ des zugehörigen Joins unterscheidet. Von den in [Men07] gefunden 178 Fehlern fallen 44 in diese Kategorie. Abb. 4 zeigt einen solchen offensichtlichen Fehler, der aus dem Nicht-Übereinstimmen zwischen dem Typ des XOR-Splits (links) und dem Typ des AND-Joins resultiert. Die beiden XOR-Konnektoren im Inneren zeigen, dass die Regeln in Listing 3 auch Fehler in nicht wohlstrukturierten Modellen finden können.

Listing 3: Kontrollflussproblem: XOR-Split beginnt einen Block und AND-Join schließt ihn

```
// XOR-AND-Mismatch
context iepc::Connector if (this.isAndJoin())
    ERROR "Mismatched XOR-split ..."
    this.allPredessors().notExists(pl p.isXorSplit() && p.seseMatch(this));
```

Bei einer Kombination aus XOR-Split und OR-Join liegen die Dinge anders: Der OR-Join wartet auf alle eingehenden Kontrollflüsse. Wird am davorliegenden XOR-Split nur einer der ausgehenden Pfade aktiviert, heißt das, dass der OR-Join auf den Abschluss dieses einen Pfades wartet und die Ausführung dann problemlos fortsetzt. Da auf diese Weise kein tatsächlicher Fehler entsteht, wird diese Situation bei den meisten Validierungsansätzen (eine Ausnahme ist [Wyn06]) nicht beachtet. Unser Werkzeug gibt dem Modellierer den Hinweis, dass der OR-Join zu einem XOR-Join geändert werden sollte, was die Lesbarkeit des Modells erhöhen kann.

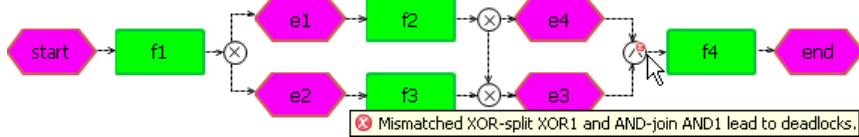


Abbildung 4: Synchronisationsproblem am AND-Join

Listing 4: Check-Regel für ein durch einen (X)OR-Split hervorgerufenes Synchronisationsproblem am AND-Join

```

context epc::Connector if this.isAndJoin() // An AND-Join J
ERROR "AND-split might not get control" :
// has no S with S is connected to J and
this.allPredessors().notExists(S)
// S is an XOR-split or an OR-split
(S.isXorSplit() || S.isOrSplit())
// and S has a successor SSucc not connected to J
&& S.successors().exists(SSucc)
SSucc != this && SSucc.isNotConnectedTo(this)
// and J has a predecessor JPred not connected from S
&& this.predecessors().exists(JPred)
JPred != S && S.isNotConnectedTo(JPred));

```

5.3 Synchronisation-Probleme in AND-Joins

Abb. 5 zeigt einen verbreiteten Fehlertyp, der in [Men07] am häufigsten gefunden wurde (102 von 178 entdeckten Fehlern). Wenn der obere am XOR-Split ausgehende Pfad durchlaufen wird, entsteht ein Deadlock am AND-Join.⁹

Listing 4 zeigt die entsprechende Check-Regel, um dieses Problem zu erkennen. Bei Ausführung der Regel wird eine Fehlermeldung erzeugt, die in Abb. 5 dargestellt ist.

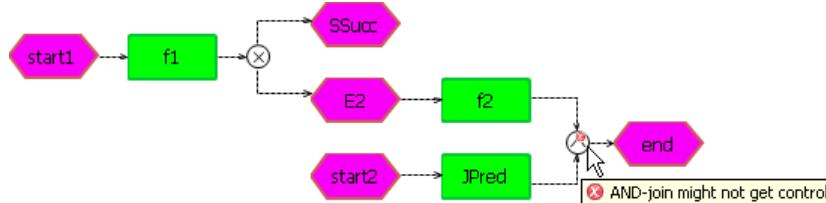


Abbildung 5: Synchronisationsproblem am AND-Join

⁹ In der Praxis würden wir nicht alle Modelle, die dieses Muster enthalten, als fehlerhaft betrachten, insbesondere wenn der in Listing 4 als JPred bezeichnete Knoten ein Startereignis ist. Diese Diskussion führt jedoch über die Zielstellung dieses Beitrags hinaus.

5.4 Unternehmensweite Stilregeln

Die Regelmenge in unserem Werkzeug ist leicht erweiterbar. Es ist beispielsweise denkbar, dass ein Unternehmen EPKs zu ausführbaren BPEL-Modellen über eine Modelltransformation überführen möchte, welche wohlstrukturierte Modelle verlangt. In diesem Fall können unstrukturierte EPKs durch unternehmensweite Stilregeln verboten werden.

Solche Anpassungen können in unserem Werkzeug leicht durch die Erweiterung der bestehenden Regelmenge realisiert werden. Ein Beispiel für ein solche mögliche Regel ist in Abb. 6 dargestellt. Hier soll nur nach Eintreten des Ereignisses e1 die Funktion f2 ausgeführt werden. Dies entspricht einer einfachen If-Anweisung ohne Else-Zweig. Da durch die direkte Kontrollflusskante zwischen den Konnektoren kein Ereignis modelliert ist, könnte dieser Pfad missverständlich auch so interpretiert werden, dass er in jedem Falle durchlaufen werden kann. Um diese Fehlinterpretation zu vermeiden, sollte auch in diesem Zweig ein Ereignis notiert werden. Mit einer einfachen Regel kann dieses Fehlermuster gefunden und das Einfügen der Negation von e1 im oberen Zweig empfohlen werden.

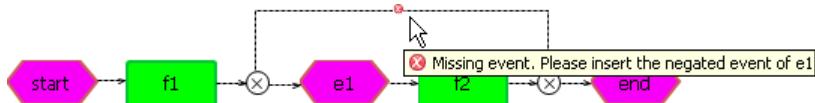


Abbildung 6: Fehlendes Ereignis zwischen XOR-Split und XOR-Join (mit Verbesserungsvorschlag)

6 Validierung

Material für eine erste Validierung unserer Korrektheitsregeln lieferte der Anhang von Mendlings Promotionschrift [Men07]. Er enthält das Ergebnis der Soundness-Überprüfung von 604 EPK-Modellen des SAP-Referenzmodells. Durch die Verwendung von Reduktionsregeln wurden (wie in Abschnitt 3 beschrieben) 178 Fehlermuster in 90 EPK-Modellen gefunden. Unser Werkzeug fand 176 dieser Fehlermuster (sowie zahlreiche weitere, die durch die Anwendung von Reduktionsregeln nicht entdeckt wurden). Die zwei von unserem Werkzeug nicht gefundenen Fehler beziehen sich auf Schleifen, in denen der Einstieg in die Schleife mit einem OR-Join statt eines XOR-Joins modelliert war. Nach der in [Men07] verwendeten Semantikdefinition sind diese Modelle nicht sound. Unserer Auffassung nach handelt es sich bei diesen Fällen jedoch eher um eine Eigenwilligkeit der Semantikdefinition in [Men07] als um einen tatsächlichen Fehler im Modell. Andere Semantiken (wie die in [Kin04] beschriebene Fixpunktsemantik) sehen solche Modelle nicht als fehlerhaft an.

In weiteren 57 Modellen im Anhang von [Men07] konnten die verwendeten Reduktionsregeln weder einen Fehler finden noch das Modell zu einem einzigen Modellelement reduzieren. Für diese Modelle prüfte Mendling die Soundness-Eigenschaft mit Hilfe eines Petrinetz-Analysewerkzeuges. Für 36 der Modelle wurde auf diese Weise gezeigt, dass sie

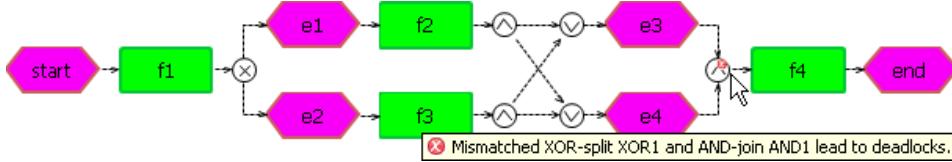


Abbildung 7: Diese EPK ist sound, unser Werkzeug meldet jedoch ein Problem.

nicht sound sind, und unser Werkzeug fand einen oder mehrere Fehler in *allen* diesen 36 Modellen. Die verbleibenden 21 EPKs erwiesen sich als sound. Unser Werkzeug lieferte nur für eine dieser EPKs eine Meldung über einen vermeintlichen Fehler.

Für jede der in [Men07] beschriebenen Fehlerklassen konnten wir eine Regel aufstellen und implementieren, die Fehler dieser Art findet. Ebenso konnten wir feststellen, dass alle im Beitrag von Koehler und Vanhatalo [KV07] beschriebenen Fehlermuster, gewonnen „aus hunderten reellen Prozessmodellen, die mit verschiedenen Werkzeugen erstellt wurden“ [KV07] durch unsere Regeln abgedeckt werden.

7 Zusammenfassung und weitere Forschungsziele

Die im vorigen Abschnitt durchgeführte Validierung zeigte, dass wir mit wenigen implementierten Regeln bereits in der Lage sind, nahezu alle Kontrollflussfehler in EPK-Modellen zu finden. Wir möchten aber an dieser Stelle die heuristische Natur unseres Ansatzes betonen. Die in den Editor integrierten Überprüfungen sollen vor allem ein schnelles Resultat liefern. Sie sollen nicht die in Abschnitt 3 beschriebenen exakten Validierungsverfahren ersetzen: Es kann nicht mit Sicherheit davon ausgegangen werden, dass ein Modell, für das unser Ansatz keine Fehler meldet, sound ist. Ebenso gelingt es, Beispiele zu konstruieren, in denen unsere Regeln fälschlicherweise einen Fehler melden. Ein Beispiel ist in Abb. 7 gezeigt. Obwohl das Modell sound ist, wird durch unsere Regeln ein Problem am AND-Join angezeigt. Ebenso wird nicht gemeldet, dass die OR-Joins durch XOR-Joins ersetzt werden sollten. Wir haben im Sinne einer schnellen Rechenzeit bewusst darauf verzichtet, solche eher exotischen Fälle bei der Formulierung unserer Regeln zu beachten.¹⁰

Eine Validierung unseres Werkzeug-Prototyps mit einer größeren Menge von EPK-Modellen aus dem SAP-Referenzmodell zeigte, dass bereits eine überschaubare Zahl von Validierungsregeln den überaus größten Teil von Fehlern in EPKs finden kann. Da unser Ansatz ohne Erzeugung des gesamten Zustandsraumes für mögliche Abläufe einer EPK auskommt, können diese Prüfungen sehr schnell erfolgen und während des Modellierens ständig im Hintergrund ablaufen. Die Tatsache, dass auf diese Art und Weise die Validierung zur Modellierungszeit stattfinden kann und somit der Modellierer noch vor Fertigstellung des Modells über mögliche Probleme und Verbesserungsmöglichkeiten in-

¹⁰ Im Übrigen sind wir der Auffassung, dass der Modellierer eines Modells in Abb. 7 durchaus gut darin beraten ist, sein Modell zu überdenken, da es schwer verständlich sein dürfte. Insofern kann eine Problemmeldung durch unser Werkzeug hier sogar gute Dienste leisten.

formiert wird, sehen wir als Hauptvorteil unseres Werkzeuges. Wir nahmen uns hierbei das UML-Werkzeug ARGO/UML [RR00] zum Vorbild, dass UML-Modellierer durch ständige Modellüberprüfungen im Hintergrund mit Hinweisen auf Verbesserungsmöglichkeiten unterstützt. Bei der Darstellung möglicher Probleme im Arbeitsbereich des Editors legten wir Wert darauf, dass das Signalisieren erkannter Fehlermuster so geschieht, dass die Probleme klar erkennbar dargestellt werden, ohne die Aufmerksamkeit des Modellierers von der Tätigkeit des Modellierens abzulenken [McF02].

Eine Richtung für unsere zukünftige Forschung wird es sein, sich auch mit Modellen auseinanderzusetzen, die zwar sound (und somit von rein technischem Standpunkt aus betrachtet korrekt) sind, für die es jedoch Modellierungsalternativen gibt, die einfacher verständlich sind [GL07].

Abschließend sei bemerkt, dass sich die Grundgedanken unseres Ansatzes auch auf andere Geschäftsprozessmodellierungssprachen wie BPMN übertragen lassen.

Literatur

- [AP08] Ahmed Awad und Frank Puhlmann. Structural Detection of Deadlocks in Business Process Models. In Witold Abramowicz und Dieter Fensel, Hrsg., *BIS*, Jgg. 7 of *Lecture Notes in Business Information Processing*, Seiten 239–250. Springer, 2008.
- [ES89] Javier Esparza und Manuel Silva. Circuits, handles, bridges and nets. In *Applications and Theory of Petri Nets*, Seiten 210–242, 1989.
- [GL06] Volker Gruhn und Ralf Laue. Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In Markus Nüttgens, Frank J. Rump und Jan Mendling, Hrsg., *EPK 2006, Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, 5. Workshop der Gesellschaft für Informatik e.V. (GI), Seiten 69–84, 2006.
- [GL07] Volker Gruhn und Ralf Laue. Good and Bad Excuses for Unstructured Business Process Models. In *Proceedings of 12th European Conference on Pattern Languages of Programs (EuroPLoP 2007)*, 2007.
- [Kin04] Ekkart Kindler. On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. In *Business Process Management*, Seiten 82–97, 2004.
- [KNS92] G. Keller, M. Nüttgens und A.W. Scheer. Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). *Veröffentlichungen des Instituts für Wirtschaftsinformatik*, (89), 1992.
- [KV07] Jana Koehler und Jussi Vanhatalo. Process anti-patterns: How to avoid the common traps of business process modeling, Part 1 - Modelling control flow. *IBM WebSphere Developer Technical Journal*, (10.4.), 2007.
- [LSW97] Peter Langner, Christoph Schneider und Joachim Wehler. Relating Event-driven Process Chains to Boolean Petri Nets. *Report*, (9707), Dezember 1997.
- [McF02] Daniel C. McFarlane. Comparison of Four Primary Methods for Coordinating the Interruption of People in Human-Computer Interaction. *Human-Computer Interaction*, 17:63–139, 2002.

- [Men07] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. Dissertation, Vienna University of Economics and Business Administration, 2007.
- [NR02] Markus Nüttgens und Frank J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen*, Seiten 64–77, 2002.
- [RR00] Jason E. Robbins und David F. Redmiles. Cognitive support, UML adherence, and XMI interchange in Argo/UML. *Information & Software Technology*, 42(2):79–89, 2000.
- [SO00] Wasim Sadiq und Maria E. Orlowska. Analyzing process models using graph reduction techniques. *Information Systems*, 25(2):117–134, June 2000.
- [van98] Wil M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
- [van99] Wil M.P. van der Aalst. Formalization and verification of event-driven process chains. *Information & Software Technology*, 41(10):639–650, 1999.
- [vDMvdA06] B.F. van Dongen, J. Mendling und W.M.P. van der Aalst. Structural Patterns for Soundness of Business Process Models. *EDOC*, Seiten 116–128, 2006.
- [vJVVv07] Boudewijn F. van Dongen, M.H. Jansen-Vullers, H. M. W. Verbeek und Wil M.P. van der Aalst. Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. *Comput. Ind.*, 58(6):578–601, 2007.
- [VVL07] Jussi Vanhatalo, Hagen Völzer und Frank Leymann. Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. *Service-Oriented Computing – ICSOC 2007*, Seiten 43–55, jan 2007.
- [vvV05] Boudewijn F. van Dongen, Wil M.P. van der Aalst und H. M. W. Verbeek. Verification of EPCs: Using Reduction Rules and Petri Nets. In *CAiSE*, Seiten 372–386, 2005.
- [WVvE] Moe Thandar Wynn, H. M. W. Verbeek, Wil M.P. van der Aalst und David Edmond. Business Process Verification - Finally a Reality! *Business Process Management Journal*, to appear.
- [Wyn06] Moe Thandar Wynn. *Semantics, Verification, and Implementation of Workflows with Cancellation Regions and OR-joins*. Dissertation, Queensland University of Technology, Brisbane, 2006.

Reducing Complexity of Large EPCs

Artem Polyvyanyy, Sergey Smirnov, and Mathias Weske

Business Process Technology Group

Hasso Plattner Institute at the University of Potsdam

D-14482 Potsdam, Germany

(artem.polyvyanyy,sergey.smirnov,mathias.weske)@hpi.uni-potsdam.de

Abstract: Business processes are an important instrument for understanding and improving how companies provide goods and services to customers. Therefore, many companies have documented their business processes well, often in the Event-driven Process Chains (EPC). Unfortunately, in many cases the resulting EPCs are rather complex, so that the overall process logic is hidden in low level process details. This paper proposes abstraction mechanisms for process models that aim to reduce their complexity, while keeping the overall process structure. We assume that functions are marked with efforts and splits are marked with probabilities. This information is used to separate important process parts from less important ones. Real world process models are used to validate the approach.

1 Introduction

Business process modeling plays an important role in the design of how companies provide services and products to their customers. To improve the understanding of processes and to enable their analysis, business processes are represented by models [Dav93, Wes07]. Business process models consist of automated and/or manual activities executed by an employee with a support of an information system. The goal of a process model is to provide a basis for defining and optimizing working procedures. Often achievement of this goal is traded for the cost of complex, “wallpaper-like” models, that tend to capture every small detail and exceptional case. Fine granular process models distract attention of a reader from the overall process logic by exhaustive details.

This paper proposes abstraction mechanisms that transform detailed process models in less detailed ones that still reflect the overall process logic. We do not assume any limitations on the initial process model control flow structure: proposed process model abstraction mechanisms implicitly define a set of addressed control flow patterns. The results are developed for EPC [KNS92, STA05]. However, they can be adapted to any graph-structured process modeling notation, for instance the Business Process Modeling Notation (BPMN) [BPM04].

The basic principle of the abstraction methodology proposed in this paper can be described as follows. Starting with a complex, detailed process model, a number of abstractions are performed. Formally, each abstraction takes a process model as input and generates a process model as output where an abstracted process fragment is replaced by a new one. The

new process fragment gives a generalized view of the substituted process fragment. Each individual abstraction leads to process details become concealed in a resulting process model.

The presented results were obtained in a joint research project with the health insurance company. Operational processes of the company are captured in about 4000 EPCs. Detailed models lead to information overload creating a demand for abstracted process models. The models are enriched with information about the effort required to complete each function of each process and probabilities of connection transitions from source to the target. The project partner uses proprietary tools to calculate the number of employees and their roles to enact all process instances that need to be executed. Since process models are the basis for head count estimations, an overall process effort after abstractions must remain unchanged.

This paper is structured as follows: Section 2 makes a survey on related work. Afterwards, the fundamental concepts are explained in Section 3. Elementary abstraction mechanisms are presented in Section 4. Concluding remarks complete this paper.

2 Related Work

The abstraction approach discussed in this paper bases on the set of elementary abstraction rules. Each rule specifies how a process model fragment can be transformed in order to simplify the process model. Graph transformation rules are well studied in literature [DJVVA07, LS03, MVD⁺08, SO00, VVL07]. These studies introduce graph reduction rules in order to facilitate analysis of process model soundness by means of state space reduction. An approach proposed in [SO00] presents rules facilitating soundness analysis of process models captured in the notation proposed by Workflow Management Coalition. The given set of rules can not analyze process models containing loops. [DJVVA07] and [MVD⁺08] specify reduction rules for structural analysis of EPCs. In [BRB07] the authors use graph reduction rules to create customized process views. Two kinds of rules are proposed: reduction rules and aggregation rules. It should be noticed that the named approaches do not define how such properties as process execution effort or execution cost can be preserved during transformations.

Cardoso et al. in [CMSA02] propose an approach for estimation of workflow properties (e.g., execution cost, execution time, and reliability) using the properties of activities constituting the process. The approach enables analysis of block-structured process models containing sequences, XOR blocks, AND blocks, and structured loops. However, the approach does not address processes which contain OR blocks and which are not block-structured.

A statistical approach to simplification of process models mined from execution logs is presented in [GA07]. It exploits various metrics for judging about the significance of process model elements and enables aggregation and reduction of insignificant elements. However, the approach does not address particularities of EPC and properties of a process, such as process effort, are not preserved.

The presented outlook of the related work witnesses: there is no comprehensive approach which solves the task discussed in this paper. Several approaches provide a solid basis of reduction rules, capable of handling sophisticated graph-structured processes. However, these approaches do not allow estimating process properties, such as effort or cost. On the other hand, there is an approach (cf. [CMSA02]) supporting process properties estimation, but it is limited to block-structured processes without OR blocks. Therefore, there is a lack of approach capable of handling graph-structured process models, i.e., providing appropriate graph transformation rules and rules for estimating process properties. In this study we target this challenge.

3 Fundamentals

This section introduces fundamentals of the approach—formalization of the extended for our purposes variant of event-driven process chains. There exist several works on formalization of EPC [Aal98, MA07, Wes07]. In this paper we use the formal definition proposed in [Wes07] and extend it by introducing concepts of function efforts and probabilities of connection transitions.

Definition 1 A tuple $(E, F, C, A, t, e_r, p_r)$ is an *extended EPC* if:

- E is a set of events, $E \neq \emptyset$
- F is a set of functions, $F \neq \emptyset$
- C is a set of connectors
- $N = E \cup F \cup C$ is a set of nodes, such that E , F , and C are pairwise disjoint
- $A \subseteq N \times N$ is a set of connections
- $t : C \rightarrow \{\text{and}, \text{or}, \text{xor}\}$ is a mapping assigning connector type to a connector
- $e_r : F \rightarrow \mathbb{R}^+$ is a mapping associating a function with an effort required to complete it (effort is measured in time units, e.g., minutes or hours)
- $p_r : A \rightarrow [0, 1]$ is a mapping assigning transition probability to a connection
- (N, A) is a connected graph
- Each function has exactly one incoming and one outgoing connection.
- There is at least one start event and at least one end event. Each start (end) event has exactly one outgoing (incoming) connection and no incoming (outgoing) connections. All the other events have exactly one incoming and one outgoing connections.
- Each event can only be followed (possibly via a connector) by a function and each function can only be followed (possibly via a connector) by an event.

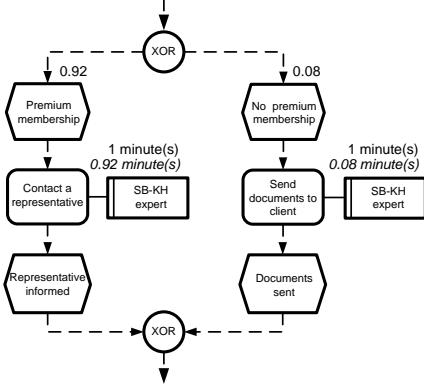


Figure 1: Real world example of the EPC fragment enriched with probabilities and efforts

- There is no cycle that consists of connectors only.
- No event is followed by an OR or a XOR split connector.

In order to address regions of an EPC we define an EPC process fragment as a connected subgraph of the (N, A) graph.

We assume that process models follow proposed formal EPC definition. However, this is not always true, e.g., in the investigated process models, events within a sequence of functions might be omitted. If this is the case, we assume a preprocessing step that modifies EPC to conform to proposed definition, i.e., missing events are automatically inserted.

To continue the discussion we need to define several auxiliary concepts.

Definition 2 *Mean occurrence number of a node* is the mean number that the node will occur in a process instance.

Definition 3 *Absolute effort of a process function* (e_a) is the mean effort contributed to the execution of the function in a process instance: $e_a : F \rightarrow \mathbb{R}^+$. Absolute effort can be obtained as the relative effort multiplied by the mean occurrence number of the function.

Definition 4 *Process absolute effort* (e_a^p) is the mean effort required to execute a process instance: $e_a^p : P \rightarrow \mathbb{R}^+$, where P is a set of process models. Process absolute effort can be obtained as the sum of absolute efforts of process functions.

Figure 1 shows the EPC fragment and illustrates presented concepts. Here, all the outgoing connections of the exclusive or split are supplied with the relative probabilities that sum up to one. All the other connections are assumed to have the relative probability of one. Each function is enriched with the relative and absolute (visualized in italic type) efforts given by the time interval in minutes that a worker needs to perform a function. For instance, the function “Contact a representative” has the relative effort of one minute meaning that

it is expected to take one minute of worker's time once reached in a process instance. On average, this function requires $1 \cdot 0.92 = 0.92$ minutes in every process instance which constitutes the absolute effort of the function. The absolute effort is obtained under the assumption that the process fragment is reached only once in a process instance with the probability of one. Semantically the effort concept is close to the concept of cost. For instance, if two activities are executed in parallel their total effort is the sum of efforts of both activities. In this study we do not address the waiting time between activities.

4 Elementary Abstractions

In this section elementary abstractions are presented. Elementary abstractions define how certain types of process fragments are generalized. The abstractions can be applied in any order or frequency, provided that a process model contains the structures required for a particular abstraction. This also assumes that any function can be the result of a prior abstraction.

4.1 Dead End Abstraction

Modeling of exceptional and alternative control flows in EPCs usually results in “spaghetti-like” process models with lots of control flow branches leading to multiple end events. As the primary goal of abstraction is to reduce excessive process details, it is of high importance to be capable of eliminating such flows, leaving only the essential information. To address this problem an elementary abstraction called *dead end abstraction* is introduced. Further discussion requires a precise definition of the term *dead end*.

Definition 5 An EPC process fragment is a *dead end* if it consists of a function, followed by a XOR split connector, followed by an event, followed by a function, followed by an end event. The XOR split connector has only one incoming connection.

Figure 2 illustrates the mechanism of the dead end abstraction. On the left side the initial process fragment containing a dead end is provided. Functions f_0 and f_k , events e_k and e_{k+1} and the XOR split connector constitute the dead end. The XOR split has k outgoing branches and after the abstraction the k -th branch is removed. On the right side of Figure 2 the abstracted process is presented.

As a result of abstraction, a XOR split branch which belongs to a dead end is completely removed from a process model. Function f_0 is replaced by an aggregating function f_D . An aggregating function in dead end abstraction has the following semantics: upon an occurrence of function f_D in a process, function f_0 is executed. Afterwards, function f_k may be executed. The probability that function f_k occurs is the probability of reaching function f_k from f_0 in the initial process. If function f_k is executed the branch is terminated and f_D is not left. Otherwise, the execution of the branch continues.

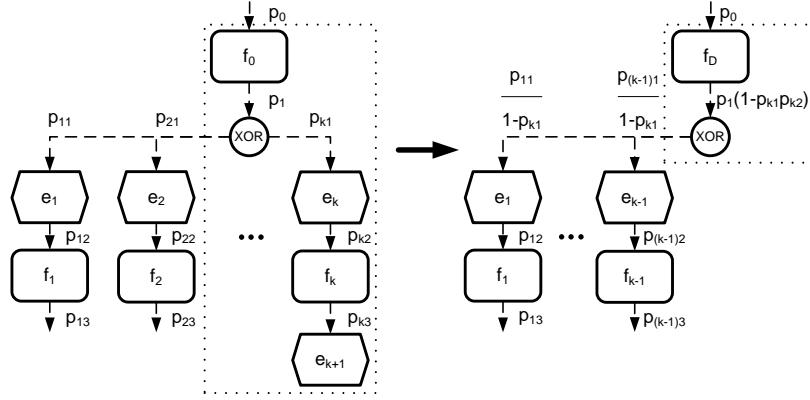


Figure 2: Dead end abstraction

The relative effort of an aggregating function takes into account the relative efforts of functions f_0 and f_k and the probability of f_k occurrence in f_D :

$$e_r(f_D) = e_r(f_0) + e_r(f_k) \cdot p_r((f_0, xor)) \cdot p_r((xor, e_k)) \cdot p_r((e_k, f_k)).$$

The relative probability of reaching a XOR split connector from function f_D is the probability of reaching the XOR connector from function f_0 and not reaching function f_k in the initial process:

$$p_r((f_D, xor)) = p_r((f_0, xor)) \cdot (1 - p_r((xor, e_k)) \cdot p_r((e_k, f_k))).$$

As a result of a dead end abstraction, the relative probability of entering the aggregating function is greater than the relative probability of leaving it: once function f_k is executed, the branch is terminated. Therefore, to find a probability of reaching one node from another, it is always required to take into account probabilities of all intermediate transitions.

Finally, we normalize the probabilities of the XOR split outgoing connections so that the following statements hold:

- the probabilities of reaching events e_i ($i = 1, 2, \dots, k-1$) from function f_D equal to the probabilities of reaching e_i from f_0 in the initial process
- the sum of the probabilities of the XOR outgoing connections is one.

The normalized relative probabilities are obtained in the following way:

$$p'_r((xor, e_i)) = \frac{p_r((xor, e_i))}{1 - p_r((xor, e_k))}.$$

If a XOR split has only two outgoing connections in the initial process it is possible to omit the XOR split after dead end abstraction.

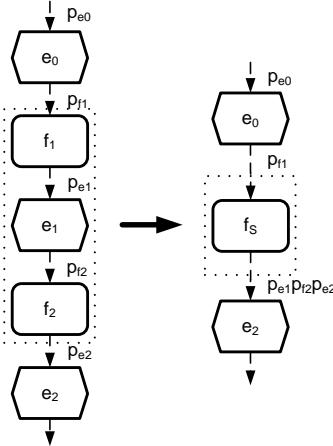


Figure 3: Sequential abstraction

4.2 Sequential Abstraction

Real world business process models of high fidelity often contain sequences of activities, which are captured in EPCs as sequences of functions. Within a *sequential abstraction* a sequence of functions and events is replaced by one function—an aggregating function. An aggregating function has a coarse granularity and brings a process model to a higher level of abstraction.

Definition 6 An EPC process fragment is a *sequence* if it is formed by a function, followed by an event, followed by a function.

Figure 3 exemplifies the concept of sequential abstraction. Functions f_1 , f_2 , and event e_1 form a sequence. As a result of sequential abstraction, a sequence is replaced by an aggregating function f_S . Semantics of the aggregating function is the following: function f_1 is executed and afterwards function f_2 occurs with the probability equal to the probability of reaching function f_2 from f_1 in the initial process.

The relative effort of an aggregating function depends on the relative efforts of functions f_1 and f_2 and the probability that f_2 occurs in f_S : $e_r(f_S) = e_r(f_1) + e_r(f_2) \cdot p_r((f_1, e_1)) \cdot p_r((e_1, f_2))$.

The relative probability of an aggregating function incoming connection is $p_r(e_0, f_1)$. The relative probability of an aggregating function outgoing connection is defined as $p_r((f_S, e_2)) = p_r(f_1, e_1) \cdot p_r(e_1, f_2) \cdot p_r(f_2, e_2)$

4.3 Block Abstraction

To model parallelism or to show that a decision is made in a process, a modeler encloses several branches of control flow between split and join connectors. Depending on the desired semantics, an appropriate connector type is selected: AND, OR, or XOR. A process fragment enclosed between connectors has a precise and self-contained business semantics. Therefore, the fragment can be replaced by one function of coarse granularity. *Block abstraction* enables this operation. To define block abstraction we use a notion of a path in EPC—a sequence of nodes such that for each node there exists a connection to the next node in the sequence.

Definition 7 An EPC process fragment is a *block* if:

- it starts with a split and ends with a join connector of the same type
- all paths from the split connector lead to the join connector
- there is at most one function on each path
- each path between the split and the join contains only events and functions
- the number of the outgoing connections of the split connector equals the number of the incoming connections of the join connector
- the split connector has one incoming connection and the join connector—one outgoing.

Figure 4 shows an example of a block. After block abstraction, an original process fragment is replaced by an event, followed by an aggregating function, followed by another event (events are added to assure that a new EPC is well-formed). The approach introduced in this paper supports AND, OR, and XOR connectors. Semantics of the aggregating function conforms to the semantics of the abstracted block and depends on the block type. In case of a XOR block the aggregating function (named f_B) means that only one function of the abstracted fragment is executed.

The relative effort of an aggregating function is independent of a block type and considers the relative efforts of functions f_i and probabilities of reaching these functions from a split connector: $e_r(f_B) = \sum_{i=1}^k e_r(f_i) \cdot p_r((c_1, e_{i1})) \cdot p_r((e_{i1}, f_i))$, where k is the number of split outgoing connections.

The relative probability of reaching event e_1 from f_0 equals to the relative probability of reaching node c_1 from its predecessor. The relative probabilities of connections (e_1, f_B) and (e_2, f_{k+1}) are one.

A method for p_x (cf. Figure 4) estimation is block type specific. Let us introduce probability p_i —the probability that a control flow reaches the join connector from the split connector on the i -th branch. Then the probability of reaching e_2 from f_B in an AND block is the probability that control flow on every branch reaches the join connector $p_r((f_B, e_2)) = \prod_{i=1}^k p_i$.

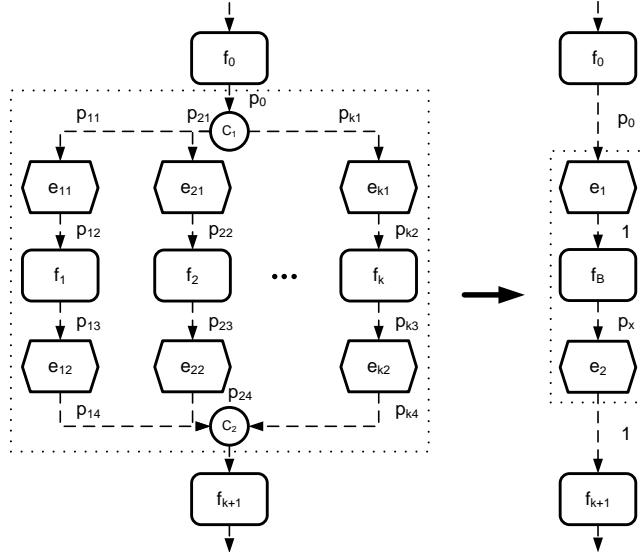


Figure 4: Block abstraction

For a XOR block this probability equals to the probability that the control flow on any branch reaches the join connector $p_r((f_B, e_2)) = \sum_{i=1}^k p_i$.

In case of an OR block where all probabilities of leaving functions are equal to one, probability of leaving the block is also one. In general case computation of the probability of leaving an OR block requires prior derivation of probabilities for selecting each branch combination of an OR block. This information can not be obtained solely based on probabilities of executing separate branches.

4.4 Loop Abstraction

It is a common situation when a task (or a set of tasks) in a business process is iterated to complete the process. In a model, capturing such a process, a task (or a set of tasks) is put into a loop construct. EPC enables loop modeling by means of control flow. Wide application of loops by modelers makes support of abstraction from loops an essential part of the approach. Therefore, we introduce one more elementary abstraction—*loop abstraction*. Let us define what kind of process fragment is considered to be a loop.

Definition 8 An EPC process fragment is a *loop* if:

- it starts with a XOR join connector and ends with a XOR split connector
- the process fragment does not contain any other connectors
- the XOR join has exactly one outgoing and two incoming connections

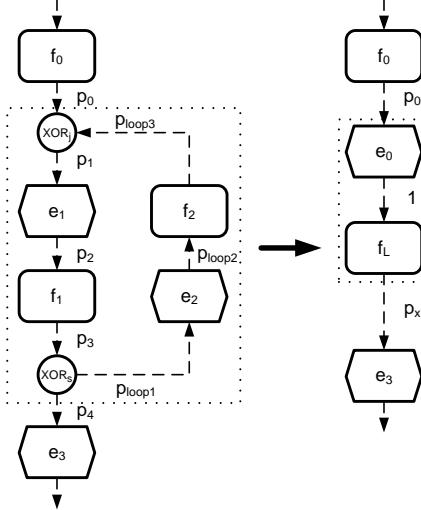


Figure 5: Loop abstraction

- the XOR split has exactly one incoming and two outgoing connections
- there is exactly one path from the split to the join and exactly one path from the join to the split
- there is at least one function in the process fragment.

The whole process fragment corresponding to a loop is replaced after the abstraction by one aggregating function f_L (see Figure 5). An extra event e_0 is inserted between the functions f_0 and f_L in order to obtain well-formed EPC. An aggregating function states that functions f_1 and f_2 are executed iteratively. The definition allows either f_1 or f_2 to be missing. Information about the number of loop iterations is hidden inside the aggregating function and is reflected in its relative effort and connections relative probabilities in the abstracted process model.

The relative effort of an aggregating function can be found as:

$$e_r(f_L) = p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot \frac{1}{1-p} \cdot (e_r(f_1) + e_r(f_2) \cdot p_r((e_1, xor_s)) \cdot p_l \cdot p_r((e_2, f_2))),$$

where $p = p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot p_r((e_1, xor_s)) \cdot p_r((e_2, f_2)) \cdot p_r((f_2, xor_j)) \cdot p_l$ and $p_l = p_r((xor_s, e_2))$.

After loop abstraction, the probability of reaching e_0 from f_0 equals the probability of reaching the XOR join from function f_0 in the initial process. The probability of reaching aggregating function f_L from event e_0 is one. Probability of leaving the aggregating function (denoted with p_x in Figure 5) is the probability of leaving a loop in the initial

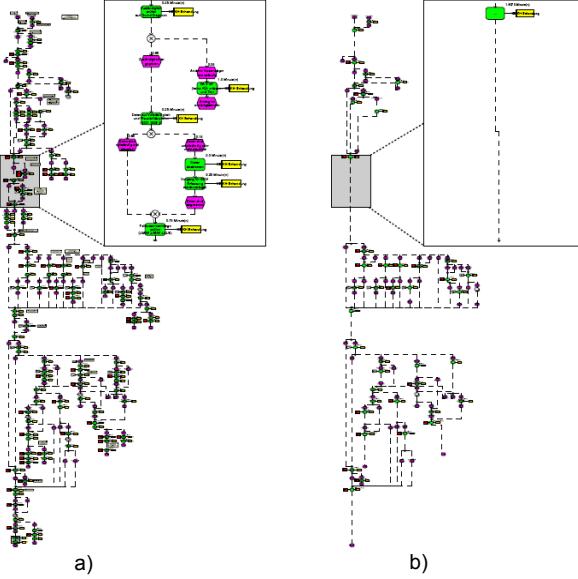


Figure 6: Original (a) and abstracted (b) process models (unreadability intended)

process. Since we assume that probabilities of leaving functions are not always one, it is possible that the control flow does not leave a loop in a process instance. The probability that a loop is stopped between xor_j and xor_s is $p_{stop}^{path} = 1 - p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot p_r((f_1, xor_s))$. The control flow stops on the path between xor_s and xor_j with probability $p_{stop}^{loop} = 1 - p_r((e_2, f_2)) \cdot p_r((f_2, xor_j))$.

Thus, the relative probability of leaving an aggregating function equals to:

$$p_r((f_L, e_3)) = 1 - \frac{1}{1-p} \cdot (p_{stop}^{path} + (1 - p_{stop}^{path}) \cdot p_l \cdot p_{stop}^{loop}).$$

4.5 Real World Example

In Figure 6 we present an example of a real world process model from our project partner (cf. Figure 6.a) and the result of its abstraction using presented elementary abstractions (cf. Figure 6.b). The initial process model is composed of 333 nodes: 130 functions, 137 events, and 66 connectors. After abstraction, the number of process model nodes was reduced to 167: 44 functions, 82 events, and 41 connectors. The overall reduction of process nodes is near 50%.

The proposed abstractions allow a company to deal with coarse grained functions in business processes, while keeping the overall process logic intact. In terms of organization and management, these coarse grained functions (with effort of minutes rather than of seconds)

facilitate process improvement on a higher level. Tedious discussions on low granularity functions are no longer required. Instead, process participants can apply improvements within the functions, keeping the overall process logic in sync with the process model.

5 Conclusions

In this paper the elementary abstractions: dead end, sequential, block, and loop abstraction were proposed. In the beginning we have described the challenges of our partner, which they came across managing their process models and which motivated this work. Proposed abstractions can be applied to an arbitrary graph-structured process model. Application of each elementary abstraction aggregates process fragment and brings model to a higher abstraction level. To the limitation of the approach one can count the fact that not an arbitrary model can be abstracted to one function, if such a behavior is desired. Also, proposed elementary abstractions only address preserving of process effort property (as in Definition 3), as it is the primary requirement of the project partner. However, the approach can be easily extended for recalculation of other properties as in [CMSA02].

Theoretical results of this work are used in the implementation of a tool prototype. The task of the tool is to provide automatic abstraction of process models captured in EPC. The tool supports all types of elementary abstractions proposed in this paper.

As the future steps we identify the task of developing additional elementary abstractions. This implies theoretical foundations of abstraction mechanisms as well as their prototypical implementation. Also, rules for composition of elementary abstractions (application order strategies) need to be studied. An important finding will be to show which class of EPCs can be abstracted to one function by a given set of elementary abstractions. It is also of great interest to learn which set of elementary abstractions is capable of reducing an EPC to one function.

References

- [Aal98] W. Aalst. Formalization and Verification of Event-driven Process Chains. Computing Science Reports, Eindhoven University of Technology, Eindhoven, 1998.
- [BPM04] BPMI.org. *Business Process Modeling Notation*, 1.0 edition, May 2004.
- [BRB07] R. Bobrik, M. Reichert, and Th. Bauer. View-Based Process Visualization. In *BPM*, volume 4714 of *LNCS*, pages 88–95. Springer, 2007.
- [CMSA02] J. Cardoso, J. Miller, A. Sheth, and J. Arnold. Modeling Quality of Service for Workflows and Web Service Processes. Technical report, University of Georgia, 2002. Web Services.
- [Dav93] T. Davenport. *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, Boston, MA, USA, 1993.

- [DJVVA07] B. Dongen, M. Jansen-Vullers, H. Verbeek, and W. Aalst. Verification of the SAP Reference Models Using EPC Reduction, State-space Analysis, and Invariants. *Comput. Ind.*, 58(6):578–601, 2007.
- [GA07] C. Günther and W. Aalst. Fuzzy Mining—Adaptive Process Simplification Based on Multi-perspective Metrics. In *BPM 2007*, volume 4714 of *LNCS*, pages 328–343, Berlin, 2007. Springer Verlag.
- [KNS92] G. Keller, M. Nüttgens, and A. Scheer. Semantische Prozessmodellierung auf der Grundlage “Ereignisgesteuerter Prozessketten (EPK)”. Technical Report Heft 89, Veröffentlichungen des Instituts für Wirtschaftsinformatik University of Saarland, 1992.
- [LS03] D. Liu and M. Shen. Workflow Modeling for Virtual Processes: an Order-preserving Process-view Approach. *Information Systems*, 28(6):505–532, 2003.
- [MA07] J. Mendling and W. Aalst. Formalization and Verification of EPCs with OR-Joins Based on State and Context. In *CAiSE*, pages 439–453, 2007.
- [MVD⁺08] J. Mendling, H. Verbeek, B. Dongen, W. Aalst, and G. Neumann. Detection and Prediction of Errors in EPCs of the SAP Reference Model. *Data Knowl. Eng.*, 64(1):312–329, 2008.
- [SO00] W. Sadiq and M. Orlowska. Analyzing Process Models Using Graph Reduction Techniques. *Information Systems*, 25(2):117–134, 2000.
- [STA05] A. Scheer, O. Thomas, and O. Adam. *Process Aware Information Systems: Bridging People and Software through Process Technology*, chapter Process Modeling Using Event-Driven Process Chains, pages 119–145. John Wiley & Sons, 2005.
- [VVL07] J. Vanhatalo, H. Völzer, and F. Leymann. Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In *ICSOC 2007*, volume 4749, pages 43–55. Springer, 2007.
- [Wes07] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer Verlag, 2007.

Ein Transformationsmodell zur Überführung von Prozessmodellen in eine Simulationsumgebung

Mathias Petsch, Hagen Schorcht, Volker Nissen, Katja Himmelreich

Technische Universität Ilmenau
Institut für Wirtschaftsinformatik
PF 100565
98684 Ilmenau

mathias.petsch|hagen.schorcht|volker.nissen@tu-ilmenau.de

Abstract: Viele Unternehmen nutzen Prozessmodelle, z.B. ereignisgesteuerte Prozessketten, im Rahmen ihrer Anstrengungen, Abläufe transparenter, effektiver und effizienter zu gestalten. Die Simulation von Geschäftsprozessen bietet die Möglichkeit, Auswirkungen von Änderungen im Geschäftsprozess am Simulationsmodell zu überprüfen. Als problematisch erweist sich in der Regel die Überführung von Prozess- in Simulationsmodelle. Aufgrund der unterschiedlichen Modellierungsziele und Detaillierungsgrade ist die direkte Übertragung häufig schwierig oder unangemessen. In diesem Beitrag wird basierend am Beispiel Krankenhaus ein Transformationsmodell als Zwischenstufe der Überführung von Prozess- in Simulationsmodelle vorgeschlagen und Überlegungen zur weiteren Entwicklung zu einem domänenunabhängigen Transformationsmodell angestellt.

1. Einleitung

Krankenhäuser stehen zunehmend unter dem Druck, einerseits eine Vielzahl von Leistungen anbieten zu müssen und andererseits dieses unter betriebswirtschaftlichen Aspekten (kostendeckend) zu gestalten. Galt es über viele Jahre insbesondere Leistungen durch neue Behandlungsmethoden und -geräte zu verbessern, werden nun unter dem Kostenaspekt zunehmend Prozessabläufe hinsichtlich möglicher Verbesserungen und Einsparungspotenziale untersucht. Dabei weisen Krankenhäuser diverse Eigenheiten auf, denen in der Prozessmodellierung und -optimierung Rechnung getragen werden muss:

1. Krankenhäuser bieten eine Vielzahl von Leistungen an, die wiederum in unterschiedlichen Prozessmodellen abgebildet werden müssen (so fallen zum Beispiel in einem Krankenhaus mit jährlich ca. 20.000 Patienten in einer Notfallaufnahme im Jahr bis zu 2.000 unterschiedliche Diagnosen an, die wiederum dementsprechend geeignete Behandlungspfade, bzw. Prozesse, bedingen);
2. Leistungen sind auf Grund des Fortschritts in Medizin und Technik (Untersuchungsmethoden, -geräte und Therapien) stark veränderlich;

3. die Prozesse sind in einem hoch-dynamischen Umfeld eingebettet und dadurch häufig selbst sehr flexibel (z.B. Reihenfolge bestimmter diagnostischer und therapeutischer Maßnahmen in Abhängigkeit von verfügbaren Ressourcen)

Trotz dieser Dynamik folgen doch in der Regel Handlungspfade bestimmten Diagnosen und weisen damit, bei allen Einschränkungen, eine quasi-Standardisierung auf [LK06]. Des Weiteren setzen sich die Behandlungspfade aus Teilprozessen zusammen, die ebenfalls gut standardisiert sind, wie zum Beispiel Röntgen oder Blutwertkontrolle. Das eröffnet die Möglichkeit, trotz erheblichen Aufwandes, zumindest die häufigsten auftretenden Behandlungspfade und Abläufe zu modellieren und gegebenenfalls zu verbessern. Um den Aufwand der Prozessmodellierung zu senken, werden heute Prozesse von Behandlungspfaden zusammengefasst, die auf unterschiedlichen jedoch ähnlichen Diagnosen beruhen. So wird für die Modellierung z.B. nicht zwischen den diversen Arten der Fraktur unterschieden, sondern auf Grund der starken Ähnlichkeit in der Behandlung und Diagnose ausschließlich ein Prozess „Fraktur“ modelliert. Auch stellt sich heraus, dass im vorliegenden Fall eines Thüringer Krankenhauses ca. 50 Diagnosen über 67% aller behandelten Fälle in der Notfallaufnahme ausmachen. Ein ähnliches Verhältnis ist im gesamten Krankenhausbereich zu beobachten. Das führt dazu, dass bei der Modellierung zum Zweck der Simulation bzw. der Prozessoptimierung auf die Mehrzahl der Behandlungspfade wegen ihres seltenen Auftretens verzichtet werden kann und nur die wichtigsten Diagnosen mit zugehörigen Behandlungspfaden modelliert werden müssen.

2. Modellierung und Simulation

Ein Schwerpunkt der Untersuchung von Abläufen im Krankenhaus liegt auf der Prozessgestaltung und Steuerung logistischer Abläufe bei der Erstellung von Leistungen am Patienten [Br99]. Dabei dient die Prozessmodellierung der Beschreibung des Verhaltens eines realen Systems und, daraus abgeleitet, der Möglichkeit Rationalisierungspotenzial zu identifizieren [BS04, 107]. Für die Patienten im konkreten Anwendungsfall sollten vor allem Wartezeiten verkürzt werden, da langes Warten in der Notaufnahme eine erhebliche Unzufriedenheit der Patienten und in der Folge eine negative Imagewirkung für das Krankenhaus zur Folge hatte. Daneben bestand im betroffenen Krankenhaus der Wunsch, Möglichkeiten der Kosteneinsparung durch die Verbesserung von Prozessabläufen zu realisieren. Aufgrund der vielschichtigen zeitlichen, inhaltlichen und kommunikativen Abhängigkeiten zwischen einzelnen Prozessen [Ha02] ist diese Aufgabe jedoch in Krankenhäusern generell äußerst komplex. Die Simulation stellt hierzu ein geeignetes Hilfsmittel dar. Ein erster Schritt zur Verbesserung bestehender Abläufe ist die Erhebung und Dokumentation der Geschäftsprozesse. Dabei stehen häufig andere Zielsetzungen als für die spätere Simulation im Vordergrund und folglich sind die für die Prozessmodellierung entwickelten Methodiken auch weitgehend unabhängig von dem Wunsch entstanden, Abläufe konkret zu simulieren. Daraus können methodische Probleme bei der Entwicklung eines Simulationsmodells erwachsen. So geht es auf der Prozessmodellierungsebene primär um Prozesstransparenz, kürzere Einarbeitungszeiten für neue Mitarbeiter oder die Vorbereitung einer Zertifizierung nach DIN ISO 9000 ff. Eine anerkannte Methode zur

Modellierung von Geschäftsprozessen in Unternehmen stellen die erweiterten Ereignisgesteuerten Prozessketten (eEPK) dar, die auch im vorliegenden Fall des Thüringer Krankenhauses im Rahmen der Ist-Aufnahme von Prozessen der Notfallambulanz eingesetzt wurden.

Die Nutzung vorhandener Modelle von Geschäftsprozessen in Unternehmen minimiert den Aufwand der Modellerstellung für eine Simulationen erheblich. Dabei ist jedoch ein wichtiger Unterschied zwischen Geschäftsprozess- und Simulationsmodell zu berücksichtigen: der aus dem Modellierungsziel folgende unterschiedliche Abstraktionsgrad [NRS03, 450]. Während mittels eEPK Unternehmensprozesse häufig sehr detailliert abgebildet werden, benötigt die Simulation einen höheren Abstraktionsgrad hinsichtlich der einzelnen Abläufe bzw. Funktionen, aber einen erheblich höheren Detailierungsgrad in Bezug auf quantitative Größen, wie Zeiten, Kosten, Verteilung, Wahrscheinlichkeiten oder die Anzahl von Objekten und Ressourcen aus der Realwelt.

Dies bedingt, dass einerseits nicht alle Informationen, die mit Hilfe der Prozessmodelle in Unternehmen erfasst und modelliert wurden relevant für ein Simulationsmodell sind und andererseits teilweise zusätzliche quantitative Größen erhoben werden müssen, die in der Regel für die Modellierung mittels eEPK nicht notwendig sind. Hier ergeben sich Schwierigkeiten bei der Überführung der Modelle. Das in diesem Papier vorgestellte Vorgehen mittels eines intermediären Transformationsmodells soll helfen, vorliegende oder leicht zu erhebende Informationen in Form von Prozessmodellen über einen Zwischenschritt leichter in Simulationsmodelle umzusetzen. In der Literatur lassen sich verschiedene, oft eher pragmatische Ansätze solcher Modelltransformationen finden [VB02] [MN04] [Re07]. Dabei wird häufig eine XML-Schnittstelle genutzt, um das ursprünglich Prozessmodell aus einem Modellierungswerkzeug wie ARISTM zu exportieren und anschließend in die Simulationsumgebung zu integrieren. Nachteil aller dieser Ansätze ist der Versuch, das Prozessmodell möglichst genau und damit unverändert in eine andere Modellform zu überführen. Diese Vorgehensweise ist jedoch, je nach vorliegendem Ursprungsmodell und damit verbundenen Zielsetzungen, oft nicht adäquat.

Im Folgenden wird ein Konzept vorgestellt, wie als eEPK vorliegende Prozessmodelle mit Hilfe einer grafisch-deskriptiven Modellierungssprache über die Zwischenstufe eines Transformationsmodells in ein Simulationsmodell überführt werden können.

3. Transformationsmodell

Grundsätzlich soll das Transformationsmodell zukünftig domänenunabhängig gestaltet werden, wurde aber bei der ersten Entwicklung für eine Problemstellung im Krankenhausbereich entworfen und hat demnach noch eine stark domänen spezifische Ausrichtung.

Die hier zugrunde liegende Krankenhausdomäne bedingt einige Besonderheiten, die in der Modellierung und Simulation beachtet werden müssen. So wurde u.a. im Simulationsmodell berücksichtigt, dass zu bestimmten Zeitpunkten ein Raumwechsel

des Patienten stattfindet und dass der Aufenthalt eines Patienten in einem bestimmten Raum wesentlich von der Art der Erkrankung (Patiententyp) abhängig ist. Weiterhin wurden, um den Gegebenheiten des verwendeten Simulationswerkzeuges Rechnung zu tragen, Raumgruppen definiert, die mindestens zwei Untersuchungsräume umfassen. Generell lassen sich Räumlichkeiten oder Ressourcen, wie z.B. Röntgengeräte etc., mit Hilfe der eEPK nur ungenügend abbilden. Diese sind aber in der Regel essentieller Bestandteil und Grundlage für ein Simulationsmodell.

Weiterhin ist eine feingranulare Abbildung von Prozessschritten in Simulationsmodellen nicht notwendig. Beinhaltet z.B. der Vorgang des Röntgen diverse Prozessschritte (z.B. Röntgenscheinprüfung, Patienten vorbereiten etc.) ist es im Simulationsmodell unter Umständen ausreichend, diese Schritte zu einem aggregierten Prozess zusammenzufügen, der mit der Gesamtzeit und den Kosten beschrieben ist. So werden folglich bei der Überführung der eEPK in das Simulationsmodell Aggregationen notwendig.

Bezeichnung	Kurzbeschreibung (kursiv: Kann-Bestandteil)	Grafische Notation	Kurzbeschreibung (kursiv: Kann-Bestandteil)
Titel-Symbol	Gibt den Namen des Patiententyps Dienstprozesses an		
QuellenSymbol	Bildet eine Quelle ab ZAZ (Zwischenankunftszeit) in Stunden		
Raum-Symbol	Bildet einen Raum eine Raumgruppe ab Bedienzeit in Minuten Zum Eintrittszeitpunkt gerufene Ressourcen Zum Austrittszeitpunkt freigebene Ressourcen		
Dienstprozess Symbol	Bildet einen Verweis auf einen Dienstprozess ab		
Patiententyp Symbol	Bildet einen Verweis auf einen Patiententyp ab		
Teilprozess Symbol	Bildet einen Verweis auf einen Teilprozess ab.		

Abbildung 1: Grafische Notation zum Transformationsmodell

Auch existiert z.B. bei der Behandlung von Patienten eine große Variationsbreite, die sich in vielfachen Verzweigungen von Behandlungspfaden manifestieren und so nur schwer in Simulationsmodellen abbilden lassen. Für die Zwecke der Simulation können hier Tätigkeiten aggregiert betrachtet und somit der Abstraktionsgrad erhöht werden.

Daneben können Eigenheiten des Simulationswerkzeuges zu Anpassungsbedarf führen. Weiterhin werden in einer Prozesssimulation im Wesentlichen nur zeitkritische Prozessschritte betrachtet, was wiederum dazu führt, dass nur diese im Simulationsmodell abgebildet werden müssen. Aus diesen Überlegungen resultiert die Idee, das ursprüngliche Prozessmodell erst über den Zwischenschritt eines Transformationsmodells in das spätere Simulationsmodell zu überführen. Dieses Vorgehen hat desweiteren den Vorteil, dass bei der Überführung einer eEPK in das Transformationsmodell der Bedarf nach einer zusätzlichen Quantifizierung von Parametern deutlich wird. Abbildung 1 zeigt die hier vorgeschlagene Symbolik einer grafischen Notation des Transformationsmodells, die nachfolgend näher erläutert wird.

Bezeichnung	Kurzbeschreibung (kursiv Kann-Bestandteil)	Grafische Notation	Kurzbeschreibung (kursiv Kann-Bestandteil)
Senken-Symbol	Bildet eine Senke ab.		<i>Wahrscheinlichkeit für die Durchführung des einzelnen Pfades</i> <i>Hilfsvariable zur Identifizierung der Verzweigungen für die spätere Simulation</i>
Raumgruppen-Symbol	Angabe einer Raumgruppe		
NotizSymbol	Notiz		
Variablen-Symbol	Schnittstelle (z.B. Dateinamen) zu externen Daten		
Verzweigungs-Symbol	Verzweigung		
Patientenfluss-Symbol	Kante, dient zur Darstellung des Patientenflusses		

Abbildung 2 (Fortsetzung): Grafische Notation zum Transformationsmodell

Die *Quelle* im Transformationsmodell entspricht dem Auslöseereignis einer eEPK und eine Senke dem Endeereignis. Im Unterschied zu einer eEPK¹ ist zu erkennen, dass in den einzelnen Elementen die für die Simulation notwendigen Wahrscheinlichkeiten und Zeiten (minimale bzw. maximale Durchlauf- respektive Behandlungszeit) hinterlegt sind. Patienten können in *Patiententypen* zusammengefasst modelliert werden. Darunter sind Patienten zu verstehen, die eine spezifische Diagnose aufweisen und dementsprechend einem gemeinsamen (aggregierten) Behandlungspfad folgen.

Weiterhin beinhaltet die Notation *Raumgruppen*, denen verschiedene *Räume* und auch Personen (Stellen) zugeordnet sind. Die Modellierung mittels Raumgruppen erfolgt bei alternativen Räumlichkeiten, wenn deren Wahl keinen Einfluss auf Ablauf oder Zeiten im Modell hat - z.B. wenn unterschiedliche Behandlungsräume existieren, die nur geringfügige Unterschiede aufweisen. Raumgruppen können in Abhängigkeit von der Anwendungsdomäne als Ressourcen (z.B. Geräte) uminterpretiert werden. Raumgruppen werden in Abhängigkeit zum konkreten Behandlungspfad modelliert, d.h. auf Räume die im Verlauf einer Behandlung nicht genutzt werden, wird in der Modellierung verzichtet.

Die Prozessmodellierung mittels eEPK erfolgt meist hierarchisch vom Grobprozess hinunter bis zur detaillierten Darstellung der einzelnen Prozessschritte. Dabei werden Teilprozesse, die in verschiedenen Prozessen benötigt werden, nur einmal modelliert und dann auf diese verwiesen. In der hier betrachteten Krankenhausdomäne wird beispielsweise der Teilprozess Röntgen bei unterschiedlichsten Erkrankungen und Diagnosen benötigt.

Im Transformationsmodell wird dieser Teilprozess als *Dienstprozess* bezeichnet. Demgegenüber versteht man unter *Teilprozessen* im Sinne des Transformationsmodells spezielle medizinische Behandlungspfade, die wiederum Bestandteil eines generellen Behandlungspfades eines Patienten sind und auf Grund ihrer Komplexität (Vielfalt an diagnostischen und/oder therapeutischen Mitteln) separat abgebildet werden.

Mit Hilfe des *Variablen-Symbols* wird die Schnittstelle zu externen Daten abgebildet. Im unten beschriebenen Beispiel wird dieses Symbol verwendet, um auf Dateien (Schnittstellen) verweisen zu können, die Daten über die Anzahl und Häufigkeit bestimmter Erkrankungen enthalten.

Beispiel:

Im Folgenden wird exemplarisch die Überführung einer eEPK in das Transformationsmodell dargestellt. In Abbildung 3 ist der Patiententyp Schnittverletzung in einer eEPK dargestellt und in Abbildung 4 der dazugehörige Ausschnitt des Transformationsmodell (jeweils durch Umrandung verdeutlicht). Nach der Anmeldung und Wartezeit betritt der Patient Untersuchungsraum 1 - in Abbildung 4 ein Teil der Raumgruppe 3, die weiterhin auch Behandlungsraum 2 und 3 umfasst. Im eEPK-Prozessmodell von Abbildung 3 sind

¹ Den Autoren ist bewusst, dass z.B. in ARIS die Möglichkeit besteht, quantitative Größen zur Beschreibung der einzelnen Elemente zu hinterlegen, die jedoch in der Regel in der grafischen Darstellung nicht ersichtlich sind.

weitere Räumlichkeiten aufgeführt (z.B. Reanimationsraum), die aber nicht dem Dienstprozess „Schnittverletzung“ zugeordnet sind und deswegen auch nicht Bestandteil des Transformationsmodells zu diesem Prozess sind.

Die beiden Zweige der eEPK „stark blutende Wunde“ und „nicht stark blutende Wunde“ lassen sich in jeweils einer Raumgruppe aggregieren. Dadurch wird Komplexität, die für die Simulation irrelevant ist, reduziert. Die in der eEPK abgebildeten XOR-Verzweigungen werden dagegen direkt in das Transformationsmodell überführt. Auch die in der eEPK ersichtlichen Personen (Chirurg, Pflegekraft) sind ebenso Bestandteil des Transformationsmodells und dort der Raumgruppe 3 zugeordnet. Weiterhin ist ersichtlich, dass die Wahrscheinlichkeiten der Belegung der Behandlungspfade (25:75 %) direkt aus der eEPK übernommen wurden. Der Dienstprozess Anmeldung ist dagegen in einem eigenständigen Transformationsmodell abgebildet.

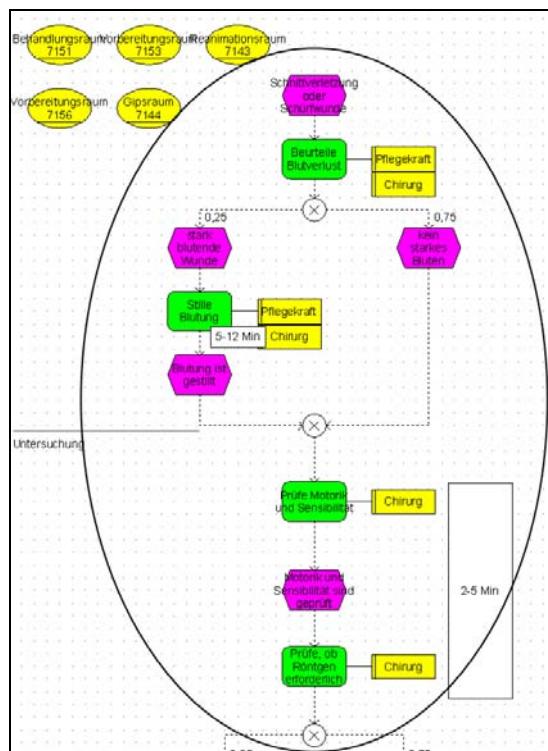


Abbildung 3: eEPK Patiententyp Schnittverletzung – Ausschnitt

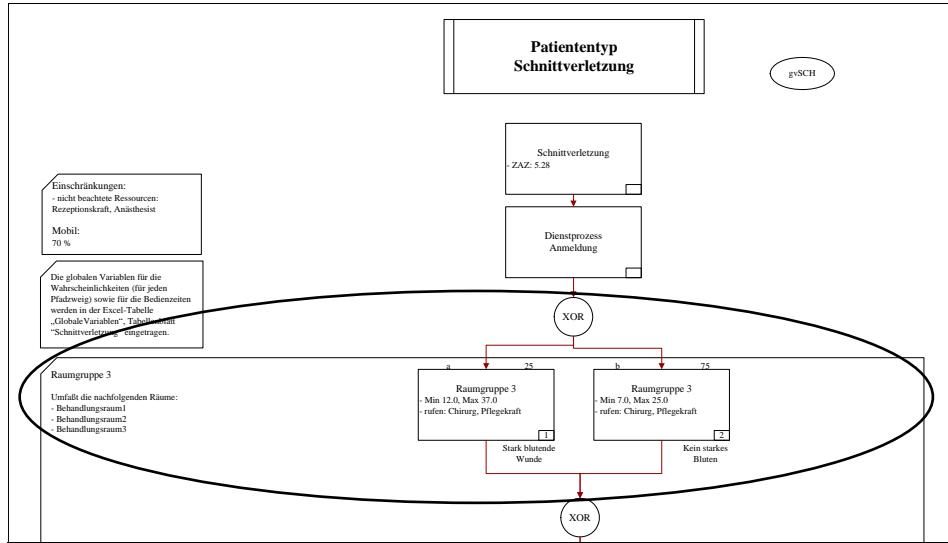


Abbildung 4: Transformationsmodell: Patiententyp Schnittverletzung – korrespond. Ausschnitt

Auf Basis des Transformationsmodells erfolgt anschließend die Modellierung der Abläufe im Simulationswerkzeug. Der wesentliche Vorteil liegt in der nun erreichten direkten Überführbarkeit in das Simulationsmodell. Insgesamt kann man die eEPK als Instrument zur detaillierten Abbildung der Krankenhausprozesse auf Fachkonzeptebene betrachten. Auf DV-Konzeptebene wird die eEPK in das Transformationsmodell überführt um anschließend auf der Implementierungsebene in einem Simulationswerkzeug abgebildet zu werden.

4. Ergebnisse

Das Transformationsmodell wurde im Rahmen der Prozessoptimierung einer Notfallaufnahme entwickelt. Die Prozesse wurden nach der Erhebung zunächst mittels eEPK abgebildet. Anschließend erfolgte die Überführung in das Transformationsmodell und dann die Abbildung im Simulationswerkzeug. Selbst bei einem so abgegrenzten und vergleichsweise übersichtlichen Bereich wie der Notfallaufnahme eines Krankenhauses entstehen, z.B. aufgrund zahlreicher Ablaufvarianten, in der Modellierung bereits komplexe eEPKs. Im Projektverlauf konnte außerdem festgestellt werden, dass die eEPK-Modelle eine sehr unterschiedliche Akzeptanz erfuhren. Dies war neben der Komplexität des Szenarios ein weiterer Grund für die Abbildung in einer Simulation, um grafisch animierte Abläufe modellieren zu können. Das Transformationsmodell erwies sich dabei als wirksames Werkzeug bei der Aggregation der Prozesse und vor allem bei der Identifizierung fehlender quantitativer Daten für die Simulation. Die generelle Nützlichkeit des Vorgehens über ein Transformationsmodell konnte daher im beschriebenen Projekt beispielhaft gezeigt werden. Weiterentwicklungs potenzial besteht insbesondere in einer zukünftig stärker formalen, möglichst regelgestützten

Vorgehensweise zur Transformation der Prozessmodelle in ein Simulationsmodell. Dieser Vorgang trägt derzeit noch zu stark subjektive Züge.

Die Modellierung im Simulationswerkzeug erwies sich aus verschiedenen Gründen als schwierig. Beispielsweise sind die verwendeten Werkzeuge für andere Anwendungsdomänen ausgelegt (vor allem für die Simulation von industriellen Anlagen, Produktion und Logistik), was eine Simulation von Prozessen im Allgemeinen und im Krankenhausumfeld im Speziellen erschwert. Zum Beispiel bestand keine Möglichkeit, eine Ressource, die in einem Prozess gebunden war, aus diesem Prozess herauszulösen. So konnte also keine medizinische Fachkraft, die z. B. einen leichten Notfall behandelte, diese Behandlung unterbrechen, um einen akuten Notfall zu versorgen. Solche Probleme ließen sich häufig nur über Eingriffe in die Programmierung oder Hilfskonstrukte lösen.

Im Ergebnis entstand ein Ist-Modell der Notfallaufnahme, welches vor allem erste Optimierungspotenziale hinsichtlich räumlicher Veränderungen, respektive Erweiterungen nahelegte. Des Weiteren wurden verschiedene Abläufe umgestaltet, die zwar nicht auf Erkenntnissen des Simulationsmodells beruhten, jedoch an diesem ebenfalls nachvollzogen werden konnten.

Die Optimierungsphase des Projektes ist noch nicht abgeschlossen, insofern sind weitere Ergebnisse, insbesondere in der Optimierung der Abläufe zu erwarten. Von der Fachseite wurde jüngst ein verstärkter Bedarf an Prozessmodellen und auch der Optimierung von weiteren Abteilungen signalisiert.

5. Zusammenfassung und Ausblick

Das hier vorgeschlagene Transformationsmodell stellt als modelltechnischer Zwischen- schritt ein wirksames Konzept dar, Prozessmodelle durch geeignete Vereinfachungen und Aggregationen in ein Simulationsmodell zu überführen. Das Verfahren verlangt vom Modellierer eine aktive kreative Gestaltung. Es basiert also nicht auf strengen formalen Regeln zur Überführung einer eEPK in ein Transformationsmodell. Es unterstützt den Modellierer vielmehr vor allem durch die Möglichkeit eines grafisch-visuellen Zwischenschrittes bei der Vereinfachung der eEPK-Modelle für die Nutzung in einem Simulationswerkzeug. Durch dieses Vorgehen können auch sehr komplexe eEPK relativ leicht in Simulationsmodelle übertragen werden. Außerdem lassen sich für die Simulation wichtige quantitative Parameter, die so in der eEPK nicht hinterlegt sind, abbilden. In zukünftigen Arbeiten sollen Regeln gestaltet werden, die den Entwurf des Transformationsmodells formal unterstützen und somit dessen Gestaltung auf eine objektivere Grundlage stellen.

Das Transformationsmodell wurde bislang in der Krankenhausdomäne an zwei *ereignisorientierten* Simulationswerkzeugen² (Enterprise Dynamics™ und Plant Simulation™) angewendet. Damit ist das Modell nicht an ein bestimmtes Werkzeug

²Bei ereignisorientierten Simulationswerkzeugen wird der Simulationsfortschritt durch Ereignisse ausgelöst.

gebunden. Die Eignung des Transformationsmodells für ein *prozessorientiertes* Simulationswerkzeug (z.B. Arena™) wird derzeit geprüft. Der Vorteil der prozessorientierten Simulation liegt in der Möglichkeit der Simulation von Workflows, ohne dass Ereignisse den jeweils nächsten Simulationsschritt auslösen müssen, sondern einfach zeitliche Abfolgen abgebildet werden können.

Bislang ist die Modell-Symbolik noch stark von der Anwendungsdomäne Krankenhaus geprägt. Überlegungen und erste Versuche, in anderen Domänen (Logistik) Prozessmodelle in Transformationsmodelle zu überführen, bestätigen, dass die Anwendbarkeit des Transformationsmodells nicht allein auf den medizinischen Sektor beschränkt ist. Zukünftig soll das Konzept auf weitere Domänen, Prozesse und Simulationswerkzeuge übertragen werden, um daraus einen generischen Ansatz der Überführung von Prozessen in Simulationen zu entwickeln.

Da die Überführung eines Prozessmodells in ein Transformationsmodell nicht auf einem strengen regelbasierten, mathematischen Verfahren beruht, ist eine automatisierte, werkzeuggestützte Transformation bislang nicht möglich. Jedoch werden gegenwärtig Überlegungen angestellt, inwieweit die Modellierung des Transformationsmodells durch Werkzeuge unterstützt werden kann und wie sich die entstandenen Modelle über entsprechende Schnittstellen automatisiert in Simulationsumgebungen überführen lassen. Weiterhin soll die grafische Notation des Modells zukünftig noch besser gestaltet werden.

Literaturverzeichnis

- [Br99] Brettel, M.: Krankenhauslogistik. In: Weber, J.; Baumgarten, H. (Hrsg.): Handbuch Logistik: Management von Material- und Warenflußprozessen. Schäffer-Poeschel, Stuttgart 1999, S. 764-774.
- [BS04] Becker, J.; Schütte, J.: Handelsinformationssysteme. Redline Wirtschaft. Frankfurt/M., 2004.
- [Ha02] Haubrock, M.; Schär, W. (Hrsg.): Betriebswirtschaft und Management im Krankenhaus. 3. Aufl., Huber, Bern, 2002.
- [LK06] Lohfert, C.; Kalmár, P.: Behandlungspfade: Erfahrungen, Erwartungen, Perspektiven. In: Der Internist 47(2006)7, S. 676-683.
- [MN04] Mendling, J.; Nüttgens, M.: Transformation of ARIS Markup Language to EPML. In: Nüttgens, M. (Hrsg.); Rump, F. J. (Hrsg.): Proceedings of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004). Luxemburg: GI e. V., 2004, S. 27-38.
- [NRS03] Neumann, S.; Rosemann, M.; Schwegmann, A.: Simulation von Geschäftsprozessen. In: Becker, J.; Kugeler, M.; Rosemann, M.: Prozessmanagement. Springer Verlag, Berlin, 2003, S. 449-468.

[Re07] Reiter, C.: Business Process Model Converter (BPMC) : Unternehmensübergreifende BPM Plattform für ein integriertes GPM. http://download.microsoft.com/download/e/f/9/ef961d71-9570-449e-905a-87d49ec28c01/HRW_BPMC_2004011.ppt, Abruf am 26.04.2007.

[VB02] Vad, J.; Beer, D. G.; Heinrich, T. ; Huhnt, W.: Werkzeuge zur Planung der Planung. In: 14. Forum Bauinformatik. Düsseldorf : VDI, September 2002, S. 197 204.

Kennzahlenbasierte Analyse von Geschäftsprozessmodellen als Beitrag zur Identifikation von SOA Services

Werner Esswein¹, Jens Weller¹, Jeannette Stark¹, Martin Juhrisch²

¹Technische Universität Dresden
Lehrstuhl für Wirtschaftsinformatik,
insbes. Systementwicklung
01062 Dresden

{werner.esswein|jens.weller|jeannette.stark}@tu-dresden.de

²Westfälische Wilhelms-Universität Münster
Projekt MIRO
Rögenstr. 9-13
48149 Münster
juhrisch@uni-muenster.de

Abstract: Seit 2005 läuft an der Universität Münster ein Projekt zum Aufbau einer Informationssystem-Architektur. Die Architektur folgt dabei dem Paradigma der Serviceorientierten Architektur (SOA). Im Verlauf des Projektes wurden Prozesse dokumentiert, auf deren Grundlage Services implementiert werden sollen. Eine manuelle Ermittlung der Prozesse die als Servicekandidaten geeignet sind hat sich jedoch als ineffizient herausgestellt. In diesem Beitrag wird deshalb ein Ansatz vorgestellt, bei dem die Identifikation von Servicekandidaten durch Kennzahlen unterstützt wird, die automatisiert aus Prozessmodellen generiert werden. Durch die Integration der Kennzahlengenerierung in ein Modellierungswerkzeug konnte der Aufwand der Serviceidentifikation wesentlich verringert werden.

1 Einleitung

Im Jahr 2005 wurde an der Universität Münster, mit Unterstützung der Deutschen Forschungsgesellschaft, das Projekt MIRO (Münster Information System for Research and Organization) gestartet. In diesem Projekt wird das Ziel verfolgt, die technischen und organisatorischen Strukturen der Universität den Anforderungen anzupassen, die heute an eine große Universität gestellt werden. Kernpunkt ist die effektive und effiziente Unterstützung existierender Geschäftsprozesse durch eine flexible Informationssystem-Architektur [BHT07].

Hierfür wurde in den letzten Jahren mit der Erhebung der Prozesse begonnen und auftretende Aktivitäten, generierte und benötigte Daten sowie Bearbeiter erfasst. Auf Basis der entstandenen Prozessmodelle werden derzeit Prozesse mit Automatisierungspotential identifiziert. Diese sollen zum Aufbau einer Service-orientierten Architektur genutzt werden [Er05]. Die Services sollen dabei sowohl wissenschaftliche als auch organisatorische Informationen (z. B. für die Universitätsverwaltung) zur Verfügung stellen.

Als große Herausforderung hat sich dabei die Identifikation derjenigen Prozesse herausgestellt, welche für die Implementierung als Service geeignet sind. Zwar gibt es bereits Methoden für die Identifikation derartiger Servicekandidaten aus Geschäftsprozessmodellen [IS05], jedoch wird die Umsetzung derartiger Methoden im Projekt MIRO durch die große Anzahl an (verteilt erstellten) Prozessmodellen sowie deren starke Verflechtung untereinander erheblich erschwert. In diesem Artikel stellen wir eine Möglichkeit vor, dieses Problem zu lösen. Unser Ansatz basiert auf der Idee, Kennzahlen zur Identifikation von Servicekandidaten aus Geschäftsprozessmodellen abzuleiten. Werden die Modelle, wie in unserem Projekt, elektronisch abgelegt, kann die Kennzahlengenerierung komplett automatisiert und damit die Identifikation der Servicekandidaten wesentlich vereinfacht werden.

Unsere Forschung wird dem Design Science zugeordnet [He04]. Angelehnt an die Forschungsmethodik von Verschuren und Hartog [VH05] beginnen wir unseren Beitrag mit einer Anforderungsanalyse (Abschnitt 2). Anschließend stellen wir im Entwurf Kennzahlen vor, mit denen Servicekandidaten identifiziert werden können (Abschnitt 3). Diese Kennzahlen wurden in der Implementierung von uns in einem Modellierungswerkzeug prototypisch umgesetzt, dessen Nutzung im Projekt MIRO schließlich die Evaluation unserer Lösung ermöglichte (Abschnitt 4). Der Beitrag schließt mit einer kritischen Würdigung und einer Diskussion über zukünftige Forschung.

2 Anforderungsanalyse

2.1 Kennzahlen als Grundlage der Analyse

Kennzahlen werden als prospektives Mittel eingesetzt und erfüllen eine wichtige Informations- und Steuerungsfunktion [St85]. Reichmann und Lachnit betonen insbesondere die Bedeutung von Kennzahlen als Informationen für Entscheidungsprozesse im betrieblichen Umfeld [RL76]. Unter dem Begriff der Kennzahl wird eine verdichtete numerische Messgröße verstanden, die ihren Aussagegehalt unabhängig von ihrer Struktur durch den Bezug auf das Erkenntnisziel gewinnt [St85].

Im Allgemeinen wird an Kennzahlen die Anforderung gestellt, quantitativ erfassbare Sachverhalte in konzentrierter Form darstellen zu können [Re95]. Daraus lassen sich zwei zentrale Eigenschaften einer Kennzahl ableiten: Zum einen wird jeder Kennzahl Informationscharakter unterstellt, da ihr Zweck in der Verdichtung großer Datenmengen zu einer Messgröße liegt, um anschließend Sachverhalte auf deren Grundlage zu beurteilen. Des Weiteren muss Quantifizierbarkeit vorliegen, um Sachverhalte auf einer metrischen Skala messen zu können [Jä01].

An die Entwicklung der Kennzahlen sind folgende Forderungen gebunden:

- Zweckeignung. Der Informationsgehalt der Kennzahl sollte mit dem ursprünglichen Informationsbedarf übereinstimmen.
- Genauigkeit. Die Genauigkeit einer Kennzahl wird durch ihre Zuverlässigkeit und Validität bestimmt.
- Aktualität. Der Zeitraum zwischen Messung und Auswertung der Kennzahl sollte minimal sein.
- Kosten-Nutzen-Relation. Die Erhebung der Kennzahl sollte nicht höhere Kosten verursachen, als ihr Erkenntniswert ist [Ha89].
- Einfachheit und Nachvollziehbarkeit. Das Messergebnis muss einfach interpretiert werden können.

Die Aussagefähigkeit von Kennzahlen lässt sich steigern, wenn man sie zu einem Kennzahlensystem verbinden kann [Es97]. Diese Schlussfolgerung beruht auf der Annahme, dass einzelne bzw. wenige Kennzahlen nicht in der Lage sind, die Komplexität eines Systems vollständig wiederzugeben bzw. eine Vielzahl von Einzelkennzahlen den Blick auf die wesentlichen Sachverhalte verhindert [Wi67]. Bei einer geordneten Gesamtheit von Kennzahlen spricht man demzufolge von einem Kennzahlensystem. Die Kennzahlen stehen dabei in Beziehung zueinander und informieren in ihrer Gesamtheit über einen Sachverhalt [Fr01]. Zur Bildung von Kennzahlensystemen müssen also mindestens zwei oder mehr Einzelkennzahlen vorliegen und miteinander verknüpft werden.

Einzelkennzahlen können entweder aus übergeordneten Kennzahlen abgeleitet werden oder simultan hergeleitet und in einem quantitativen Modell in eine Beziehung zueinander gestellt werden. Fehlen quantitative Zusammenhänge, so lassen sich die Beziehungen zwischen den Kennzahlen auch aus empirischen Zusammenhängen ableiten [Es97]. Die Entwicklung von Kennzahlen und deren Zusammenhängen sollte stets auf einer umfassenden theoretischen Fundierung basieren und nicht theoriefrei der Beantwortung spezifischer Fragestellungen dienen. Für weitere Informationen bzgl. der Entwicklung von Kennzahlensystemen verweisen wir auf [Es97].

In der Regel bezieht sich die Messung von Kennzahlen in Geschäftsprozessmodellen auf einzelne Teilprozesse oder Teilbestandteile des Gesamtmodells. Dieser Ansatz bietet sich insofern an, als dass sich Prozessmodelle vergleichsweise einfach in Teilprozesse untergliedern lassen. Unter Berücksichtigung des Systemgedankens muss jedoch sichergestellt sein, dass Messergebnisse einzelner Teilprozesse im Gesamtzusammenhang betrachtet werden und es somit nicht zu einer Herausbildung von Suboptima kommt [ER02]. Zudem erfordert die mit der Betrachtung von Teilsystemen verbundene Vielzahl von Kennzahlen auch eine Konzentration auf einige wesentliche Kennzahlen, die einen möglichst großen Aussagegehalt in sich vereinen.

Problematisch ist, dass mit Kennzahlen ausschließlich quantitative oder zu quantifizierende Sachverhalte erfasst werden können [Es97]. Werden ergänzende qualitative Aussagen benötigt, die nicht mit Kennzahlen abzubilden sind, bleiben diese weitgehend außerhalb der Betrachtungen.

Der Einsatz von Kennzahlen in Geschäftsprozessmodellen wurde in der Literatur bereits mehrfach diskutiert. Existierende Arbeiten zu diesem Thema fokussieren jedoch zumeist auf der Messung der Komplexität von Geschäftsprozessmodellen um erstens Verständlichkeit [GL06, Ca06] und zweitens Wartbarkeit und Korrektheit von Modellen [Ca06] zu messen. Da in dieser Arbeit Kennzahlen zur Identifikation geeigneter Servicekandidaten genutzt werden, reicht deren Basierung auf Komplexität zur Messung von Wartbarkeit, Korrektheit und Verständlichkeit des Modells jedoch nicht aus, weswegen im Folgenden zunächst Kriterien zum Aufbau der Kennzahlen vorgestellt werden.

2.2 Identifikation von Servicekandidaten

Das SOA Paradigma sieht Services als Elemente vor, um Anwendungen zu entwickeln. Autonome, offene Komponenten exportieren Services, die eine definierte fachliche Funktionalität anbieten und als Bestandteile eines größeren Verarbeitungsablaufs verwendet werden können [RHS05]. Der intendierte Verwendungszweck eines Service steigert den Abstraktionsgrad beim Serviceentwurf gegenüber der Objektorientierung [FJ01]. Services sollen unabhängig voneinander auch in solchen Domänen einsetzbar sein, deren Besonderheiten beim Entwurf nicht berücksichtigt wurden.

Schwemm et al. leiten fünf Entwurfsprinzipien aus der Literatur ab: Bedarfsorientierung, Autonomie, Modularität, Schnittstellenorientierung und Interoperabilität [Sc06]. Services sind bedarfsorientiert, wenn sich ihr Leistungsumfang an den benötigten Objekten orientiert. Sie sind modular und autonom, wenn ihre Ressourcen mit großer Abhängigkeit untereinander so zusammengefasst sind, dass sie gegenüber den anderen Subsystemen eine möglichst geringe Abhängigkeit aufweisen.

Die Entwurfsprinzipien Schnittstellenorientierung und Interoperabilität basieren auf der Annahme, dass Services stabile Schnittstellen darstellen, die über Metadaten technisch und fachlich vollständig spezifiziert sind [Sc06]. Da eine vollständige technische und fachliche Spezifikation von Prozessen in Prozessmodellen unüblich und daher in unserem Projekt nicht vorhanden sind, beschränken wir uns bei der Ableitung von Kennzahlen aus Modellen auf die Prinzipien Bedarfsorientierung, Autonomie und Modularität.

2.2.1 Bedarfsorientierung

Entwurfsrichtlinien der Bedarfsorientierung referenzieren die Granularität der Servicefunktionen. Die Granularität entspricht dem Bereich der Funktionalität, der durch eine Servicefunktion bereitgestellt wird [Gr98]. Ein Service ist bedarfsorientiert, wenn er die zur Erbringung einer spezifischen Leistung benötigten Objekte enthält [Sc06]. Diese Objekte können in der Modellierung als Informationsobjekte dargestellt werden, die im Datenmodell in Verbindung zueinander gesetzt und im Prozessmodell den Prozessen zugeordnet werden. Servicekandidaten sind in diesem Fall ein Prozess bzw. eine Menge von Prozessen, die eine gemeinsame Leistung erbringen und auf gleiche Informationsobjekte zugreifen. Folglich müssen die Informationsobjekte der Prozesse, die einen Servicekandidaten bilden, einen hohen Zusammenhang zueinander aufweisen. Ein Maß für den Zusammenhang von Systemen ist die Kohäsion [Mc76]. Bei einer hohen Kohäsion weisen die Elemente eines Services einen hohen und bei einer geringen Kohäsion nur einen geringen Zusammenhang auf.

2.2.2 Autonomie

Das Ausmaß in dem ein Service autonom ist, bestimmt die Wartbarkeit eines Services. Gemäß Simon sind autonome Systeme gegenüber abhängigen Systemen besser zu warten, da Änderungen an ihnen nur geringe Änderungen an benachbarten Systemen hervorrufen [Si62,WW90]. Simon operationalisiert Autonomie mit Hilfe der Kopplung [Si62]. Kopplung ist ein Maß für den paarweisen Zusammenhang zwischen mehreren Subsystemen [WW90]. Ein einzelner bzw. eine Menge an Prozessen kann als Servicekandidat identifiziert werden, wenn dieser Prozess bzw. diese Menge an Prozessen unabhängig von anderen Prozessen ist. Ein Prozess bzw. eine Menge an Prozessen ist unabhängig von anderen Prozessen, wenn seine Informationsobjekte erstens nicht von anderen Prozessen genutzt werden und zweitens die an anderen Prozessen transferierten Informationsobjekte wenig komplex sind [Yo79]. Der Prozess bzw. die Menge an Prozessen kann demzufolge als Service automatisiert werden, ohne dass die anderen Prozesse gefährdet werden.

2.2.3 Modularität

Durch das Erfüllen der Anforderung der Modularität beim Entwurf von Services, kann die Komplexität des Services reduziert, eine parallele Ausführung von Services realisiert und Unsicherheit beseitigt werden [BC00]. Grundlegende Arbeiten über Modularisierung gehen auf Parnas zurück [Pa71]. Dabei werden abgeschlossene funktionale Einheiten zusammengefasst und mit definierten Schnittstellen ausgestattet [Ba98]. Balzert definiert ein Modul als Darstellung einer funktionalen Einheit oder einer semantisch zusammengehörenden Funktionsgruppe, die abgeschlossen ist, definierte Schnittstellen für Externbezüge besitzt und „im qualitativen und quantitativen Umfang handlich, überschaubar und verständlich“ ist [Ba98]. Modularität wird ähnlich wie die Bedarfsorientierung durch die Kohäsion operationalisiert. Dabei wird versucht Abstandsmaße für verschiedenen Zerlegungen und Beziehungen zu quantifizieren.

3 Entwurf

Gemäß den Entwurfsprinzipien Bedarfsorientierung, Autonomie und Modularität können einzelne Prozesse, aber auch eine Menge über Informationsobjekte verknüpfte Prozesse als ein Servicekandidat identifiziert werden. Einzelne Prozesse werden als Servicekandidat identifiziert, wenn sie durch eine geringe Kopplung und eine hohe Kohäsion gekennzeichnet sind. Eine Menge an Prozessen kann als Servicekandidat identifiziert werden, wenn diese eine hohe Kopplung und eine hohe Kohäsion zueinander und eine geringe Kopplung zu anderen Prozessen aufweisen.

Aus diesen Anforderungen werden nun Kennzahlen zur Identifikation einzelner Prozesse (Abschnitt 3.1) und einer Menge an Prozessen (Abschnitt 3.2) als Servicekandidaten generiert. Des Weiteren werden Anforderungen an Modellierungssprachen zur Umsetzung der Identifikation von Services abgeleitet (Abschnitt 3.3). Um die Nutzung der Kennzahlen zu verdeutlichen, wird für jede Kennzahl ein Beispiel an den in Abbildung 1 dargestellten ARIS Modellen angeführt. Im Folgenden wird keine Unterscheidung zwischen Prozess, Teilprozess bzw. Elementarfunktion vorgenommen. Stattdessen werden alle diese Begriffe und dem Terminus Prozess zusammengefasst.

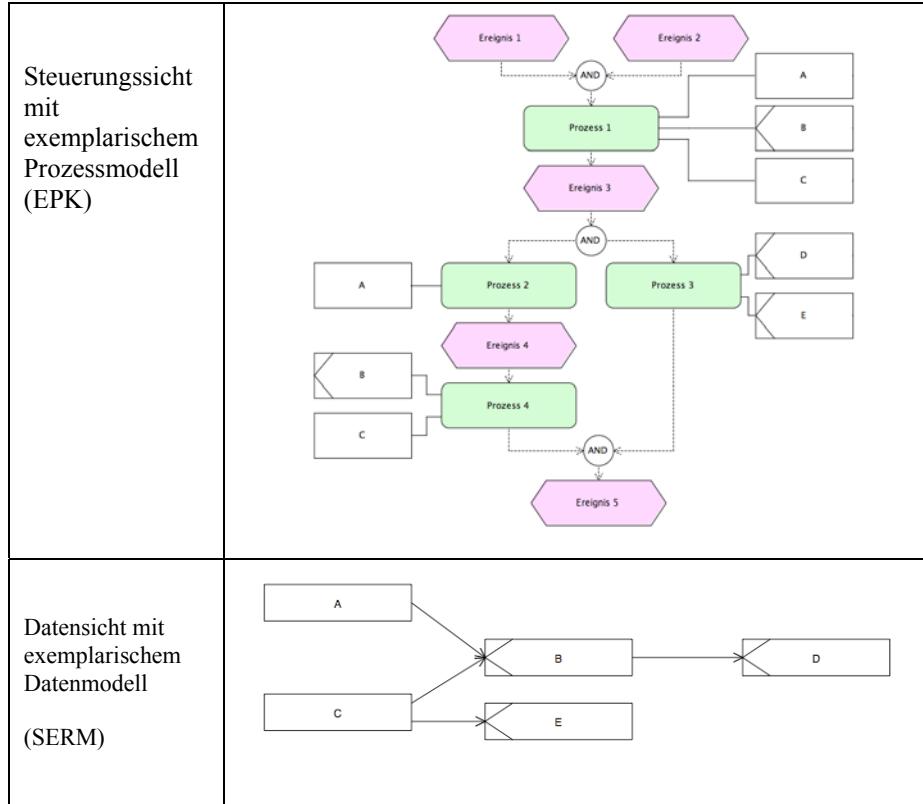


Abbildung 1: Beispielmodell zur Verdeutlichung der Kennzahlen

3.1 Kennzahlen zur Identifikation einzelner Prozesse als Servicekandidaten

Einzelne Prozesse können als Servicekandidat identifiziert werden, wenn deren Informationsobjekte erstens so wenig wie möglich von anderen Prozessen genutzt werden und zweitens die an anderen Prozessen transferierten Informationsobjekte wenig komplex sind [Yo79]. Um herauszufinden, wie viele Informationsobjekte eines Prozesses P_i durch andere Prozesse genutzt werden, bilden wir die Summe der Schnittmengen (S_{ig}) der Informationsobjekte von P_i und den anderen Prozessen. Der Beispielprozess 1 (P_1) nutzt das Informationsobjekt A mit P2 und die Informationsobjekte B und C mit P4 gemeinsam. Die Gesamtschnittmenge an gemeinsam genutzten Informationsobjekten von P_1 mit den anderen Prozessen (S_{lg}) sind A, B und C und deren Betrag drei.

Zur Identifikation der Komplexität der an anderen Prozessen transferierten Informationsobjekte leiten wir die Anzahl der Beziehungen ab, die zur Verbindung der Informationsobjekte im Datenmodell notwendig ist. Die Summe der Anzahl der Beziehungen der jeweiligen Schnittmengen an Informationsobjekten eines Prozesses mit denen anderer Prozesse ergibt die Gesamtkomplexität eines Prozesses $\sum K(S_{ij})$. Die Gesamtkomplexität von P1 ergibt sich beispielsweise aus der Summe der Beziehungen, die zur Verbindung der Informationsobjekte in den Schnittmengen $S_{1,2}$, $S_{1,3}$ und $S_{1,4}$ notwendig sind. $S_{1,2}$ enthält lediglich das Informationsobjekt A, welches nicht mit anderen Informationsobjekten verbunden wird. $S_{1,3}$ enthält keine Informationsobjekte und $S_{1,4}$ die Informationsobjekte B und C. Zur Verbindung von B und C ist eine Beziehung notwendig. Demzufolge hat die Gesamtkomplexität $\sum K(S_{1j})$ eine Ausprägung von eins (vgl. Abbildung 2). Die Kopplung wird nach folgender Formel berechnet:

$$Kopplung_i = |S_{ig}| + \sum_{j=1}^n K(S_{ij})$$

Je weniger ein Prozess mit anderen Prozessen gekoppelt ist, desto unabhängiger ist der betreffende Prozess und desto weniger beeinflusst eine Automatisierung dieses Prozesses die anderen Prozesse. Demzufolge sind Prozesse mit einer geringen Kopplung mögliche Servicekandidaten. In unserem Fallbeispiel ist P3 ein geeigneter Servicekandidat.

	P1 (A,B,C)	P2(A)	P3(D,E)	P4(B,C)
P1 (A,B,C)	-	$S_{2,1}=\{A\}$ $K(S_{2,1})=0$	$S_{3,1}=\{\emptyset\}$ $K(S_{3,1})=0$	$S_{4,1}=\{B,C\}$ $K(S_{4,1})=1$
P2 (A)	$S_{1,2}=\{A\}$ $K(S_{1,2})=0$	-	$S_{3,2}=\{\emptyset\}$ $K(S_{3,2})=0$	$S_{4,2}=\{\emptyset\}$ $K(S_{4,2})=0$
P3 (D,E)	$S_{1,3}=\{\emptyset\}$ $K(S_{1,3})=0$	$S_{2,3}=\{\emptyset\}$ $K(S_{2,3})=0$	-	$S_{4,3}=\{E\}$ $K(S_{4,3})=0$
P4 (B,C)	$S_{1,4}=\{B,C\}$ $K(S_{1,4})=1$	$S_{2,4}=\{\emptyset\}$ $K(S_{2,4})=0$	$S_{3,4}=\{\emptyset\}$ $K(S_{3,4})=0$	-
S_{ig}	$S_{1g}=\{A,B,C\}$	$S_{2g}=\{A\}$	$S_{3g}=\{\emptyset\}$	$S_{4g}=\{B,C,E\}$
$ S_{ig} $	3	1	0	3
$\sum K(S_{ij})$	1	0	0	1
Kopplung	4	1	0	4

Legende:	
S_{ig}	Schnittmenge an gemeinsam genutzten Informationsobjekten des Prozesses i mit den anderen Prozessen
$ S_{ig} $	Betrag der Schnittmenge an gemeinsam genutzten Informationsobjekten des Prozesses i mit den anderen Prozessen
$\sum K(S_{ij})$	Die Summe der Anzahl der Beziehungen der jeweiligen Schnittmengen an Informationsobjekten eines Prozesses i mit denen anderer Prozesse (Gesamtkomplexität)

Abbildung 2: Fallbeispiel für die Kopplung (einzelne Prozesse)

Eine zweite Kennzahl zur Identifikation einzelner Prozesse als Servicekandidaten wird aus der Kohäsion abgeleitet. In diesem Paper wird Kohäsion eines Prozesses durch die Komplexität seiner Informationsobjekte abgeleitet. Demzufolge ist ein Prozess durch eine hohe Kohäsion gekennzeichnet, wenn seine Informationsobjekte eine enge Verbindung zueinander haben und demnach über eine möglichst geringe Anzahl von Beziehungen miteinander verbunden sind. Je geringer die Anzahl der Beziehungen und folglich je kleiner die Komplexität der Beziehungen zwischen den Informationsobjekten ist, desto höher resultiert die Kohäsion und desto geeigneter ist der Prozess als Servicekandidat. Um eine Normierung der Prozesse zu erreichen, führen wir im Folgenden N als die Anzahl der von einem Prozess genutzten Informationsobjekte ein. Folglich ergibt sich für die Kohäsion folgende Kennzahl:

$$Kohäsion_i = N - K_i$$

Für unser Fallbeispiel lassen sich die in Abbildung 3 genannten Kennzahlen ableiten. P1 nutzt beispielsweise die Informationsobjekte A, B und C. Diese drei Informationsobjekte sind durch zwei Beziehungen miteinander verbunden. Demzufolge hat die Komplexität K_1 der Informationsobjekte A, B und eine Ausprägung von zwei. Da P1 drei Informationsobjekte nutzt, resultiert nach Abzug einer Komplexität von zwei eine Kohäsion von eins. Je höher die Kohäsion, desto geeigneter ist der Prozess als Servicekandidat. In unserem Beispiel bilden demzufolge P1 und P2 geeignete Servicekandidaten.

Prozess	K_i	Kohäsion
P1 (A,B,C)	2	1
P2 (A)	0	1
P3 (D,E)	3	-1
P4 (B,C)	1	-1

Legende:	
K_i	Anzahl der Beziehungen der Schnittmengen an Informationsobjekten eines Prozesses i mit denen anderer Prozesse (Komplexität)

Abbildung 3: Fallbeispiel für die Kohäsion (einzelne Prozesse)

3.2 Kennzahlen zur Identifikation gemeinsamer Prozesse als Servicekandidaten

Eine Menge an Prozesse stellt ein Servicekandidat dar, wenn diese Prozesse zueinander durch eine hohe Kopplung und eine hohe Kohäsion und zu anderen Prozessen durch eine geringe Kopplung gekennzeichnet sind. Services sollten eine angemessene Granularität aufweisen. Im Falle einer sehr groben Granularität wird die Wiederverwendung des Services reduziert und im Falle einer sehr flachen Granularität das Verständnis der Prozesse durch fehlenden Kontext erschwert [AS04, SW05]. Aus diesem Grund beschränken wir uns exemplarisch auf die Untersuchung zweier Prozesse als ein Servicekandidat. Analog unseres Vorgehens können jedoch weitere Prozesse sukzessiv hinzugefügt werden, so dass die Anwendung nicht auf zwei Prozesse beschränkt bleibt. Zur Identifikation zweier Prozesse als ein Servicekandidat wird die Anzahl, der in der Schnittmenge gemeinsam genutzter Informationsobjekte genutzt und ihr Maximum bestimmt:

$$Kopplung_i = \max |S_{ij}|$$

Die Prozesse, die zu einem Servicekandidaten zusammengefasst werden können, lassen sich aus der maximalen Anzahl von Informationsobjekten in ihrer Schnittmenge bestimmen. Wie in Abbildung 4 ersichtlich ergeben sich Schnittmengen für P1 und P2 sowie für P1 und P4. Dabei weisen P1 und P4 die größte Schnittmenge von zwei Informationsobjekten auf.

	P1 (A,B,C)	P2(A)	P3(D,E)	P4(B,C)
P1 (A,B,C)	-	$S_{2,1}=\{A\}$	$S_{3,1}=\{\emptyset\}$	$S_{4,1}=\{B,C\}$
P2 (A)	$S_{1,2}=\{A\}$	-	$S_{3,2}=\{\emptyset\}$	$S_{4,2}=\{\emptyset\}$
P3 (D,E)	$S_{1,3}=\{\emptyset\}$	$S_{2,3}=\{\emptyset\}$	-	$S_{4,3}=\{\emptyset\}$
P4 (B,C)	$S_{1,4}=\{B,C\}$	$S_{2,4}=\{\emptyset\}$	$S_{3,4}=\{\emptyset\}$	-
$ S_{ij} $	$ S_{12} =1$ $ S_{13} =0$ $ S_{14} =2$	$ S_{21} =1$ $ S_{23} =0$ $ S_{24} =0$	$ S_{31} =0$ $ S_{32} =0$ $ S_{34} =0$	$ S_{41} =2$ $ S_{42} =0$ $ S_{43} =0$
Kopplung	2	1	0	2

Legende:	
$ S_{ig} $	Betrag der Schnittmenge an gemeinsam genutzten Informationsobjekten des Prozesses i mit den anderen Prozessen

Abbildung 4: Fallbeispiel Kopplung gemeinsamer Prozesse

Prozesse, die gemeinsam einen Servicekandidaten ergeben, sind durch eine geringe Kopplung zu anderen Prozessen gekennzeichnet. Für unser Fallbeispiel ergeben sich die gemeinsamen Prozesse P12 aus P1 und P2 und P14 aus P1 und P4. Durch Zusammenfassen aller Informationsobjekte von P1 und P2 zu P12 und von P1 und P4 zu P14 kann nun die Formel für die Identifikation einzelner Prozesse als Servicekandidaten ($Kopplung = |S_{ig}| + \sum K(S_{ij})$) angewendet werden. Abbildung 5 zeigt, dass sich für Prozess P12 eine Kopplung von 3 und für P14 eine Kopplung von 1 zu den anderen Prozessen ergibt. Demzufolge ist P14 dem Prozess P12 vorzuziehen.

	P1 (A,B,C)	P2(A)	P3(D,E)	P4(B,C)	$ S_{ij} $	$K(S_{ij})$	Kopp- lung
P12 (A,B,C)	-	-	$S_{12,3} = \{\emptyset\}$	$S_{12,4} = \{B, C\}$	$ S_{12,3} =0$ $ S_{12,4} =2$	$K(S_{12,3})=0$ $K(S_{12,4})=1$	3
P14 (A,B,C)	-	$S_{14,3} = \{A\}$	$S_{14,2} = \{\emptyset\}$	-	$ S_{14,2} =1$ $ S_{14,3} =0$	$K(S_{14,2})=0$ $K(S_{14,3})=0$	1

Legende:	
S_{ij}	Schnittmenge an gemeinsam genutzten Informationsobjekten des Prozesses i mit den anderen Prozessen

Abbildung 5: Fallbeispiel Kopplung gemeinsamer Prozesse zu anderen Prozessen

Prozesse, die gemeinsam einen Servicekandidaten ergeben, sind des Weiteren durch eine hohe Kohäsion gekennzeichnet. Auch hier können gemeinsame Prozesse wieder auf einzelne Prozesse zurückgeführt werden, so dass die Formel Kohäsion = $N - K_i$ zur Anwendung kommt. Die Kohäsion für P12 und P14 sind in Abbildung 6 zusammengefasst. Da P12 und P14 auf gleichen Informationsobjekten zurückgreifen, weisen beide die gleiche Kohäsion von eins auf und können demzufolge aufgrund der Kohäsion nicht unterschieden werden.

Prozess	K_i	Kohäsion
P12 (A,B,C)	2	1
P14 (A,B,C)	2	1

Legende:	
K_i	Anzahl der Beziehungen der Schnittmengen an Informationsobjekten eines Prozesses i mit denen anderer Prozesse (Komplexität)

Abbildung 6: Fallbeispiel für die unabhängige und abhängige Kopplung

3.3 Kennzahlengenerierung mit Hilfe der Ereignisgesteuerten Prozesskette (EPK)

Um die obigen Kennzahlen zu ermitteln, müssen die zur Bildung der Kennzahlen notwendigen Informationen im Modell enthalten sein. Diese Informationen können generiert werden, wenn erstens Informationsobjekte durch Beziehungen miteinander verbunden werden können. Durch Verbindung der Beziehungen zwischen den Informationsobjekten ist die Komplexität dieser Objekte ableitbar. Zweitens, ist die Abbildung von Beziehung zwischen Prozessen und von deren genutzten Informationsobjekten zu ermöglichen. So sind beispielsweise den verschiedenen Prozessen gemeinsam mit anderen Prozessen genutzte Informationsobjekte als auch deren Komplexität zuzuordnen.

Die erweiterte EPK selbst erlaubt die Zuordnung von Informationsobjekten zu ihren Prozessen, aber nicht die Verbindung der Informationsobjekte durch Beziehungen. Die EPK kann jedoch in Verbindung mit der Architektur integrierter Informationssysteme (ARIS; [Sc92]) genutzt werden, die eine Verbindung der Informationsobjekte durch Beziehungen innerhalb der Datensicht erlaubt. Somit erfüllt die EPK im ARIS-Kontext die Bedingungen zur Ableitung der Kennzahlen.

4 Implementierung und Evaluation

4.1 Prozesserfassung & Kennzahlengenerierung

Für die Dokumentation der Prozesse wird im Projekt MIRO das Modellierungswerkzeug Cubetto Toolset [Cu08] eingesetzt. Dies wurde uns durch die Technische Universität Dresden kostenfrei zur Verfügung gestellt. Das Werkzeug bietet drei wesentliche Vorteile, die für den Einsatz im Projekt MIRO von Bedeutung sind: Erstens kann die verwendete Modellierungssprache den Bedürfnissen im Projekt angepasst werden und so bspw. die für die Analyse zusätzlich notwendigen Konzepte eingepflegt werden. Zweitens wird die verteilte Erfassung von Prozessen durch ein integriertes Konfigurationsmanagement (KM) System samt KM Server ermöglicht und drittens ist es mit Hilfe von Plugins zu erweitern, wodurch die für die Analyse notwendige Funktionalität ohne Kenntnis des Quellcodes hinzugefügt werden konnte.

Für die Modellierung wurde im Projekt MIRO die Architektur integrierter Informationssysteme (ARIS) verwendet. Diese ist im Cubetto Toolset verfügbar und erfüllt die Anforderungen aus Abschnitt 3.3. Da die Erfassung der einzelnen Geschäftsprozesse innerhalb der Fachabteilungen (Universitäts- und Landesbibliothek, Studierendensekretariat, Rechenzentrum, Prüfungsämter) erfolgte, eine Analyse jedoch für das Gesamtmodell erfolgen sollte, mussten die einzelnen Teilmodelle integriert werden. Dieses konnte durch das im Werkzeug integrierte KM-System realisiert werden

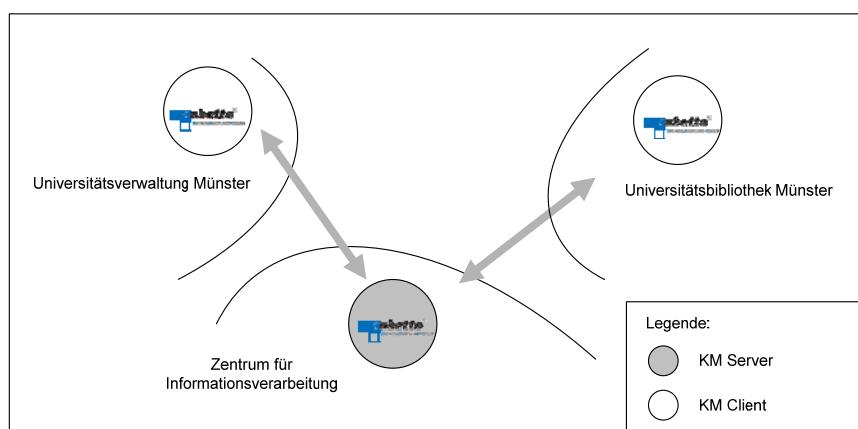


Abbildung 7: Verteilte Prozesserfassung an der Universität Münster

Hierfür wurde zunächst ein initiales Modell erzeugt und auf dem KM-Server abgelegt. Die Fachabteilungen nutzen nun den Checkin/Checkout Mechanismus des Servers und arbeiten auf diese Weise parallel an einem Gesamtmodell (vgl. Abbildung 7). Durch die Aufteilung des Modells in einzelne Sichten (Teilmodelle) konnte dennoch eine Konzentration auf die eigenen Prozesse erfolgen. Die Integration der Teilmodelle wurde während der Prozesserfassung durch einen Modellierungsexperten sichergestellt, der die einzelnen Fachexperten zur Nutzung existierender Modellelemente in verschiedenen Teilmodellen motivierte und die formale Qualitätskontrolle der Modelle übernahm.

Für die Erzeugung der Kennzahlen musste das Cubetto Toolset um entsprechende Funktionen zur Auswertung der abgelegten Modelle erweitert werden. Hierfür wurde prototypisch ein Cubetto-Plugin erstellt, welches die abgelegten Modelle analysiert und die oben vorgestellten Kennzahlen generiert (vgl. Abbildung 8). Die Kennzahlen konnten anschließend zur Identifikation der Servicekandidaten herangezogen werden.

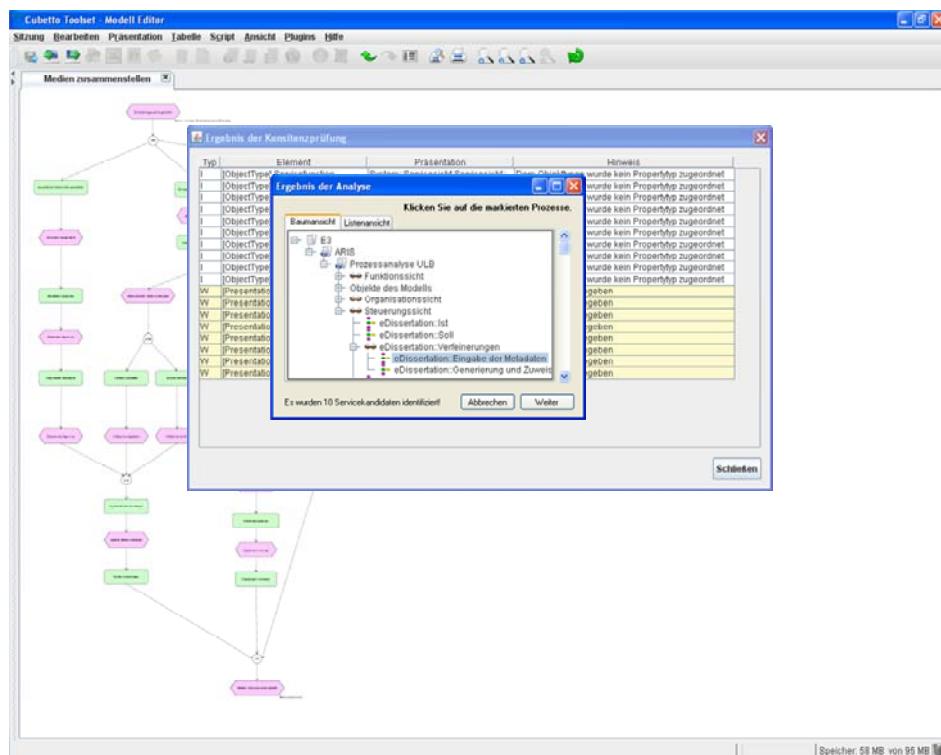


Abbildung 8: Generierte Kennzahlen

4.2 Evaluation

Da die Teilmodelle der Universität im Rahmen der Prozesserfassung bereits zu einem Gesamtmodell integriert wurden, konnten mit Hilfe des Plugins auch komplexe Verflechtungen erkannt werden. Dies ermöglichte es uns im Projekt MIRO Servicekandidaten zu identifizieren, welche universitätsweit eingesetzt werden können. Dadurch wird die Anzahl an Services insgesamt niedrig gehalten, ohne auf Funktionalität verzichten zu müssen.

Mit Hilfe der automatisierten Analyse zur Identifikation von Servicekandidaten wurden im Projekt MIRO bereits 68 Prozesse aus den vorliegenden Geschäftsprozessmodellen als Servicekandidaten identifiziert, wobei die für die Analyse notwendige Zeit im Vergleich zu einer manuellen Lösung wesentlich verkürzt werden konnte. Da Kennzahlen Entscheidungen immer nur unterstützen können, war eine anschließende Beurteilung der Servicekandidaten notwendig, bevor sie in die Implementierung einfließen konnten. Hierbei zeigte sich, dass nicht alle Servicekandidaten tatsächlich auch als Service implementiert werden konnten, da sie entweder zu grob- oder aber zu feingranular waren.

So stellte sich bspw. der Prozess „sende Email an Empfänger“ als zu feingranular dar. Die Funktion ist Ergebnis einer aus fachlicher Sicht nicht sinnvollen Zerlegung eines betrieblichen Prozesses in nahezu atomare Funktionen. Zwangsläufig wurde dabei „Mail“ als Objekt mitmodelliert. Einzige Funktion von „sende Email an Empfänger“ ist es, einer übergebenen Person eine Mail mit Passwort-Informationen zuzuschicken. Damit liegt ein Grad der Granularität vor, der dem Entwurfsprinzip der Bedarfsgerechtigkeit (vgl. Abschnitt 2.2.1) entgegensteht. Demnach soll dieser so gewählt werden, dass die Granularität fachlichen Funktionen entspricht und eben nicht bloßen technischen Grundfunktionen. Eine Implementierung des Service wurde daher nicht durchgeführt.

Wir konnten jedoch auch das andere Extrem identifizieren. So wurde aufgrund der Kennzahlenanalyse der Prozess „Prüfungsabwicklung und Notenverbuchung“ ermittelt. Dieser ist jedoch zu grobgranular, so dass auch hier auf eine direkte Umsetzung verzichtet wurde. Stattdessen ist eine weitere Analyse aus Sicht der Softwareentwicklung notwendig. Im vorliegenden Fall kann die Servicefunktion schwerwiegende Auswirkungen in den jeweiligen Systemen haben (z. B. Exmatrikulation wegen endgültig nicht bestandener Prüfung) und muss in ganz besonderer Weise gegen Missbrauch und Manipulation gesichert werden.

5 Zusammenfassung

Mit Hilfe des vorgestellten Ansatzes und der Implementierung im Cubetto Toolset konnten im Projekt MIRO Servicekandidaten identifiziert werden. Die Verwendung der automatisch generierten Kennzahlen führte dabei zu einer wesentlichen Zeiterbsparnis. Wie wir im Rahmen der Evaluation dargelegt haben, führten die ermittelten Kennzahlen jedoch auch zu Servicekandidaten deren Implementierung entweder nicht sinnvoll war weiteren Analysebedarf nach sich zogen. Dies kann darauf zurückgeführt werden, dass wir uns bei der Ermittlung der Kennzahlen nicht auf bestimmte Abstraktionsebenen im Modell eingeschränkt haben.

Die Gründe hierfür sind, dass eine objektive Unterteilung komplexer Modelle in Bezug auf den Abstraktionsgrad schwer bzw. gar nicht möglich ist, weil es die Existenz solcher fest definierten Abstraktionsebenen unterstellen würde. Dies ist bei Verwendung eines konstruktivistischen Modellverständnisses jedoch auszuschließen. Zur Vermeidung der Konflikte beim automatisierten Vergleich und der Bewertung verteilt entwickelter Geschäftsprozessmodelle setzen verschiedene Ansätze am Prozess der Erstellung der Modelle. Um die Konstruktion einfach vergleichbarer Modellierungsartefakte zu begünstigen, wird eine an Konventionen orientierte konstruktive Einschränkung der Freiheitsgrade bei der fachkonzeptuellen Modellierung vorgenommen. Ansätze dieses Bereichs basieren auf der Annahme, dass der Vergleich von Modellen mit willkürlicher Struktur die Identifikation von semantischen Überlappungsbereichen und unterschiedlicher Abstraktionsgrade behindert. Für eine detaillierte Beschreibung eines diesbezüglich anwendbaren Ansatzes wird auf [JW08] verwiesen.

Weiterer Forschungsbedarf leitet sich aus den Serviceentwurfsprinzipien ab. So wurden in diesem Beitrag Kennzahlen aus den Prinzipien Bedarfsorientierung, Autonomie und Modularität abgeleitet wurden. Keine Kennzahlen wurden bisher aus den verbleibenden Serviceentwurfsprinzipien Schnittstellenorientierung und Interoperabilität abgeleitet. Die Eignung dieser Prinzipien ist in zukünftiger Forschung zu überprüfen.

Darüber hinaus ist die grundlegende Vorgehensweise, Geschäftsprozesse automatisiert zu analysieren auch für andere Anwendungsszenarien denkbar. So können beispielsweise im Rahmen einer Prozessverbesserung Medienbrüche innerhalb von Prozessen identifiziert werden und so Hinweise auf Verbesserungspotentiale bieten. Die Zusammenführung von verschiedenen Kennzahlen in einem zentralen Prozessrepository und die Integration mit Informationssystemen des Controllings bieten dabei weiteren Forschungsbedarf für die Wirtschaftsinformatik.

6 Literaturverzeichnis

- [AS04] Aier, S.; Schönherr, M.: Modularisierung komplexer Unternehmensstrukturen. In Industrie Management 2 (4), S.39-42

- [Ba98] Balzert, H.: Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum, Heidelberg, Berlin, 1998; S. 571-574
- [BC00] Baldwin, C.Y.; Clark, K.B.: Design Rules, The power of modularity, The MIT Press, Cambridge, 2000
- [BHT07] Böhm, B.; Held, W.; Tröger, B.: Integriertes Informationsmanagement einer großen Universität. In (Degkwitz, A.; Schirmbacher, P. Hrsg.): Informationsinfrastrukturen im Wandel. Informationsmanagement an deutschen Universitäten, Bock + Herchen Verlag, Bad Honnef, 2007
- [Ca06] Cardoso, J.; Mendling, J.; Neumann, G.; Reijers, H.: A discourse on complexity of process models. In (Eder, J.; Dustar, S. Hrsg.) Proceedings of the BPM Workshop, 2006
- [Cu08] Cubetto: Specification, Semture GmbH, Dresden, <http://www.semture.de/cubetto>. 2008
- [Er05] Erl, T.: Service-oriented architecture: concepts, technology, and design, Prentice Hall PTR, 2005
- [ER02] Engelke, M.; Rausch, A.: Supply Chain Management mit Hilfe von Key Performance Indikatoren. In (Stölzle, W.; Gareis, K. Hrsg.): Integrative Management- und Logistikkonzepte. Festschrift für Prof. Dr. Dr. h.c. H.-Chr. Pfohl, Wiesbaden, 2002; S. 205-235
- [Es97] Ester, B.: Benchmarks für die Ersatzteillogistik: Benchmarkingformen, Vorgehensweise, Prozesse und Kennzahlen. In (Pfohl, C. Hrsg.): Unternehmensführung und Logistik, Erich-Schmidt Verlag, 1997
- [FJ01] Frank, U.; Jung, J.: Prototypische Vorgehensweise für den Entwurf anwendungsnaher Komponenten. In Proceedings of the Verbundtagung VertIS, 2001.
- [Fr01] Frank, U.: Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze. In Wirtschaftsinformatik 43 (3), 2001; S. 283-293
- [GL06] Gruhn, V.; Laue, R.: Complexity metrics for business process modes. In (Abramowics, W.; Mayr, H. Hrsg.) Proceedings of the 9th international conference on business information systems, 2006
- [Ha89] Haufs, P.: DV-Controlling: Konzeption eines operativen Instrumentariums aus Budgets, Verrechnungspreisen, Kennzahlen. Heidelberg, 1989
- [JW08] Juhrisch, M.; Weller, J.: Connecting Business and IT: A model-driven Webservice based Approach. In: Proceedings of the 12th Pacific Asia Conference on Information Systems (PACIS), 2008, S. 1469-1479
- [MC76] McCabe, T.J.: A complexity measure. In IEEE Trans. Software Eng. 2 (4), 1976; S.308-320

- [Gr98] Griffel, F.: Componentware. Konzepte und Techniken eines Softwareparadigmas. dPunkt, Heidelberg, 1998
- [IS05] Ivanov, K.; Stähler, D.: Prozessmanagement als Voraussetzung für SOA. In OBJEKTspektrum 12, 2005; S. 1-5
- [He04] Hevner, A.R.; March, S.T.; Park, J.; Sam, S.: Design Science in Information Systems Research. In MIS Quarterly 28 (1), 2004; S.75-105
- [Jä01] Jäger-Goy, H. Führungsinstrumente für das IV-Management, Dissertation, Johannes-Gutenberg-Universität Mainz, Frankfurt am Main, 2001; S. 127
- [Pa71] Parnas, D.L.: Information distribution aspects of design methodology. In (Freiman, C.V.; Griffith, J.E.; Rosenfeld, J.L. Hrsg.): Information Processing 71 (1), North-Holland, 1971; S. 339-344
- [Re95] Reichmann, T.: Controlling mit Kennzahlen und Managementberichten: Grundlage einer systemgestützten Controlling-Konzeption, 4. Aufl., Vahlen, München, 1995
- [RHS05] Richter, J.P.; Haller, H.; Schrey, P.: Serviceorientierte Architektur. In Informatik-Spektrum 28, 2005; S.413-416
- [RL76] Reichmann, R.; Lachnit, L.: Planung, Steuerung und Kontrolle mit Hilfe von Kennzahlen. In Schmalenbachs Zeitschrift für betriebliche Forschung 28, 1976; S. 705 – 723.
- [Sc06] Schwemm, J.W.; Heutschi, R., Vogel, T.; Wende, K.; Legner, C.: Serviceorientierte Architekturen: Einordnung im Business Engineering. Working Paper of the HSG, University of St. Gallen, 2006.
- [Si62] Simon, H.A.: The architecture of complexity. In Proceedings of the American Philosophical Society 106, 1962.
- [St85] Staudt, E. et al.: Kennzahlen und Kennzahlensysteme. Grundlagen zur Entwicklung und Anwendung. Berlin, 1985; S. 20-24
- [SW05] Schwinn, A.; Winter, Ro: Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration. In (Ferstl, O.K.; Sinz, E.J.; Eckert, S.; Isselhorst, T. Hrsg.): Wirtschaftsinformatik. Physica-Verlag, Heidelberg 2005
- [VH05] Verschuren, P.; Hartog, R.: Evaluation in Design-Oriented Research. In Quality & Quantity 39, 2005; S. 733-762
- [Wi67] Wissenbach, H.: Betriebliche Kennzahlen und ihre Bedeutung im Rahmen der Unternehmerentscheidung – Bildung, Auswertung und Verwendungsmöglichkeiten von Betriebskennzahlen in der unternehmerischen Praxis. Berlin; S. 15
- [WW90] Wand, Y., Weber, R.: An ontological model of an information system. In IEEE Trans. Software Eng. 16 (11), 1990; S.1282-1292

- [Yo79] Yourdon, E.: Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Prentice Hall, Upper Saddle River, 1979.

Integrierte Produkt- und Prozessmodellierung: Rahmenkonzept und Anwendungsfall zur EU-Dienstleistungsrichtlinie

Frank Hogrebe¹; Markus Nüttgens²

¹Landeshauptstadt Düsseldorf
Organisations-, Personal-, IT- und Wirtschaftsförderungsdezernat
KompetenzCenter eGovernment und neue Medien
Burgplatz 1, D-40213 Düsseldorf
frank.hogrebe@stadt.duesseldorf.de

²Universität Hamburg
Fakultät Wirtschafts- und Sozialwissenschaften
Lehrstuhl für Wirtschaftsinformatik
Von-Melle-Park 9, D-20146 Hamburg
markus.nuettgens@wiso.uni-hamburg.de

Abstract: Die Zuständigkeiten für Produkte und Prozesse liegen im Regelfall verteilt im Organisations-, IT- oder Finanzbereich. Zusammenhänge und Interdependenzen werden demzufolge oftmals nur unzureichend erkannt. Integrierte Produkt- und Prozessmodelle (IPP) sind ein viel versprechender Ansatz, diesen Defiziten zu begegnen. Verteilte Zuständigkeiten sind aufgrund der traditionellen Arbeitsteilung in öffentlichen Verwaltungen der Regelfall. Diese sind mit Blick auf die EU-Dienstleistungsrichtlinie bis Ende 2009 gefordert, ihre Produkt- und Prozessorganisation neu auszurichten. Wesentliche Kernanforderungen der Richtlinie sind die Einrichtung einheitlicher Ansprechpartner für Unternehmen und die elektronische Verfahrensabwicklung von Formalitäten und Verfahren zur Aufnahme und Ausübung einer Dienstleistungstätigkeit. Dies hat unmittelbare Auswirkungen auf die Ausgestaltung der zugrunde liegenden Informationssysteme und der IT-Infrastruktur, besonders bezogen auf die eGovernment-Angebote öffentlicher Verwaltungen. Ausgehend von einem IPP-Rahmenkonzept beschreibt der Beitrag einen Modellierungsansatz, der auf Basis objektorientierter Ereignisgesteuerter Prozessketten (oEPK) ein Integriertes Produkt- und Prozessmodell für die öffentliche Verwaltung zum Ziel hat. Am Anwendungsfall der Gewerbe-Anmeldung wird der Ansatz im Rahmen der Implementierung der Dienstleistungsrichtlinie konkretisiert.

1 Einführung

Wesentliche Zielsetzung einer integrierten Produkt- und Prozessmodellierung ist ein gemeinsames Modellverständnis von Organisatoren und Modellierern auf der einen Seite und die Wiederverwendbarkeit von Prozessmodellen, bausteinbasierten Diensten und Softwarekomponenten auf der anderen Seite. Hieraus resultiert eine flexiblere Anpas-

sung bei Prozessänderungen und damit kürzere und kostengünstigere Implementierungszeiten. Integrierte Produkt- und Prozessmodelle (IPP) sind ein viel versprechender Ansatz, die Anforderungen und die resultierende Komplexität modellbasiert abzubilden. Das Nutzenpotenzial eines IPP kann auf Basis einer dienstebasierten Architektur weiter verstärkt werden. Dies erfordert aber auch einen ganzheitlichen Ansatz im Systementwurf und einen neuen Grad der Zusammenarbeit in der IT, den Organisationsabteilungen und über die Organisationsbereiche hinweg bis zur Kundenintegration (Konzept des Ko-Produzenten). Für den Industriesektor liegen konzeptionelle Ansätze zur Integration von Produkt- und Prozessmodellen vor (vgl. z.B. [ES05], [We04], [Kl03], [Br00]), im Bereich der öffentlichen Verwaltung haben sich diese bisher nicht etabliert.

Im Diskursbereich öffentliche Verwaltung steht E-Government aus der Sicht der Europäischen Kommission vor einem entscheidenden Wendepunkt. Weitere wesentliche Fortschritte sind danach nur noch dann möglich, wenn bestimmte grundlegende Voraussetzungen geschaffen werden. Vor diesem Hintergrund hat die EU-Kommission im Frühjahr 2006 den „E-Government-Aktionsplan im Rahmen der i2010-Initiative: Beschleunigte Einführung elektronischer Behördendienste in Europa zum Nutzen aller“ zum europaweiten Zugang zu elektronischen Behördendiensten für den Zeitraum bis 2010 aufgestellt [EU06a]. Im Rahmen der Schaffung der Voraussetzungen kommt der Richtlinie 2006/123/EG des Europäischen Rates über Dienstleistungen im Binnenmarkt, kurz EU-Dienstleistungsrichtlinie [EU06b], eine besondere Bedeutung zu, die hohe Anforderungen an die öffentliche Verwaltung stellt.

Ausgehend von einem IPP-Rahmenkonzept beschreibt der Beitrag einen Modellierungsansatz, der auf Basis objektorientierter Ereignisgesteuerter Prozessketten (oEPK) ein Integriertes Produkt- und Prozessmodell für die öffentliche Verwaltung zum Ziel hat. Der Beitrag ist wie folgt aufgebaut: Im zweiten Abschnitt wird zunächst das zugrunde liegende Forschungsdesign dargestellt. Anschließend werden die Kernanforderungen an die öffentliche Verwaltung zur Umsetzung der EU-Dienstleistungsrichtlinie beschrieben. Am Anwendungsfall der Gewerbe-Anmeldung wird auf Basis eines IPP-Rahmenkonzeptes der Modellierungsansatz im Kontext der Umsetzung der EU-Dienstleistungsrichtlinie konkretisiert. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf den weiteren Forschungsbedarf.

2 Forschungsdesign

Der Beitrag ordnet sich in eine Forschungsarbeit ein, deren zentrale Zielsetzung die (1) Gestaltung und (2) Evaluation eines Integrierten Produkt- und Prozessmodells ist, das sich in Form eines Referenzmodells und einer generalisierbaren Modellierungsmethodik für (3) den Anwendungsbereich der öffentlichen Verwaltung eignet. Dabei erfolgt eine Fokussierung auf (4) die Anforderungen der EU-Dienstleistungsrichtlinie an (5) die kommunale Verwaltungsebene (Städte, Kreise und Gemeinden). Diese ist in besonderem Maße gefordert, ihre Strukturen und Abläufe bis Ende 2009 auf die EU-Dienstleistungsrichtlinie auszurichten, da sie die meisten Verfahrens- und Entscheidungszuständigkeiten des Staates in Deutschland auf sich vereint [BM06, S. 15]. In Deutschland sind dies allein über 12.400 Städte und Gemeinden [BBR08]. Das Konzept

der Forschungsarbeit gliedert sich in drei Teilbereiche: (a) Integrierte Produkt- und Prozessmodellierung, (b) Anwendungsfälle und (c) Vergleichsstudien. Erste Arbeitsergebnisse zu den Teilbereichen finden sich zu (a) bei [HN08], zu (b) bei [HKN08] und zu (c) bei [HBN08].

2.1 Forschungsfrage

Der Teilbereich „Integrierte Produkt- und Prozessmodellierung“ bildet den „Kernbereich“ der Forschungsarbeit. Untersuchungsschwerpunkt sind die Grundlagen zur integrierten Produkt- und Prozessmodellierung, welche die konzeptionelle, methodische und technische Basis zum Aufbau von integrierten Produkt- und Prozessmodellen legen. Der Untersuchungsgang dieses Beitrages wird von folgender Forschungsfrage geleitet: Sind objektorientierte Ereignisgesteuerte Prozessketten (oEPK) eine geeignete fachkonzeptuelle Ausgangsbasis, betriebswirtschaftlich relevante Prozessinformationen mit UML-Modellierungskonzepten zu verknüpfen, um auf dieser Basis die Grundlage für einen systematischen und prozesskonformen Aufbau einer dienstbasierten Architektur legen zu können?

2.2 Forschungsmethodik

Der Methodenbegriff in der Wirtschaftsinformatik kann grob in Entwicklungsmethoden (zur Informationssystemgestaltung) auf der einen Seite und in Forschungsmethoden als Instrument der Erkenntnisgewinnung auf der anderen Seite unterschieden werden [WH07, S. 281]. Die erkenntnistheoretische Methodologie unterteilt sich weiter in designwissenschaftliche (konstruktionswissenschaftliche) und behavioristische (verhaltenswissenschaftliche) Forschungsparadigmen. Die Teilbereiche der Forschungsarbeit stellen unterschiedliche Anforderungen an die Forschungsmethodik und werden in Folge dessen mit unterschiedlichen Forschungsmethoden bearbeitet. Zum Methodenspektrum in der Wirtschaftsinformatik vgl. z.B. [WH07], [Ha02], [Kö96], [He95].

Die Bearbeitung des Teilbereiches „Integrierte Produkt- und Prozessmodellierung“ erfolgt mittels der Forschungsmethode der Referenzmodellierung und der -darauf aufbauenden- Entwicklung und Test eines Prototypen (sog. Prototyping). Ziel der Referenzmodellierung ist die modellbasierte Abbildung einer geplanten oder optimierten Realität [WH06, S.7]. Zentrale Zielsetzung der Forschungsarbeit ist die Gestaltung und Evaluation eines Integrierten Produkt- und Prozessmodells, das sich für den Anwendungsbereich der öffentlichen Verwaltung eignet und als Referenzmodell in einer Vielzahl von Verwaltungen eingesetzt werden kann.

Ziel des Prototypings ist die Entwicklung einer schnell verfügbaren, lauffähigen Vorabversion eines Anwendungssystems [WH06, S.10]. Zur Unterstützung der Modellierungsaktivitäten in der Praxis ist im Rahmen eines Prototypings auch die Entwicklung und Evaluation eines oEPK-Modellierungstools vorgesehen, das in gemeinsamen Projekten sowohl von Organisatoren als auch von IT-Vertretern eingesetzt werden kann.

Im diesem Beitrag wird ein methodischer Ansatz vorgestellt, der (1) das Modellierungskonzept der objektorientierten Ereignisgesteuerten Prozesskette (oEPK) mit (2) UML-

Modellierungskonzepten verknüpft, um auf dieser Basis die Grundlage für einen systematischen und prozesskonformen Aufbau eines (3) dienstbasierten Architekturkonzeptes legen zu können. Vergleichbare Arbeiten, welche diese drei Konzepte in einen integrativen Modellierungsansatz miteinander kombinieren sind bisher nicht publiziert.

2.3 Verwandte Arbeiten

Etablierte Ansätze zur objektorientierten Produkt- und Prozessmodellierung finden sich in der wissenschaftlichen Literatur primär zur Unified Modeling Language (UML) [OMG08]; [Oe06], zum Semantischen Objektmodell (SOM) [FS06]; [FS95], zu Objekt-Petrinetzen [Ba90]; [Ob96]; [ZH00]; [SL05]; [KLO08] und zur objektorientierten Ereignisgesteuerten Prozesskette (oEPK) [NZ98]; [SNZ97]. Ansätze zur Integration der Prozessorientierung in das objektorientierte Paradigma im Kontext Ereignisorientierter Prozessketten finden sich weiter in [Sc01]; [Zi98]; [ScSc97].

3 EU-Dienstleistungsrichtlinie

Die EU-Kommission versteht unter dem Begriff der Dienstleistung „jede selbständige wirtschaftliche Tätigkeit, die in der Regel gegen Entgelt erbracht wird“ [EU07, S. 11]. Von diesem auf die unternehmerische Tätigkeit bezogenen Dienstleistungsbegriff sind verwaltungsbezogene Tätigkeiten und Dienstleistungen abzugrenzen. Deren Zielsetzung ist es, die Unternehmensaktivitäten zu unterstützen, zu ordnen und damit die Aufnahme und Ausübung der unternehmerischen Tätigkeit zu gewährleisten. Hierzu zählen besonders Unterstützungsleistungen in Form der Bearbeitung von erforderlichen Genehmigungen und Registrierungen auf Basis von Rechtsnormen (wie Gesetze und Verordnungen).

Die EU-Dienstleistungsrichtlinie (EU-DLR) fordert die Mitgliedsstaaten auf, die erforderlichen Rechts- und Verwaltungsvorschriften in Kraft zu setzen, die notwendig sind, um den Zielsetzungen der Richtlinie bis Ende 2009 nachzukommen. Durch die Richtlinie soll der freie Dienstleistungsverkehr innerhalb der Gemeinschaft deutlich vereinfacht und erleichtert werden. Den Kern der Zielsetzungen bildet die Verwaltungsvereinfachung zugunsten von Unternehmen (Kapitel 2 der Richtlinie). Die Mitgliedsstaaten sind danach aufgefordert:

- die geltenden Verfahren und Formalitäten zur Aufnahme und Ausübung einer Dienstleistungstätigkeit auf ihre Einfachheit hin zu überprüfen und ggf. zu vereinfachen (Art. 5 - Verwaltungsvereinfachung),
- einheitliche Ansprechpartner einzurichten, über welche die Dienstleistungserbringer alle Verfahren und Formalitäten im Rahmen ihrer Dienstleistungstätigkeit abwickeln können (Art. 6 - einheitlicher Ansprechpartner),
- sicher zu stellen, dass alle Verfahren und Formalitäten problemlos aus der Ferne und elektronisch über den einheitlichen Ansprechpartner oder bei der zuständigen Behörde abgewickelt werden können (Art. 8 - elektronische Verfahrensabwicklung).

Profitieren von der Richtlinie sollen alle inländischen und ausländischen Unternehmen mit Firmensitz innerhalb der Europäischen Union [EU07, S. 23]. Dabei stehen sich die

Anforderungen der Unternehmen einerseits und die Konsequenzen für die öffentlichen Verwaltungen andererseits gegenüber [HKN08].

4 Rahmenkonzept

Prozesse in der öffentlichen Verwaltung haben bei abstrakter Betrachtung hinsichtlich ihrer Zielsetzung nur eine geringe strukturelle Varianz (i.d.R. Antragsbearbeitungen), lassen aber aufgrund der historisch gewachsenen Organisationsstrukturen, Verwaltungsvorschriften und Anwendungsfälle beträchtliche Unterschiede erkennen [SNZ97]. Diese Heterogenität ist neben Fragen des Datenschutzes und der politischen Willensbildung ein wesentlicher Grund, warum es hier bisher kaum integrierte Anwendungssysteme gibt, sondern Insellösungen für einzelne organisatorische Bereiche [Zi98, S. 183].

Die effektive Bereitstellung von Verwaltungsleistungen erfordert jedoch eine Verknüpfung der angebotenen Dienstleistungen und Dienstleistungsbündel (Produkte), der zur Verfahrensabwicklung eingesetzten Ressourcen (Prozesse) und der technischen Dienste und folglich eine Produkt- und Prozessarchitektur mit darauf ausgerichteten Anwendungssystemen und IT-Infrastruktur. Integrierte Produkt- und Prozessmodelle sind somit ein vielversprechender Ansatz, die Informationstechnik in der öffentlichen Verwaltung auf die neuen Anforderungen hin auszurichten. Durch die Integration von Produkten, Prozessen und Systemen sowie der Bereitstellung von wieder verwendbaren Diensten können wichtige Integrationspotenziale abgebildet und umgesetzt werden. Abbildung 1 zeigt die Komponenten eines Rahmenkonzeptes für ein Integriertes Produkt- und Prozessmodell, das für den Untersuchungsgegenstand der EU-Dienstleistungsrichtlinie geeignet ist.

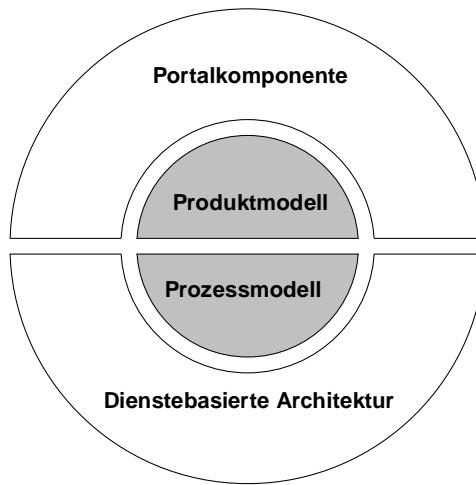


Abbildung 1: IPP-Rahmenkonzept zur Umsetzung der EU-DLR

Die wesentlichen Komponenten des IPP-Rahmenkonzeptes umfassen demnach:

- Portalkomponente: Die Umsetzung der EU-Dienstleistungsrichtlinie fordert, dass unternehmensbezogene Verwaltungsleistungen elektronisch verfügbar gemacht werden und auch „aus der Ferne“ in Anspruch genommen werden können. Um dies zu ermöglichen, stellt eine Portalkomponente im Frontoffice die unternehmensbezogenen Verwaltungsleistungen so bereit, dass über das Portal individuelle am Bedarf ausgerichtete Servicepakete durch die Unternehmen gebildet werden können.
- Produktmodell: Das Produktmodell umfasst alle unternehmensbezogenen Verwaltungsleistungen und bildet so die inhaltliche Basis für das Portalangebot. Es beschreibt die Gesamtheit der in der jeweiligen Organisation erbrachten Dienstleistungen für Unternehmen. Unter Produkt wird hierbei ein aus Kundensicht erkennbares Ergebnis eines Verwaltungsprozesses verstanden, wobei der Ressourcenaspekt zunächst ausgeklammert wird [BHL98].
- Prozessmodell: Jeder Produkterstellung liegt ein mehr oder minder arbeitsteiliger Geschäftsprozess zugrunde. Ein Geschäftsprozess wird dabei als eine ereignisgesteuerte Bearbeitung von Geschäftsobjekten mit dem Ziel der Produkterstellung verstanden. Auf der Grundlage von Modellen zu Ist-Prozessen können Schwachstellen analysiert und Soll-Prozessmodelle abgeleitet werden. Zudem sind Prozessmodelle im IPP-Rahmenkonzept der zentrale Ausgangspunkt für die Modellierung von Datenmodellen als Grundlage für die technische Realisierung elektronischer Verfahrensabwicklungen.
- Dienstbasierte Architektur: Eine dienstbasierte Architektur [Jo08; LDL07] bildet im Weiteren die technische Basis für den Einsatz von wieder verwendbaren, bausteinbasierten Services im Rahmen der Bereitstellung des Portalangebotes sowie der Teil-/Automatisierung der Prozesse. Dienstbasierte Architekturen (SOA) bilden auf diese Weise eine geeignete Plattform für die produkt- und prozessorientierte Umsetzung der EU-Dienstleistungsrichtlinie [MG05]. Somit können bereits vorhandene IT-Infrastrukturen und Softwarekomponenten integriert werden, ohne diese für die Ausrichtung auf die neuen Anforderungen komplett ersetzen zu müssen.

Das Rahmenkonzept folgt dabei konsequent dem objektorientierten Paradigma. Ein Objekt ist demnach definiert als „ein Ding (eine Entität) mit einer eindeutigen Identität. Es wird durch seinen Zustand und sein Verhalten charakterisiert. Der Zustand wird durch Attribute und Beziehungen repräsentiert, das Verhalten durch Operationen. Objekte, die auf gleiche Weise charakterisiert sind, gehören der gleichen Klasse an.“ [GI08].

Ein Geschäftsobjekt wird definiert als “a representation of a thing active in the business domain, including at least its business name and definition, attributes, behavior, relationships and constraints. A business object may represent, for example, a person, place or concept. The representation may be in a natural language, a modeling language, or a programming language” [OMG97]. Analoge Begriffsbestimmungen finden sich u.a. auch bei [Oe06; NZ98; CD91; RBP91]. Verwaltungsobjekte in diesem Sinne sind materielle Güter (wie Formulare oder Buchungsbelege), Personen (Antragsteller, Mitarbeiter) oder immaterielle Güter (wie Rechte, Genehmigungen oder Registrierungen).

4.1 Prozessmodell „Gewerbe-Anmeldung“

Bei dem Modellierungskonzept der objektorientierten Ereignisgesteuerten Prozessketten werden Workflows mit Ereignissen und Objekten modelliert, die alternierend mit gerichteten Kanten verbunden werden. Durch (AND-, OR- und XOR-) Konnektoren kann der Prozessfluss aufgespalten und wieder zusammengeführt werden. Abbildung 2 zeigt einen Modellierungsansatz auf Basis der in [SNZ97] eingeführten objektorientierten Ereignisgesteuerten Prozessketten. Die oEPK-Notation für das Geschäftsobjekt ist in dieser Abbildung eng an die Notation für UML-Klassen [OMG08] angelehnt, was die Vorbereitungen auf eine IT-bezogene Teil-/Automatisierung von Prozessen begünstigt.

Die „Gewerbe-Anmeldung“ wird hier als Geschäftsobjekt symbolisiert und beinhaltet alle zur Bearbeitung und Zustandsänderungen benötigen betriebswirtschaftlichen Attribute und Methoden. Das Geschäftsobjekt „Gewerbe-Anmeldung“ bildet so das „Kernobjekt“ im Prozess, das durch die Veränderung seines Zustandes den Prozessfortschritt bestimmt und lenkt. Dabei stellt nicht jede Änderung eines Attributwertes eine Zustandsänderung dar, sondern nur solche Ereignisse, die das Verhalten maßgeblich beeinflussen [Oe06, S. 319].

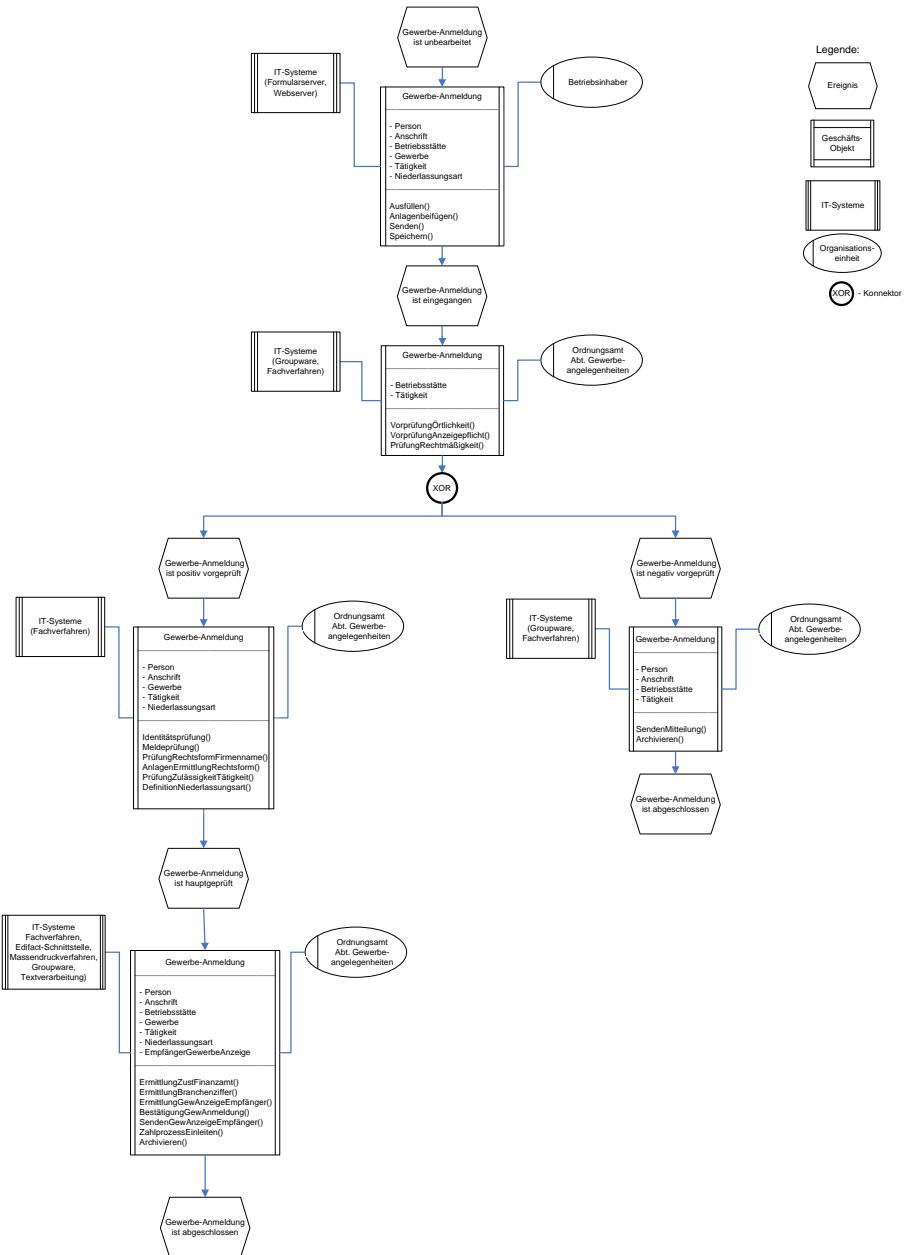


Abbildung 2: oEPK-Prozessmodell zur Gewerbe-Anmeldung

4.2 Klassendiagramm „Gewerbe-Anmeldung“

Als Basis für die technische Realisierung der elektronischen Verfahrensabwicklung im Sinne der EU-DLR werden im Weiteren die Geschäftsobjekte mit ihren betriebswirtschaftlichen Attributen und Methoden aus der oEPK in einem UML-Klassendiagramm weiter spezifiziert; Abbildung 3 zeigt das Klassendiagramm zur „Gewerbe-Anmeldung“. Die eindeutige Zuordnung von Methoden zu Attributen kann dabei je nach Prozesskomplexität und notwendiger Granularität sowohl durch Verfeinerung der oEPK, als auch in separaten Diagrammtypen erfolgen.

Die Klassen beinhalten neben den Attributen alle Operationen, die zur Bearbeitung und zur Zustandsänderung der „Gewerbe-Anmeldung“ erforderlich sind. Die hervorgehobenen Operationen werden im Ausgangsfall technisch nicht unterstützt, sondern erfolgen manuell durch die Sachbearbeitung. Gleichwohl sind diese Operationen ggf. in einer Vielzahl von Verwaltungsprozessen erforderlich. So kommt bspw.

- die Operation „ErmittlungBranchenziffer“ nicht nur im Prozess zur Gewerbe-Anmeldung, sondern auch bei der Gewerbe-Ummeldung zum Einsatz, zudem ist diese Operation in weiteren Verwaltungsprozessen relevant, bspw. in Prozessen der Wirtschaftsförderung oder
- die Operation „ErmittlungZustFinanzamt“ auch nicht nur im Prozess zur Gewerbe-Anmeldung zum Einsatz, sondern zudem bei der Gewerbe-Ummeldung sowie in weiteren Verwaltungsprozessen, bspw. in Prozessen zur Gewerbebesteuerung.

Mit Blick auf das IPP-Rahmenkonzept bilden die hervorgehobenen Operationen in Abbildung 3 die Basis zur Ermittlung neuer technischer Dienste, die als wieder verwendbare Bausteine in verschiedenen Prozessen genutzt werden können. Das gewählte Klassendiagramm bildet damit die methodische Basis zum Aufbau der dienstbasierten Architektur im Rahmen des IPP. Sofern das gewählte Modellierungstool -wie im vorliegenden Fall- auf Basis der Klassen auch bereits einen korrespondierenden Java-Code generiert, wird auf diese Weise auch eine Ausgangsbasis für die technische Realisierung der ausgewählten Dienste gelegt.

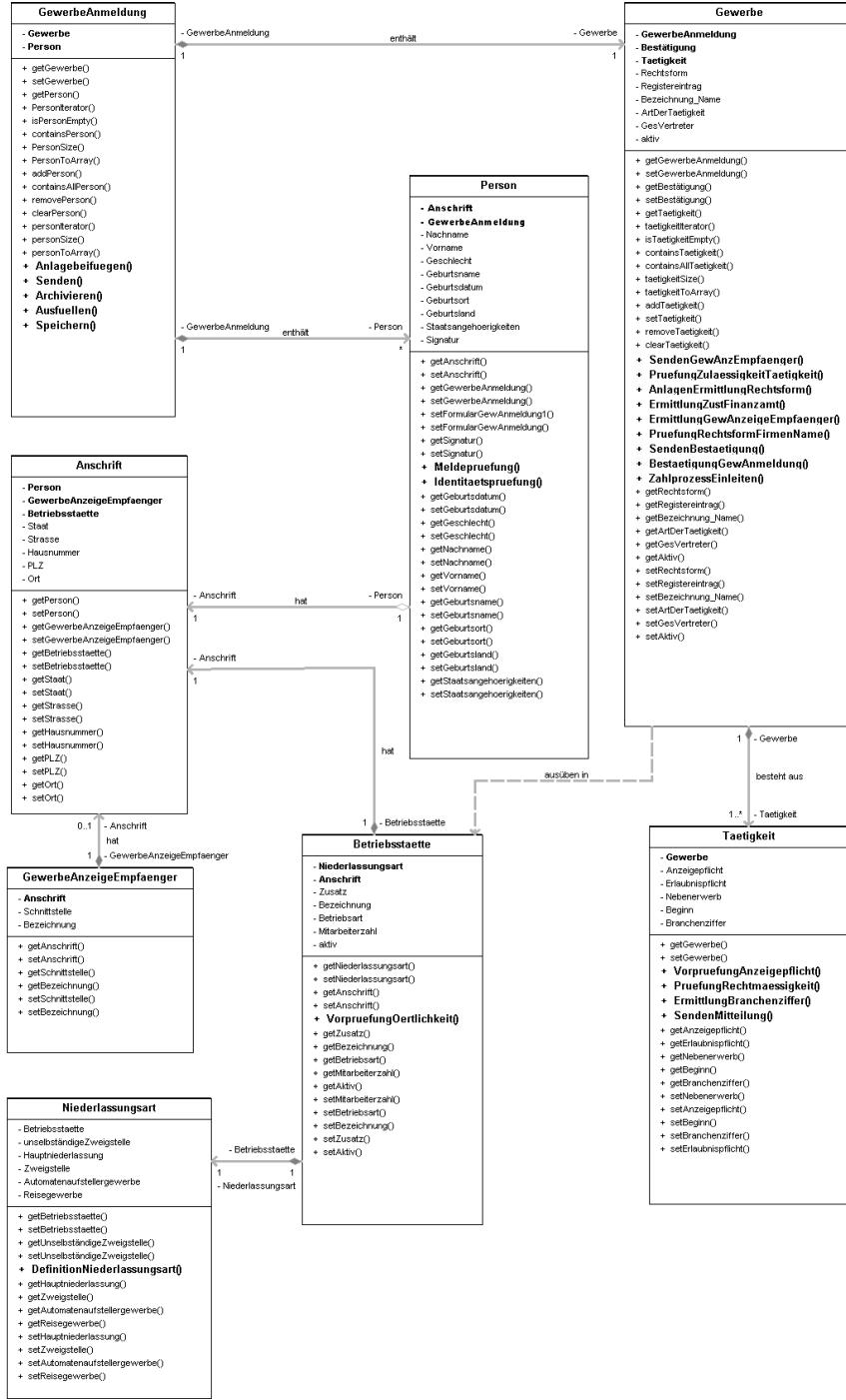


Abbildung 3: UML-Klassendiagramm zur Gewerbe-Anmeldung

5 Zusammenfassung und weiterer Forschungsbedarf

5.1 Zusammenfassung

Der Beitrag beschreibt ein Rahmenkonzept zur integrierten Produkt- und Prozessmodellierung im öffentlichen Sektor. Dieser ist traditionell durch (teil-)redundante papierbasierte Formulare geprägt. Grundvoraussetzung für die Digitalisierung, Bündelung und Virtualisierung der Dienstleistungen ist zunächst die Schaffung eines gemeinsamen Modellverständnisses bei allen Beteiligten (wie Organisatoren, Modellierern, Programmierern) sowie die Entwicklung von wiederverwendbaren Referenzprozessen und bausteinbasierten Diensten. Am Beispiel des Verwaltungsprozesses zur Gewerbe-Anmeldung wird das Modellierungskonzept der objektorientierten Ereignisgesteuerten Prozesskette (oEPK) mit UML-Modellierungskonzepten (Klassen, Klassendiagramm) so verknüpft, dass die Grundlage für einen systematischen und prozesskonformen Aufbau eines dienstbasierten Architekturkonzeptes gelegt werden kann.

5.2 Limitationen

In der wissenschaftlichen Literatur haben sich unterschiedliche Ansätze zur objektorientierten Produkt- und Prozessmodellierung etabliert (vgl. 2.3). Der dargestellte Modellierungsansatz stellt auf Basis der objektorientierten Ereignisgesteuerten Prozessketten (oEPK) Ergebnisse der laufenden Forschungsarbeit vor. In wieweit andere Ansätze für die Zielsetzung geeignet sind, bleibt weiteren Untersuchungen vorbehalten.

Zudem lässt die derzeitige Anzahl der untersuchten und mittels der oEPK modellierten Prozesse noch keine belastbaren Aussagen zur Handhabbarkeit des Modellierungswerkzeuges für den Praxiseinsatz zu. Dies ist in Folgeuntersuchungen weitere zu evaluieren.

5.3 Weiterer Forschungsbedarf

Auf Basis des derzeitigen Forschungsstandes ergeben sich weitere wissenschaftliche Fragestellungen, die im Rahmen zukünftiger Arbeiten zu beantworten sind:

- Ist das Modellierungskonzept der objektorientierten Ereignisgesteuerten Prozesskette (oEPK) ein geeignetes Beschreibungsmittel, zur Modellierung unternehmensbezogener Verwaltungsprozessen für den Anwendungsfall einer Großstadt?
- Ist das vorgestellte IPP-Rahmenkonzept und der darauf aufgebaute Modellierungsansatz auf andere Großstädte sowie auf kleine und mittlere Kommunen übertragbar?

Das in diesem Beitrag vorgestellte IPP-Rahmenkonzept und der gewählte Modellierungsansatz dienen derzeit als Grundlage zum Aufbau eines umfassenden EU-DLR-Referenzmodells. Am Anwendungsfall der Landeshauptstadt Düsseldorf, die im Rahmen einer wissenschaftlichen Begleitforschung bei ihrer Umsetzung der EU-Dienstleistungsrichtlinie unterstützt wird, wird die Forschungsarbeit weiter entwickelt und zudem auf kleine und mittlere Kommunen ausgeweitet. Neben der Handhabbarkeit steht jeweils auch der generierte Zusatznutzen für die Endanwender im Vordergrund. Die Validierung erfolgt in engem Dialog mit Wirtschaftsverbänden und Unternehmen im

Rahmen der pilotierten Einführung. Der Anwendungsfall ist ein Best-Practice-Beispiel [HKK08] zur Umsetzung der EU-Dienstleistungsrichtlinie im Rahmen der Initiative Deutschland-Online (Blaupause EU-DLR) [DOL08].

Literatur

- [Ba90] Baumgarten, B. (1990): Petri-Netze. Grundlagen und Anwendungen. BI-Wissenschaftsverlag, Mannheim.
- [Br00] Brandner, S. (2000): Produktdaten- und Prozeßmanagement in virtuellen Fabriken. Reinhardt, G. (Hg.): Forschungsberichte iwb, Band 136. München.
- [BBR08] Bundesamt für Bauwesen und Raumordnung (Hg.) (2008): Gemeinden/Gemeindeverbände in Deutschland. <http://www.raumbeobachtung.de/>, zuletzt geprüft am 10.10.08.
- [BHL98] Breitling, M.; Heckmann, M.; Luzius, M., Nüttgens, M. (1998): Service Engeneering in der Ministerialverwaltung. In: Information Management & Consulting (IM), 13 (1998) Sonderausgabe „Service Engineering“, S. 91-98.
- [BM06] Bundesministerium des Innern (Hg.) (2006): Der öffentliche Dienst in Deutschland (in German). http://www.bmi.bund.de/Internet/Content/Common/Anlagen/Broschueren/2006/Der_oeffentliche_Dienst_in_Deutschland_Id_21754_de,templateId=raw,property=publicationFile.pdf, zuletzt geprüft am 10.10.2008.
- [CD91] Coad, P.; Yourdon, E. (1991): Object-Oriented Analysis, Prentice-Hall, Upper Saddle River. Yourdon Press Computing Series. New Jersey.
- [Dau99] Daumenlang, K. (1999): Querschnitt- und Längsschnittmethoden. In: Roth, E.; Holling, H. (Hg.): Sozialwissenschaftliche Methoden: Lehr- und Handbuch für Forschung und Praxis, S. 309-326.
- [DOL08] Deutschland-Online Vorhaben "Dienstleistungsrichtlinie" (2008). http://www.deutschland-online.de/DOL_Internet/broker.jsp?uMen=58c105dd-ba3ea511-4fbf-1b1ac0c2f214, zuletzt geprüft am 10.10.2008.
- [Ei89] Eisenhardt, K.M. (1989). Building Theories from Case Study Research. In: Academy of Management Review, Volume 14, No. 4, 532-550.
- [ES05] Eversheim, W.; Schuh, G. (Hg.) (2005): Integrierte Produkt- und Prozessgestaltung. Berlin.
- [EU06a] Kommission der Europäischen Gemeinschaften (2006): E-Government-Aktionsplan im Rahmen der i2010-Initiative: Beschleunigte Einführung elektronischer Behördendienste in Europa zum Nutzen aller, vom 25.04.2006. In: Mitteilung der Europäischen Kommission an den Rat der Europäischen Gemeinschaften, Brüssel
- [EU06b] Europäisches Parlament und Europäischer Rat (2006): Richtlinie 2006/123/EG über Dienstleistungen im Binnenmarkt (EU-Dienstleistungsrichtlinie). 2006/123/EG, vom 12.12.2006. In: Amtsblatt der Europäischen Union, Brüssel.
- [EU07] Kommission der Europäischen Gemeinschaften (2007): Handbuch zur Umsetzung der EU-Dienstleistungsrichtlinie. Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften (Hg.). Luxemburg.
- [FS95] Ferstl O.K.; Sinz, E.J. (1995): Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: Wirtschaftsinformatik, Band 37, Heft 3. S. 209-220.
- [FS06] Ferstl O.K.; Sinz, E.J. (2006): Grundlagen der Wirtschaftsinformatik. Oldenburg.
- [GI08] Gesellschaft für Informatik e.V. (GI) (2008). Informatik-Begriffsnetz. <http://public.fth-berlin.de/~giak/>, zuletzt geprüft am 10.10.2008.

- [Ha02] Hars, A. (2002): Wissenschaftstheorie für Wirtschaftsinformatiker. Tutorial im Rahmen der Multikonferenz Wirtschaftsinformatik 2002. 9.-11. September 2002. Nürnberg.
- [HBN08] Hogrebe, F.; Blinn, N.; Nüttgens, M. (2008): Benchmarkstudie „Kommunale Online-Dienstleistungen“: Eine Analyse unternehmensbezogener eGovernment-Angebote deutscher Kommunen, in Nüttgens, M. (Hg): Arbeitsberichte zur Wirtschaftsinformatik der Universität Hamburg Nr.3/2008, Hamburg.
- [He95] Heinrich, L. J. (1995): Forschungsziele und Forschungsmethoden der Wirtschaftsinformatik. In: Wächter, H. (Hg.): Selbstverständnis betriebswirtschaftlicher Forschung und Lehre. Gabler Verlag, Wiesbaden 1995, 27 – 54.
- [HKK08] Hogrebe, F.; Kruse, W.; van Kempen, B.; Nüttgens, M. (2008): Die Landeshauptstadt Düsseldorf auf dem Weg zur Umsetzung der EU-Dienstleistungsrichtlinie: Integriertes Produkt- und Prozessmodell für dienstbasierte Anwendungen und Architekturen, in Nüttgens, M. (Hg): Arbeitsberichte zur Wirtschaftsinformatik der Universität Hamburg Nr. 2 / Juli 2008, Hamburg.
- [HKN08] Hogrebe, F.; Kruse, W.; Nüttgens, M. (2008): One-Stop-eGovernment für Unternehmen: Ein Bezugsrahmen zur Virtualisierung und Bündelung öffentlicher Dienstleistungen am Beispiel der Landeshauptstadt Düsseldorf. Tagungsband, Multikonferenz Wirtschaftsinformatik 2008, S. 353-364. München.
- [HN08] Hogrebe, F.; Nüttgens, M. (2008): Integriertes Produkt- und Prozessmodell für dienstbasierte Anwendungen und Architekturen am Beispiel der EU-Dienstleistungsrichtlinie. In: Hegering, H.-G.; Lehmann, A.; Ohlbach, H.J.; Scheideler, C. (Hg.): Informatik 2008 - Beherrschbare Systeme – dank Informatik, Band 1, S.87-90. GI-Edition. Lecture Notes in Informatics (LNI). 38. Jahrestagung der Gesellschaft für Informatik, 8-13.09.08, Workshop SOA und EU-Dienstleistungsrichtlinie in der öffentl. Verwaltung, München.
- [Jo08] Josuttis, N. (2008): SOA in der Praxis. System-Design für verteilte Geschäftsprozesse. Heidelberg.
- [Kl03] Klabunde, S. (2003): Wissensmanagement in der integrierten Produkt- und Prozessgestaltung - Best-Practice-Modelle zum Management von Meta-Wissen. Dissertation. Scheer, A.-W. (Hg.). Deutscher Universitäts-Verlag (Schriften zur EDV-orientierten Betriebswirtschaft). Saarbrücken.
- [KLO08] Klink, S.; Li, Y.; Oberweis, A. (2008): Income 2010 - a Toolset for Developing Process-Oriented Information Systems Based on Petri Nets. In: Proceedings of International Workshop on Petri Nets Tools and Applications (PNTAP 2008, associated to SIMU-Tools 2008). ACM digital library, Marseille.
- [Kö96] König, W.; Heinzl, A.; Rumpf, M.; Poblotzki von, A. (1996): Zur Entwicklung der Forschungsmethoden und Theoriekerne der Wirtschaftsinformatik in den nächsten zehn Jahren. Eine kombinierte Delphi- und AHP-Untersuchung, in: Heilmann, H. (Hg.): Information Engineering, München u.a., S. 35-66.
- [LDL07] Leyking, K.; Dreifus, F.; Loos, P. (2007): Serviceorientierte Architekturen. In: Wirtschaftsinformatik, Band 49, Heft 5. S. 394-402.
- [MG05] Moitra, D.; Ganesh, J. (2005): Web services and flexible business processes: towards the adaptive enterprise. In: Information and Management, vol. 42, no. 7, S. 921-933.
- [NZ98] Nüttgens, M.; Zimmermann, V. (1998): Geschäftsprozeßmodellierung mit der objekt-orientierten Ereignisgesteuerten Prozeßkette (oEPK). In: Maicher, M.; Scheruhn, H.-J (Hg.): Informationsmodellierung - Branchen, Software- und Vorgehensreferenzmodelle und Werkzeuge, S. 23-36. Wiesbaden.
- [Ob96] Oberweis, A. (1996): Modellierung und Ausführung von Workflows mit Petri-Netzen. Teubner-Reihe Wirtschaftsinformatik, B.G. Teubner Verlag.
- [Oe06] Österreich, B. (2006): Analyse und Design mit UML 2.1 – Objektorientierte Softwareentwicklung. München.
- [OMG97] Object Management Group (1997). Business objects definition.
<ftp://ftp.omg.org/pub/docs/enduser/97-04-01.pdf>, zuletzt geprüft am 10.10.2008.

- [OMG08] Object Management Group (2008). Unified Modeling Language 2.0.
<http://www.uml.org/>, zuletzt besucht am 10.10.2008.
- [RBP91] Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorenzon, W. (1991): Object-Oriented Modelling and Design. Englewood Cliffs.
- [ScSc97] Schwegmann, A.; Schlagheck, B. (1997): Integration der Prozessorientierung in das objektorientierte Paradigma: Klassenzuordnungsansatz vs. Prozessklassenansatz. In: Becker, J.; Grob, H.L.; Klein, St. (Hg.): Arbeitsberichte des Instituts für Wirtschaftsinformatik. Arbeitsbericht Nr. 60. Institut für Wirtschaftsinformatik. Universität Münster.
- [Sc01] Scheer, A.W. (2001): ARIS-Modellierungsmethoden, Metamodelle, Anwendungen. 4. Aufl., Springer. Berlin et al.
- [SL05] Sarshar, K.; Loos, P. (2005): Modellierung überbetrieblicher Behandlungsprozesse durch Objekt-Petrinetze. In: Wirtschaftsinformatik, Band 47, Heft 3. S. 203-210.
- [SNZ97] Scheer, A.W.; Nüttgens, M.; Zimmermann, V. (1997): Objektorientierte Ereignissteuerte Prozeßketten (oEPK) – Methode und Anwendung. Institut für Wirtschaftsinformatik, Heft 141, Universität des Saarlandes. Saarbrücken.
- [We04] Wegehaupt, P. (2004): Führung von Produktionsnetzwerken. Rheinisch-Westfälische Technische Hochschule Aachen. Dissertation. http://www.werkzeugbau-aachen.de/de/563d2e5c6b82304ac12570340024caa9/Patrick_2004.pdf , zuletzt geprüft am 10.10.2008.
- [WH06] Wilde, T.; Hess, T. (2006): Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung. Institut für Wirtschaftsinformatik und Neue Medien. Hess, T. (Hg.): Arbeitsbericht Nr. 2/2006. Universität München.
- [WH07] Wilde, T.; Hess, T. (2007): Forschungsmethoden der Wirtschaftsinformatik. Eine empirische Untersuchung. In: Wirtschaftsinformatik, Band 49, Heft 4. S. 280-287.
- [ZH00] Zapf, M.; Heinzl, A. (2000): Ansätze zur Integration und Petri-Netzen und objektorientierten Konzepten. In: Wirtschaftsinformatik, Band 42, Heft 1. S. 36-46.
- [Zi98] Zimmermann, V. (1998): Objektorientiertes Geschäftsprozessmanagement. Integrationsansatz - Modellierungsmethode - Anwendungsbeispiel. Dissertation. Scheer, A.-W. (Hg.). Deutscher Universitäts-Verlag (Schriften zur EDV-orientierten Betriebswirtschaft). Saarbrücken.

Komponentenorientierte betriebliche Anwendungssysteme

Zur systematischen Identifikation von Services: Kriterien, aktuelle Ansätze und Klassifikation

Dominik Birkmeier, Sebastian Klöckner und Sven Overhage

Forschungsgruppe Component und Service Engineering,
Lehrstuhl für Wirtschaftsinformatik und Systems Engineering,
Universität Augsburg, Universitätsstraße 16, 86159 Augsburg
{dominik.birkmeier | sebastian.kloeckner | sven.overhage}@wiwi.uni-augsburg.de

Abstract: Die Einführung serviceorientierter Architekturen verspricht mit ihrem modularen Ansatz eine Vielzahl von Vorteilen für die betriebliche Anwendungsentwicklung. Eine erfolgreiche Einführung dieses Architekturkonzepts hängt jedoch entscheidend von einer ausreichenden methodischen Unterstützung des serviceorientierten Paradigmas ab. Derzeit steht insbesondere die Entwicklung systematischer Methoden für die Identifikation von Services im Mittelpunkt des wissenschaftlichen Interesses, was die Vielzahl veröffentlichter einschlägiger Ansätze verdeutlicht. Die in der Literatur zu findenden Ansätze weisen hinsichtlich ihrer Konzeption und Vorgehensweise allerdings eine starke Heterogenität auf. In diesem Beitrag werden daher grundlegende Kriterien zur Einordnung der verschiedenen Ansätze entwickelt und vorhandene Ansätze anhand eines detaillierten Klassifikationsschemas einander gegenübergestellt. Neben dem aktuellen Stand der Technik und den Unterschieden der einzelnen Ansätze werden dabei vor allem noch vorhandene Schwächen identifiziert, aus denen sich der weitere Forschungsbedarf ableiten lässt.

1 Motivation

An die betriebliche Anwendungsentwicklung wird heute eine Vielzahl von Ansprüchen gestellt: So zählen zur Zeit vor allem die Beherrschung der hohen *Anwendungskomplexität*, die Möglichkeit zur *flexiblen Anpassung* von Informationssystemen an Änderungen im Geschäftsumfeld, sowie die schnelle *Realisierung neuer Funktionalität* auf Basis der bestehenden Anwendungssysteme zu den zentralen Herausforderungen, die es mit geeigneten Konzepten zu bewältigen gilt [Bro00, CD01, PTD⁺06]. Als Architekturansatz, der mit seinem modularen Paradigma einen wichtigen Beitrag zur Lösung der genannten Herausforderungen verspricht, wird derzeit vor allem die Einführung *serviceorientierter Architekturen* (SOA) diskutiert [KL04, PTD⁺06].

Die erfolgreiche Anwendung des serviceorientierten Paradigmas in der Praxis hängt allerdings entscheidend davon ab, dass der Entwicklungsprozess ausreichend *methodisch unterstützt* wird [MDW02, PTD⁺06]. Neben der Frage, wie Services zu beschreiben, in Katalogen aufzufinden und (nach den Vorgaben eines Geschäftsprozesses) zu kombinieren sind, steht dabei vor allem die Entwicklung von Methoden und Praktiken, mit denen

sich informationstechnisch umzusetzende Services *systematisch identifizieren* lassen, im Mittelpunkt des wissenschaftlichen Interesses [Ars04, EAK06].

Die Identifikation geeigneter Services steht am Beginn des serviceorientierten Entwicklungsprozesses und stellt die Grundlage für alle weiteren Entwicklungsschritte sowie die anschließende Phase der Komposition bzw. Nutzung dar [Erl05]. Ihr kommt deshalb eine zentrale Bedeutung für den gesamten Prozess zu, die sich auch in einer Vielzahl bereits veröffentlichter Ansätze hinsichtlich der dabei einzuschlagenden Vorgehensweise widerspiegelt. Diese Ansätze weisen allerdings eine *starke Heterogenität* auf. So reichen sie bspw. von allgemeinen Hinweisen, die bei der Identifikation von Services zu berücksichtigen sind, über umgangssprachlich dargestellte bewährte Praktiken bis hin zu algorithmisch spezifizierten Vorgehensweisen. Ferner liegen den Ansätzen abweichende Servicedefinitionen und Vorgehensweisen zugrunde, wodurch sie zu deutlich abweichenden Ergebnissen gelangen können. Ein bestimmter Ansatz hat sich dabei schon aufgrund der relativ kurzen Zeit, seit der die Identifikation von Services untersucht wird, noch nicht durchsetzen können. Zudem fehlen *vergleichende Betrachtungen*, die die entstandenen Ansätze gegenüberstellen und das Forschungsgebiet auf diese Weise strukturieren. Durch eine Kategorisierung lassen sich zum einen komplementäre, einander ergänzende Ansätze identifizieren oder bislang noch unbehandelte Forschungsfragen ableiten, aus denen sich weiterer Forschungsbedarf ergibt. Zum anderen lässt eine Kategorisierung Rückschlüsse auf die Anwendbarkeit einzelner Ansätze (z.B. in verschiedenen Entwicklungskontexten) zu.

In diesem Beitrag werden vorhandene Ansätze, die zu einer systematischen Identifikation von Services beitragen sollen, gegenübergestellt und anhand eines differenzierten Schemas eingeordnet. Dazu werden in Kapitel 2 *grundlegende Kriterien* benannt, die für Ansätze zur Identifikation von Services charakteristisch sind. Diese bilden einen *Klassifikationsrahmen* für die Gegenüberstellung und Bewertung der einzelnen Ansätze. In Kapitel 3 werden dann Ansätze aus der Literatur vorgestellt und in das Klassifikationsschema eingeordnet. Als zentrale Forschungsmethode liegen diesem Beitrag Literaturoauswertungen zugrunde, die zur Benennung der Vergleichskriterien und zur Zusammenstellung der Serviceidentifikationsansätze durchgeführt wurden. Im Rahmen des Vergleichs werden die *Unterschiede* zwischen den einzelnen Ansätzen verdeutlicht und *vorhandene Schwächen* argumentativ-deduktiv abgeleitet¹. Nachdem in Kapitel 4 auf verwandte Arbeiten eingegangen wird, schließt Kapitel 5 mit einem Ausblick auf den *verbleibenden Forschungsbedarf*, der sich aus der Diskussion der vorgestellten Ansätze ergibt.

2 Kriterien für die Identifikation von Services

Die in der Literatur vorgeschlagenen Ansätze zur Identifikation von Services unterscheiden sich zum Teil deutlich hinsichtlich ihrer jeweiligen Konzeption und Vorgehensweise voneinander. Zum Vergleich und zur Einordnung der verschiedenen Ansätze lässt sich eine Reihe *grundlegender Kriterien* benennen, die einen detaillierten Aufschluss über dessen jeweilige Gestaltung geben. Die grundlegenden Kriterien wurden in Anlehnung an die Me-

¹Zur Diskussion von Forschungsmethoden der Wirtschaftsinformatik vgl. [WH07].

thodenforschung der Ingenieursdisziplinen bzw. der (Wirtschafts-) Informatik aufgestellt [PBFG03, BS04], die sich mit der Konzeption von systematischen Vorgehensweisen für die Entwicklung bzw. Konstruktion beschäftigt. Sie beschreiben

- die *konzeptionellen Grundlagen*, von denen der jeweilige Ansatz Gebrauch macht,
- das *allgemeine Vorgehen*, das mit dem Ansatz vorgeschlagen wird,
- die *Modellbildung*, die zur Identifikation von Services vorgenommen wird sowie
- Maßnahmen, die die *Anwendung* des Ansatzes unterstützen.

Die im Folgenden näher dargestellten Kriterien sollen dabei vor allem Auskunft darüber geben, inwiefern ein Ansatz tatsächlich geeignet ist, zur gewünschten *systematischen Identifikation* von Services beizutragen und wo ggf. Schwächen bestehen. Dabei ist anzumerken, dass die genannten Kriterien keinen Anspruch auf Vollständigkeit erheben. Jedoch haben sie sich als charakteristisch für eine systematisch-methodische Vorgehensweise herausgestellt [PBFG03, BS04].

2.1 Konzeptionelle Grundlagen

Die konzeptionellen Grundlagen beschreiben das begriffliche Verständnis, das dem jeweiligen Ansatz zur Serviceidentifikation zugrunde liegt. Dabei ist zunächst zu betrachten, welche Servicedefinition zur Identifikation verwendet wird. Ferner ist zu unterscheiden, wie exakt die jeweilige Vorgehensweise beschrieben wird und ob sie ggf. in ein übergeordnetes Vorgehensmodell mit Bezug zu den angrenzenden Entwicklungsphasen eingebettet ist. Im Einzelnen ergeben sich somit folgende Kriterien:

Servicedefinition. Betrachtet man die in der Literatur vorhandenen Servicedefinitionen, zeigt sich, dass die Autoren den Begriff *Service* sehr unterschiedlich fassen. Die verschiedenen Fassungen reichen dabei in einer ersten Näherung von technisch geprägten Auslegungen des Begriffs Service im Sinne von Web-Services als Software-Komponenten [SGM02, Nat03, Ars04, MSJL06] bis hin zu rein betriebswirtschaftlich geprägten Definitionen im Sinne von Dienstleistungen [BS06]. Technisch orientierte Definitionen legen den Fokus zumeist auf die technische Spezifikation und Implementierung von Services als softwaretechnisch umgesetzte Artefakte, die eine festgelegte Funktionalität anbieten. Im Mittelpunkt dieser Definitionen steht dabei die lose Kopplung von wiederverwendbaren und plattformunabhängigen, durch wohldefinierte Schnittstellen beschriebenen Services.

Stärker fachlich ausgerichtete Autoren beschreiben Services als eine zusammengehörende Menge von vermarktbaren Diensten, welche über in einer standardisierten Sprache verfasste Schnittstellenbeschreibungen verfügen [OT07]. Ein Dienst wird hier als eine Aktion eines (betrieblichen) Anwendungssystems [verstanden], welche die Bewältigung einer bestimmten Menge von (betrieblichen) Aufgaben unterstützt [Tur03]. Daneben hat mittlerweile auch eine betriebswirtschaftliche Interpretation von Services als Dienstleistung

in die Literatur Eingang gefunden [BS06]. Diese neue Definition der Service Science, die deutlich über den Fokus der hier diskutierten Serviceorientierten Architekturen hinausgeht, wird allerdings nicht weiter betrachtet. Das divergierende Verständnis des Begriffs *Service* spiegelt sich, schon wegen der unterschiedlichen Ausrichtung und den sich daraus ergebenden unterschiedlichen Zielsetzungen, auch in den darauf aufbauenden Methoden zur Identifikation von Services wider. Deshalb werden im Folgenden zunächst Ansätze mit eher fachlichem Fokus von solchen mit starker technischem Bezug unterschieden. Eine genauere Analyse, z.B. auf Basis einer Gegenüberstellung der jeweils zugrundeliegenden Servicedefinitionen, wäre darüber hinaus erstrebenswert. Jedoch ist festzustellen, dass zahlreiche Ansätze ohne Servicedefinitionen auskommen und allenfalls ein implizites Begriffsverständnis zugrundelegen. Dies erschwert nicht nur den angestrebten Vergleich, sondern zugleich auch die Anwendung der jeweiligen Ansätze in der Praxis.

Formalisierungsgrad. Auch der Formalisierungsgrad der verschiedenen Ansätze zur Serviceidentifikation variiert bisweilen deutlich. Er reicht prinzipiell von sog. ad-hoc Strategien, die Services allerdings nur mit einer unscharf beschriebenen opportunistischen Vorgehensweise identifizieren, über kodifizierte allgemeine Ratschläge bis hin zu strukturierten Vorgehensweisen und algorithmisch beschriebenen Methoden. Während die den ad-hoc Strategien zuzurechnenden Ansätze allenfalls sehr grob dargestellte Verfahren zur Serviceidentifikation skizzieren, geben allgemeine Ratschläge zumindest generalisierte Hinweise darauf, was bei der Identifikation von Services zu berücksichtigen ist. Strukturierte Vorgehensweisen beinhalten hingegen sowohl detailliert vorgegebene und beschriebene Arbeitsschritte, die für eine systematische Identifikation von Services zu durchlaufen sind, sowie klar spezifizierte Identifikations- und Selektionskriterien. Algorithmisch beschriebene Methoden verketten diese Arbeitsschritte zu einem klar vorgegebenen Identifikationsprozess, der ggf. von Werkzeugen unterstützt oder sogar automatisiert werden kann.

Übergeordnetes Vorgehensmodell. Ansätze für die Identifikation von Services werden üblicherweise im Rahmen einer umfassenden Entwicklungsmethodik eingesetzt, die den Entwicklungsprozess insgesamt steuert und begleitet. Die Integration der einzelnen Arbeitsschritte erfolgt dabei üblicherweise durch ein übergeordnetes Vorgehensmodell, das den Entwicklungsprozess strukturiert und die Weiterverwendung der in den einzelnen Phasen erreichten Teilergebnisse in einer abgestimmten Weise regelt [CD01, DW99, Sch98]. Verzichten einzelne Ansätze auf die Integration in ein übergeordnetes Vorgehensmodell und die Abstimmung mit den Erfordernissen späterer Entwicklungsphasen, sind deren Ergebnisse dort unter Umständen nur eingeschränkt weiterverwendbar.

2.2 Allgemeines Vorgehen

Das allgemeine Vorgehen beschreibt die grundsätzliche Strategie, die von einem Ansatz zur Identifikation von Services verfolgt wird. Dabei sind folgende Aspekte zu betrachten:

Ableitungsrichtung. Die Ableitungsrichtung beschreibt die Richtung der Analyse bei der Identifikation. Sog. *Top-Down*-Ansätze [Mil71] zur Serviceidentifikation betrachten dabei zunächst die Anwendungsdomäne und leiten Services aus den erstellten fachkonzeptionellen Modellen ab. In einem zweiten Schritt werden diese informationstechnisch spezifiziert und dann ggf. auf die vorhandene Anwendungslandschaft abgebildet. *Bottom-Up*-Ansätze gehen dagegen von der existierenden Anwendungslandschaft aus und modularisieren diese zunächst nach technischen Gesichtspunkten. In einem zweiten Schritt werden die identifizierten Module dann mit einer fachkonzeptionellen Bedeutung versehen und als Services bereitgestellt.

Da sowohl Top-Down- als auch Bottom-Up-Ansätze in einer jeweils idealtypischen Form die Gefahr von Fehlentwicklungen bergen (bspw. die redundante Realisierung von Funktionalität als Bestandteil verschiedener Services oder die Realisierung von aus fachlicher Sicht nicht eigenständigen Services), kombinieren einige Ansätze beide Analyserichtungen. Diese werden im Folgenden als *Meet-In-The-Middle*-Ansätze bezeichnet.

Optimierendes Verfahren. Die Identifikation von Services kann zugleich als mathematisches Optimierungsproblem aufgefasst werden. Wird bspw. nach dem etablierten Ansatz verfahren, zusammengehörige Funktionalität möglichst zu bündeln und so die Kommunikation zwischen Services zu minimieren [Par72], entsteht eine Aufgabenstellung, die etwa durch *Clustering*-Verfahren gelöst werden kann. Von den nicht-optimierenden sind deshalb optimierende Vorgehensweisen abzugrenzen, die ihrerseits wiederum in exakte und heuristische Verfahren unterteilt werden können.

2.3 Modellbildung

Zur Identifikation von Services werden jeweils konzeptionelle Modelle als Abbild der Realität herangezogen. Die verschiedenen Ansätze unterscheiden sich vor allem im Hinblick auf die Komplexität und den Inhalt der herangezogenen Modelle. Theoretische Grundlage eines methodischen Vorgehens bei der Serviceidentifikation sollten jedoch zumindest implizit die Konzepte der *Systemtheorie*, die auch auf (informations-) technische Systeme anzuwenden sind [Rop99], sowie die der *Konstruktionslehre* [PBFG03] sein. Dementsprechend sind folgende Kriterien zu betrachten:

Einbindung verschiedener Modellsichten. Die Systemtheorie schlägt zur Konzeption und Beschreibung (informations-) technischer Systeme drei Modellsichten vor, die auch in Modellierungsansätzen der (Wirtschafts-) Informatik verbreitet sind [Rop99, Sch98, DW99, Dav93]. Die *statische Sicht* (auch Daten-, Typ- oder Informationssicht) beschreibt die Beschaffenheit wesentlicher Systemattribute und deren Struktur. Die *operationale Sicht* (auch Funktionssicht) beschreibt das beobachtbare Systemverhalten und verknüpft dabei Systemattribute zu Ein- und Ausgaben. Darüber hinaus wird eine funktionale Dekomposition durchgeführt, die den Zusammenhang zwischen komplexen Funktionen und ihren Teilfunktionen beschreibt. Die *dynamische Sicht* (auch Prozesssicht) beschreibt schließ-

lich die zeitliche Abfolge von Systemfunktionen.

Für die Identifikation von Services sind grundsätzlich alle Modellsichten heranzuziehen, schon da sie erst in der Zusammenschau eine umfassende Systemsicht ergeben [Rop99]. Von den verschiedenen Ansätzen werden jedoch verschiedene Teilmengen der hier genannten Modellsichten genutzt, woraus sich spezifische Vor- und Nachteile ergeben.

Berücksichtigung existierender Systemstrukturen. Die Identifikation von Services erfolgt häufig in einer bereits existierenden Systemumgebung, die Einfluss auf die einzuschlagende Vorgehensweise haben kann. So sind ggf. bereits existierende Services oder zu modularisierende Legacy-Anwendungen einzubinden [DW99]. In der Literatur wird dieser Umstand bislang nicht einheitlich behandelt.

Berücksichtigung von Abhängigkeiten zum Umsystem. Services ergeben in der Regel erst im Zusammenspiel mit anderen Services die benötigte Funktionalität. Sie werden also entwickelt, um mit anderen Services verbunden zu werden [SGM02]. Häufig wird deshalb zugleich eine Identifikation mehrerer, aufeinander abgestimmter Services in einem umfassenderen Ansatz durchgeführt, der bspw. danach strebt, zusammengehörige Teilfunktionen möglichst einem Service zuzuordnen und so bspw. Abhängigkeiten zwischen verschiedenen Services zu minimieren. Weiterhin bestehende Abhängigkeiten eines Services zum Umsystem lassen sich mit einem solchen Ansatz zudem exakt spezifizieren. Andere Ansätze betrachten dagegen jeweils nur einen Service und lassen bestehende Abhängigkeiten ggf. vollständig außer Acht.

Unterscheidung von Servicehierarchien. Bei der Identifikation lassen sich prinzipiell zusammengesetzte Services, die aus der Komposition anderer, ggf. bereits bestehender Services entstehen, und elementare Services, die nicht weiter in Services unterteilt werden, unterscheiden. Ansätze, die eine solche Unterscheidung ermöglichen, berücksichtigen das sog. hierarchische Systemkonzept der Systemtheorie, das eine *schrittweise Verfeinerung* identifizierter Services bis hin zu elementaren Einheiten erlaubt [Rop99]. Andere Ansätze gehen dagegen davon aus, dass eine solche schrittweise Verfeinerung (bspw. schon wegen der grundsätzlich zu verbergenden Innensicht von Services) nicht notwendig ist und verzichten auf eine entsprechende Differenzierung. Zur weiteren Verfeinerung von Services sind solche Ansätze ggf. wiederholt anzuwenden.

Unterscheidung von Servicetypen. Schließlich können bei Serviceorientierten Architekturen grundsätzlich Teile unterschieden werden, deren primäre Aufgabe entweder die *Verwaltung von Informationen* (Entity Service) oder die *Bearbeitung von Aufgaben* (Task Service) ist [CD01, HS00]. Einige Ansätze berücksichtigen dies schon bei der Identifikation von Services und erreichen auf diese Weise eine Trennung zwischen daten- und funktionsbezogenen Diensten. Es ist jedoch umstritten, ob eine solche Trennung zu einem optimalen Ergebnis führt. So verlangt Parnas bspw. eine Gruppierung von Daten und darauf arbeitenden Funktionen in einem Modul [Par72]. In verschiedenen Ansätzen wird die Unterscheidung von Entity und Task Services deshalb nicht explizit getroffen.

2.4 Anwendung

Die Maßnahmen zur Unterstützung der Anwendung eines Ansatzes geben Aufschluss darüber, wie effizient sich dieser in der Praxis einsetzen lässt. Neben einer nachvollziehbar beschriebenen Vorgehensweise und einer adäquaten Modellbildung sind dafür vor allem folgende Kriterien ausschlaggebend:

Werkzeugunterstützung. Die Anwendbarkeit der verschiedenen Ansätze in der Praxis wird durch die Verfügbarkeit von entsprechenden Werkzeugen erleichtert, die die vorhandene Komplexität für den Entwickler besser handhabbar machen und ihn durch die notwendigen Schritte zur Serviceidentifikation leiten. Das Fehlen entsprechender Werkzeuge behindert insbesondere eine ggf. mögliche Optimierung des Ergebnisses bei der Identifikation von Services.

Qualitätsaussage. Die Qualität des Ergebnisses einer durchgeführten Serviceidentifikation ist für die praktische Anwendbarkeit eines propagierten Ansatzes von grundlegender Bedeutung, da die Realisierung der geplanten SOA mit nicht zu vernachlässigenden strategischen, aber auch finanziellen Konsequenzen verbunden ist. Im Idealfall kann die verwendete Identifikationsmethode die Korrektheit eines Ergebnisses, insbesondere die Vermeidung lokaler Optima bei einer ggf. durchgeführten Optimierung, garantieren. Dies ist vor allem bei der Verwendung algorithmisch beschriebener Methoden zu fordern. Kommt ein heuristisches Verfahren zum Einsatz, das aufgrund methodeninhärer Einschränkungen eine solche Garantie nicht geben kann, bieten sich andere Möglichkeiten die Lösungsqualität zu analysieren. Oftmals ist es möglich die maximale Abweichung von einem globalen Optimum theoretisch zu bestimmen (vgl. hierzu bspw. [Jun07]). Weiterhin bietet eine Sensitivitätsanalyse dem Anwender die Möglichkeit zur Überprüfung der Empfindlichkeit einer erreichten Lösung hinsichtlich der Variation von Inputparametern. Zahlreiche Ansätze verzichten jedoch gänzlich auf eine Evaluation des Identifikationsergebnisses.

Validierung. Die Validierung eines Serviceidentifikationsansatzes erlaubt weitgehende Rückschlüsse auf die Korrektheit, Verwendbarkeit und Erprobung der dargestellten Vorgehensweisen. Während eine Plausibilitätsprüfung zwar die grundsätzliche Korrektheit eines Vorgehensmodells demonstriert, zeigt erst die Betrachtung von Anwendungsfällen die notwendigen Nutzungsbedingungen sowie potentielle Einsatzmöglichkeiten und -grenzen auf. Im Idealfall kann der präsentierte Identifikationsansatz bereits durch vielfache praktische Anwendung und daraus resultierende „Best Practices“, die u.a. den Einsatz weiter operationalisieren, bestätigt werden.

2.5 Klassifikationsschema

Die zuvor beschriebenen Kriterien lassen sich zu einem Klassifikationsschema zusammenfassen, dass in Tabelle 1 dargestellt ist. Die Ausprägungen der einzelnen Kriterien

wurden dabei zu einem morphologischen Kasten zusammengefasst. Das Klassifikations-schema dient als Grundlage zur Einordnung der verschiedenen Ansätze zur Identifikation von Services, die im nächsten Kapitel vorgestellt werden.

Klassifikationsrahmen für Methoden zur Serviceidentifikation				
Zugrundeliegende Servicedefinition	Keine	Technisch	Fachlich	
Formalisierungsgrad	Ad-Hoc	Allgemeine Ratschläge	Strukturiert	Algorithmus
Übergeordnetes Vorgehensmodell	Ja		Nein	
Ableitungsrichtung	Top Down	Bottom Up		Meet In The Middle
Optimierendes Verfahren	Ja (Exakt)	Ja (Heuristisch)		Nein
Modellsichten	Daten	Funktionen		Prozesse
Berücksichtigung existierender Systemstrukturen	Gegeben		Nicht gegeben	
Berücksichtigung von Umsystemabhängigkeiten	Ja		Nein	
Unterscheidung von Servicehierarchien	Ja		Nein	
Unterscheidung von Servicetypen	Ja		Nein	
Werkzeugunterstützung	Gegeben		Nicht gegeben	
Qualitätsaussage	Keine	Sensitivitätsanalyse		Qualitätsgarantie
Validierung	Keine	Plausibilitäts-prüfung	Anwendungsfall	Best Practices

Tabelle 1: Klassifikationsrahmen

3 Klassifikation von Ansätzen zur Serviceidentifikation

Um serviceorientierte Architekturen erfolgreich einsetzen zu können, bedarf es vor allem eines Ansatzes zur systematischen Identifikation von Services [Ars04], der auf das Konzept einer SOA abgestimmt ist. Ziel dieses Kapitels ist es daher, in der Literatur publizierte Ansätze zur Identifikation von Services nach den in Abschnitt 2 identifizierten Kriterien zu klassifizieren und im Hinblick auf die geforderte systematische Identifikation zu bewerten. Die Auswahl der Ansätze erfolgte dabei durch eine breit angelegte LiteratURAUSWERTUNG, die einschlägige Journale und Tagungsbände der (Wirtschafts-) Informatik umfasste. In die Betrachtung einbezogen wurden dabei primär Ansätze, die sich konkret der Identifikation von Services widmen und dabei eine Vorgehensweise erkennen lassen. Arbeiten, in

denen die Identifikation dagegen lediglich erwähnt wird, wurden nicht in die Betrachtung einbezogen.

In Abschnitt 3.1 werden dabei zunächst Ansätze vorgestellt, die speziell auf die Identifikation von Services bei der SOA-Gestaltung ausgelegt wurden. Diese werden in Abschnitt 3.2 um weitere Ansätze ergänzt, die zur Identifikation von Informationssystemteilen allgemein bzw. von Software-Komponenten entwickelt wurden, sich prinzipiell jedoch auch zur Identifikation von Services verwenden lassen. Abschnitt 3.3 schließt mit der zusammenfassenden Gegenüberstellung der vorgestellten Ansätze.

3.1 Spezialisierte Ansätze zur Identifikation von Services

Service-Oriented Analysis and Design (Zimmermann et al. und Arsanjani). Die Eignung und Übertragung bewährter Methoden der Softwareentwicklung bei der Einführung von SOA untersuchen Zimmermann et al. [ZKG04]. Elemente aus *Object-Oriented Analysis and Design* (OOAD), *Enterprise Architecture* (EA) *Frameworks* und *Business Process Modeling* (BPM) werden zu einem *Service-Oriented Analysis and Design* (SOAD) Ansatz kombiniert und erweitert. Qualitätsfaktoren zu SOAD werden definiert und allgemeine Ratschläge zu allen Phasen des Einführungszyklus gegeben. Ein ganzheitliches Vorgehensmodell wird nicht beschrieben. Die meisten der in Kapitel 2.3 angeführten Kriterien zur Modellbildung werden angesprochen, jedoch bleiben konkrete Empfehlungen ebenso aus, wie eine Definition des Servicebegriffs. Im Hinblick auf eine Serviceidentifikation wird darauf hingewiesen, dass SOA meist nicht auf der grünen Wiese, sondern auf Grundlage bereits existierender Strukturen eingeführt wird, weshalb ein reiner *Top-Down*-Ansatz nicht ausreichend wäre. Die mangelnde Anwendbarkeit klassischer Entwicklungsmethoden auf die Servicefindung kommentieren die Autoren mit „*there is room for additional creative thinking*.“ [ZKG04], ohne jedoch Alternativen darzustellen. Das zur Veranschaulichung angeführte, theoretische Fallbeispiel unterstreicht den Empfehlungscharakter des Beitrags, der als Grundlage für weitere Forschung dienen kann.

Aufbauend auf dem von Zimmermann et al. propagierten SOAD Ansatz formuliert Arsanjani [Ars04] konkretere Empfehlungen zur Identifikation, Spezifikation und Realisierung von Services. Im Sinne einer eher technischen Sichtweise von Services wird insbesondere die Identifikationsphase konkretisiert. Die Durchführung von *Top-Down*- und *Bottom-Up*-Analysen wird ergänzt durch ein *goal-service modeling*, um nicht identifizierte aber benötigte Services aufzudecken. Auch dieser weitestgehend theoretische Ansatz ohne Referenzbeispiel kommt nicht über eine lose Sammlung von Ratschlägen hinaus.

Service-Oriented Analysis (Erl). Erl beschreibt in seinen Büchern zu Konzeption und Design von serviceorientierten Architekturen einen Ansatz zur Identifikation von Services im Rahmen eines ganzheitlichen Entwicklungsmodells [Erl05, Erl07]. Basierend auf einer Analyse von Geschäftsprozessen sowie existierender Systemstrukturen werden Servicekandidaten bestimmt, aus denen wiederum die Services herauskristallisiert werden können. Erl unterscheidet in dem *Meet-In-The-Middle*-Ansatz elf Servicehierarchien und

-klassen, die sich teils auch von seiner allgemeinen, eher technischen Servicedefinition entfernen. Abhängigkeiten zwischen Services werden nur am Rande berücksichtigt. Eine mögliche Unterstützung durch Werkzeuge wird nicht in Betracht gezogen, ebensowenig eine Zusicherung von Qualitätsgarantien. Das Vorgehen ist im Rahmen des übergeordneten Vorgehensmodells strukturiert, jedoch wird kein explizites Verfahren angegeben, vielmehr bleibt es weitgehend bei Ratschlägen und allgemeinen Anleitungen. Die Vorgehensweise insgesamt wird durchgehend anhand von Case Studies demonstriert.

Enterprise Services Design Guide (SAP). Die SAP AG veröffentlichte 2005 im Rahmen des SAP Developer Networks einen *Enterprise Services Design Guide* [SAP05]. Dieser umfasst neben den Grundlagen von Enterprise Services insbesondere auch die Phasen Discovery und Design. Ausgehend von einer eher technischen Definition wird beschrieben wie Services in einem *Meet-In-The-Middle*-Ansatz auf zwei verschiedenen Hierarchieebenen identifiziert werden können. Einem Designer von Enterprise Services werden 16 verschiedenen Indikatoren zur Hand gegeben, die ihm helfen sollen potentielle Services aus einem Geschäftsprozess und zugehörigen Szenarien zu identifizieren. Zehn Guidelines unterstützen die darauf folgende Einteilung in einfache und zusammengesetzte Services. SAP stellt mit dem Dokument eine Anleitung mit Hinweisen zur Gestaltung von Services bereit, jedoch bleibt die konkrete Umsetzung allein dem Designer überlassen.

EA Builder (Aier). Aier beschreibt ein Vorgehen zur Identifikation einer Enterprise Architecture und insbesondere der zugehörigen Services auf Basis von Geschäftsprozessen und IT-Systemen [Aie06]. Die zu einer Architektur gehörenden Modelle werden auf Graphen abgebildet und anschließend partitioniert. Hierbei findet ein Clustering Algorithmus Anwendung, der ursprünglich zur Identifikation von Gemeinschaften in sozialen Netzwerken entwickelt wurde. Die zugrunde liegende Servicedefinition bleibt ebenso unklar wie viele weitere Details des *Meet-In-The-Middle*-Ansatz. Die Unterscheidung von Servicehierarchien und -typen wird angesprochen, deren Umsetzung im unterstützenden Werkzeug, dem *EA Builder*, ist jedoch nicht näher erwähnt. Eine Garantie für die Qualität der Ergebnisse wird nicht gegeben, allerdings wurde das Vorgehen an einem Anwendungsfall getestet.

SOA Framework (Erradi et al.). Die Identifikation von Services als Teil eines architektonischen *SOA Frameworks* (SOAF), welches die Phasen der Spezifikation und Realisierung umfasst, wird von Erradi et al. präsentiert [EAK06]. Der Ansatz kommt ohne die Formulierung einer konkreten Definition von Services aus. Auf Basis von Geschäftsmodellen werden *zu erfüllende* Services in einem *Top-Down*-Ansatz identifiziert. *Bestehende* Services werden *bottom-up* aus dem vorhandenen Code und der zugehörigen Datenstruktur herausgearbeitet. Noch zu realisierende Services werden dann durch einen Abgleich von bestehenden und zu erfüllenden Services bestimmt. Ein sog. Tool-based Mining unterstützt hierbei die *Bottom-Up*-Analyse der bestehenden Code- und Datenfragmente. Die *Top-Down*-Analyse der Geschäftsprozesse erfolgt durch eine Kombination aus Interviews und Tools. Abgesehen von Hinweisen auf eine mögliche Werkzeugunterstützung, werden

keine Werkzeuge erwähnt und auch das Vorgehen für einen Abgleich der vorhandenen sowie benötigten Services nicht näher beschrieben. Ausschließlich das Ergebnis eines Fallbeispiels, nicht jedoch die Anwendung der präsentierten Methode, wird dargestellt.

BPM and SOA Handshake (Inaganti und Behara). Ein strukturiertes Vorgehen zur Identifikation von Services beschreiben Inaganti und Behara [IB07]. In vier Schritten werden potentielle Services sowohl *top-down*, als auch *bottom-up* ermittelt und gegenübergestellt. Darüber hinaus notwendige Services werden auf unbestimmte Art und Weise ergänzt. Obwohl der Beitrag die Identifikation detaillierter und strukturierter beschreibt als bspw. Zimmermann et al. [ZKG04] und Arsanjani [Ars04], reicht er selten über eine Auflistung von Möglichkeiten und Ratschlägen hinaus. Die Entwicklung von Optimierungsverfahren und deren Unterstützung durch Werkzeuge wird nicht betrachtet. Ein Referenzbeispiel findet keine Erwähnung.

Identifikation und Gestaltung von Services (Winkler). Winkler stellt einen Ansatz vor, der neben der Serviceidentifikation auch die Phasen der Gestaltung und Umsetzung von Services abdeckt [Win07]. In der Identifikationsphase werden Services auf Basis von UML Aktivitätsdiagrammen so bestimmt, dass drei zuvor definierte Anforderungen an Services erfüllt werden. Die Anforderungen umfassen die Wiederverwendbarkeit von Services, eine Vermeidung redundanter Implementierungen in verschiedenen Services, sowie eine lose Kopplung von Services über klar definierte, einfache Schnittstellen. Die Servicefindung durchläuft die vier aufeinander folgenden Schritte der „*Erstellung von Aktivitätsdiagrammen*“, der „*Aufbereitung der Aktivitätsdiagramme*“ sowie der „*Identifikation potentieller Services*“ und zuletzt einer „*Analyse der Häufigkeit der Verwendung*“. Auf der Grundlage einer impliziten, fachlichen Servicedefinition, werden Services in einem semistrukturierten *Top-Down*-Ansatz bestimmt. Eine Optimierung der Servicestruktur findet im Rahmen der Identifikationsphase nicht statt. Auch die Einhaltung der definierten Anforderungen wird während der Servicefindung nicht garantiert. Die Unterscheidung von Servicehierarchien wird ebenso wie die Berücksichtigung von Abhängigkeiten und bestehenden Strukturen erwähnt, jedoch bleibt unklar inwiefern dies Auswirkungen auf die Servicefindung hat. Der komplette Prozess wird anhand eines kurzen Beispiels aus dem Finanzdienstleistungsbereich beschrieben. Eine Unterstützung durch Werkzeuge findet keine explizite Erwähnung.

Konzeptionsmethode zu SOA (Beverungen et al.). Beverungen et al. vergleichen unterschiedliche Ansätze zur Entwicklung serviceorientierter Architekturen und stellen als Resultat einen eigenen Ansatz vor, der die Phasen der Identifikation und Spezifikation von Services abdeckt und in ein übergeordnetes Vorgehensmodell integriert ist [BKM08]. Services werden hierbei *top-down* durch eine Dekomposition von Geschäftsprozessen identifiziert. Besonderes Augenmerk wird auf die Analyse der sog. Übernahme- und Sichtbarkeitspotenziale einzelner Prozessschritte für Geschäftspartner gelegt. Existierende Strukturen und Services finden in der Identifikationsphase Berücksichtigung. Abhängigkeiten zwischen Services werden dagegen erst in der Spezifikationsphase identifiziert. Es wird

zwischen den zwei Servicehierarchien *Process* und *Basic* unterschieden. Eine Unterscheidung von Servicetypen wird zwar angesprochen, ist aber nicht integraler Bestandteil der Methode. Ein strukturiertes Vorgehen zur Identifikation wird propagiert, allerdings bleiben Details unerwähnt. Die Möglichkeiten der Implementierung einer Optimierung, sowie die Unterstützung des Identifikationsprozesses durch Werkzeuge wird ebenfalls nicht angesprochen. Ein realer Anwendungsfall zeigt die Praktikabilität der Methode, wobei Teilschritte jedoch nicht expliziert werden.

3.2 Allgemeine Ansätze zur Identifikation von Modulen

Business Systems Planning (IBM). IBM beschreibt einen *Top-Down*-Ansatz, mit dem sich betriebliche Informationssysteme auf der Basis erhobener Geschäftsprozess- und Datenmodelle systematisch modularisieren lassen [IBM84]. Dabei werden detaillierte Arbeitsschritte und Vorgehensweisen angegeben. Für die Identifikation von Systemteilen (die sich ggf. als Services nutzen lassen) wird eine graphische Methode beschrieben, die Aktivitäten eines Geschäftsprozesses und die von ihnen jeweils verarbeiteten Daten gegenüberstellt. Auf heuristischer Basis kann dabei eine Gruppierung zusammengehöriger Aktivitäten solange optimiert werden, bis bei der Verarbeitung möglichst wenige Daten aus anderen Informationssystemteilen bezogen werden (also bestehende Datenabhängigkeiten zwischen Aktivitätsbündeln minimiert sind). Die vorgestellte Methode macht allerdings weder eine Qualitätsaussage bezüglich der erreichten Optimierung, noch lassen sich dabei bereits bestehende modulare Strukturen berücksichtigen.

CompMaker (Jain et al.). Einen aus der komponentenorientierten Anwendungsentwicklung stammenden Ansatz beschreiben Jain et al. [JCIR01]. Durch Einbeziehung von Informationen eines Domänenmodells in UML Notation sowie existierender objektorientierter Klassen, werden auf Wiederverwendung ausgelegte Komponenten in einem *Bottom-Up*-Ansatz identifiziert. Eine durch das Gruppieren von Klassen erreichte Startlösung wird mittels *Add-*, *Move-* und *Exchange*-Heuristiken, sowie *manuell* verbessert. Klassen spielen also als Bausteine der Komponenten eine hohe Rolle. Das Ergebnis wird weiterhin stark durch die vom Designer zu definierenden Präferenzen beeinflusst. Services werden in diesem sehr strukturierten, komponentenorientierten Ansatz nicht explizit fokussiert, können jedoch abgeleitet werden. Der Identifikationsprozess wird durch das Werkzeug *CompMaker* unterstützt und auf ein Fallbeispiel aus dem Autoversicherungsbereich angewandt. Eine Validierung der Ergebnisse findet keine Erwähnung.

Modularitätsanforderungen (Szyperski et al.). In seinem Buch zur komponentenorientierten Anwendungsentwicklung beschreibt Szyperski 15 *Modularitätsanforderungen*, die von identifizierten Software-Komponenten und Services idealerweise zu erfüllen sind [SGM02]. So sollten identifizierte Services nicht nur aus fachlicher und technischer Sicht eigenständig, sondern bspw. auch unabhängig von anderen weiter zu entwickeln, auszuliefern, zu installieren und zu warten sein. Daneben sollten sie auch im Rahmen der Ab-

rechnung und bei der Abwicklung von Haftungsfragen möglichst unabhängig voneinander betrachtet werden können. Die genannten Kriterien sind zwar detailliert formuliert, gehen jedoch nicht über das Niveau *allgemeiner Ratschläge* hinaus. Szyperski gibt darüber hinaus keine Hinweise, wie die Einhaltung der Kriterien bei der Identifikation ggf. durch eine bestimmte Vorgehensweise gewährleistet werden kann oder welche Modellbildung dafür zu erfolgen hat. So ergibt sich aus den Kriterien allenfalls ein konzeptioneller Rahmen, anhand dessen sich identifizierte Services validieren lassen.

BCI und BCI-3D (Albani et al.). Albani et al. beschreiben in [ADZ05, AD06] die *Business Component Identification* (BCI) Methode zur Identifikation von Komponenten, sowie deren Weiterentwicklung zu *BCI-3D*. In beiden Versionen werden Komponenten auf Basis der Prozess- und Datenstruktur eines fachlichen Domänenmodells mit Hilfe von Heuristiken identifiziert. Die Komponentenidentifikation erfolgt in einem algorithmischen *Top-Down*-Ansatz, der durch ein Werkzeug unterstützt wird und sich in den *Business Component Modeling Process* (BCMP) integriert. Der erste Ansatz lehnt sich hierbei an [IBM84] an und betrachtet lediglich Abhängigkeiten zwischen einzelnen Funktionen und Datenobjekten. Die Weiterentwicklung verwendet graphentheoretische *Clustering*-Methoden, die mit Hilfe von Start- und Verbesserungsheuristiken aus Prozessschritten und Datenobjekten Komponenten identifizieren. Hierbei werden auch Beziehungen zwischen und innerhalb von Funktionen sowie Daten berücksichtigt. Eine Qualitätsgarantie auf die heuristisch ermittelte Lösung wird nicht gegeben. Vorhandenen Strukturen sowie auftretenden Abhängigkeiten wird mittels einer Integration in das Domänenmodell nur teilweise Rechnung getragen. Fallstudien zu beiden Methoden wurden durchgeführt [SBAK05, AD06]. Durch die Festlegung von Gewichtungen während der Optimierung kann der Designer Einfluss auf das Ergebnis nehmen, was mangels klarer Empfehlungen die Anwendbarkeit der Methode erschwert.

3.3 Vergleich der Ansätze

Ein Überblick über die Ergebnisse der Klassifikation aller Ansätze ist in Tabelle 2 zusammengefasst. Hieraus wird deutlich, dass keiner der gängigen Ansätze in allen Kriterien überzeugt. So wird beispielsweise kaum einer der Ansätze durch Werkzeuge unterstützt und von keinem eine Qualitätsgarantie gegeben. Eine Validierung der Ansätze durch Best Practices ist bisher ebenfalls noch nicht erfolgt. Auffällig ist ferner, dass die aus der älteren Disziplin der komponentenorientierten Anwendungsentwicklung stammenden Identifikationsmethoden meist strukturiertere, durch Algorithmen unterstützte und besser validierte Vorgehensweisen beschreiben, als die Ansätze aus dem Gebiet der vergleichsweise jungen serviceorientierten Architekturen. Hervorzuheben ist noch, dass zahlreiche der vorgestellten Ansätze ganz offenbar ohne jede Servicedefinition auskommen.

	Zimmermann et al. [ZK04]	Arjanjani [Ar04]	Erl [Erl05, Erl07]	SAP [SAP05]	AIER [Aie06]	Ermäß [EAK09]	Inaganti, Behara [InB07]	Winkler [Win07]	Beverungen et al. [BWi08]	IBM [IBM04]	Jain et al. [JaiR01]	Szoperski et al. [SzW02]	Albani et al. [Alb02; Alb06]
Zugrundeliegende Servicedefinition	Keine	Technisch	Technisch	Keine	Fachlich	Keine	Fachlich	Keine	Keine	Keine	Keine	Keine	Keine
Formalisiungsgrad	Allgemeine Ratschläge	Allgemeine Ratschläge	Allgemeine Ratschläge	Algorithmus	Strukturiert	Strukturiert	Strukturiert	Strukturiert	Algorithmus	Allgemeine Ratschläge	Algorithmus	Allgemeine Ratschläge	Algorithmus
Übergeordnetes Vorgehensmodell	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓
Ableitungsrichtung	Meet in The Middle	Meet in The Middle	Meet in The Middle	Meet in The Middle	Meet in The Middle	Meet in The Middle	Top Down	Top Down	Top Down	Bottom Up	Bottom Up	Bottom Up	Top Down
Optimierendes Verfahren	✗	✗	✗	✗	(Heuristisch)	✓	✗	✗	✗	✓	✓	✗	✓
Modellsichten	Prozesse und Daten	Prozesse, Daten und Funktionen	Prozesse, Daten und Funktionen	Prozesse, Daten und Funktionen	Prozesse und Daten	Prozesse und Daten	Prozesse, Daten und Funktionen	Prozesse und Daten	Prozesse und Daten	Prozesse und Daten	Prozesse und Daten	Prozesse und Daten	Prozesse und Daten
Berücksichtigung existierender Systemstrukturen	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
Berücksichtigung von Unsystemsabhängigkeiten	✗	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Unterscheidung Servicetypenarchiven	✗	✓ ⁽²⁾	✓*	✓ ⁽²⁾	✗	✓ ⁽²⁾	✓ ⁽²⁾	✗	✓ ⁽²⁾	✗	✗	✗	✗
Unterscheidung von Servicetypen	✗	✗	✓*	✓	✗	✗	✗	✗	✓ ⁽²⁾	✗	✗	✗	✗
Werkzeugunterstützung	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓
Qualitätsaussage	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine	Keine
Validierung	Keine	Keine	Plausibilitätsprüfung	Keine	Anwendungsfall	Plausibilitätsprüfung	Keine	Anwendungsfall	Anwendungsfall	Anwendungsfall	Anwendungsfall	Anwendungsfall	Anwendungsfall

*11 Servicearten, ** nur Teilschritte, *** durch Integration in Domänenmodell, **** Klassen aus der OO

Tabelle 2: Einordnung der vorgestellten Methoden in den Klassifikationsrahmen

4 Verwandte Ansätze

Die Entwicklung von Ansätzen zur systematischen Identifikation von Services ist, schon aufgrund der relativ kurzen Zeit seit der dieses Thema betrachtet wird, derzeit zum großen Teil erst noch im Gange. Dennoch existieren bereits einige Publikationen, in denen entsprechende Ansätze vorgeschlagen wurden. Die Qualität dieser Ansätze und damit ihre Eignung, die gewünschte systematische Identifikation von Services im Sinne eines methodischen Vorgehens zu unterstützen, variiert jedoch deutlich. Parallel zur Entwicklung neuer Ansätze für die Identifikation von Services empfiehlt sich deshalb auch eine vergleichende Einordnung der vorhandenen Ansätze, anhand derer sich der aktuelle Stand der Technik ermitteln lässt.

In der Literatur finden sich bislang jedoch nur wenige Vergleiche existierender Ansätze zur Identifikation von Services, die eine systematische Einordnung versuchen. In ihrem Beitrag zur Konzeption serviceorientierter Architekturen vergleichen Beverungen et. al zwar unter anderem Ansätze zur Identifikation von Services [BKM08]. Allerdings liegt der Fokus der Arbeit auf dem Vergleich ganzheitlicher Ansätze, die den gesamten Entwurfsprozess einer SOA abdecken. Die zusammengestellten Ansätze sind daher nur bedingt auf die Identifikation von Services ausgelegt bzw. erwähnen diese oft nur am Rande. Spezialisierte Ansätze für die Identifikation von Services bleiben dagegen meist unberücksichtigt, da ihnen der ganzheitliche Fokus fehlt. Zudem erscheinen die genannten Vergleichskriterien mitunter willkürlich gewählt und sind zudem nicht aus der Literatur abgeleitet.

Einen Vergleich strukturierter Vorgehensweisen zur Identifikation von Software-Komponenten, die wichtige Hinweise auf die Gestaltung entsprechender Ansätze im Bereich der Serviceorientierung liefern können, führen Wang et. al durch [WXZ05]. Dort liegen allerdings vor allem Kriterien zugrunde, die Aufschluss über die gewählte Vorgehensweise geben. Dabei werden Ansätze mit einer sog. Domain-Engineering-Strategie, einer Gegenüberstellung von Daten und verarbeitenden Funktionen sowie solche mit einem Verfahren zur Gruppierung zusammengehöriger Funktionen unterschieden. Die Unterscheidung entsprechender Vorgehensweisen lässt sich auf die hier verglichenen Ansätze zur Serviceidentifikation jedoch nur schwer übertragen, da die meisten über kein vergleichbares Vorgehen verfügen. Gerade dieser Umstand kann ggf. als Indiz für die noch mangelnde Reife der meisten heute verfügbaren Ansätze fungieren.

5 Schlussbetrachtung

In Kapitel 3 wurden aktuelle Ansätze zur Identifikation von Services einander gegenübergestellt und ihre jeweiligen Stärken und Schwächen diskutiert. Dafür wurde zuvor ein detaillierter Katalog mit Kriterien aus der Methodenforschung der Ingenieursdisziplinen bzw. der (Wirtschafts-) Informatik aufgestellt, die Ansätze zur systematischen Serviceidentifikation idealerweise erfüllen sollten. Wesentliche Erkenntnisse, die den Reifegrad der bislang vorgestellten Ansätze in Frage stellen, wurden dabei bereits in Abschnitt 3.3 hervorgehoben und sollen hier nicht noch einmal wiederholt werden.

Anzumerken ist darüber hinaus jedoch, dass die verschiedenen Ansätze hinsichtlich der zu berücksichtigenden Abhängigkeiten zu anderen Services oder der unterschiedlichen Servicehierarchien bzw. Servicetypen voneinander abweichen. Eine ggf. durchzuführende schrittweise Verfeinerung bei der Serviceidentifikation wird deshalb nicht von allen Ansätzen automatisch unterstützt. Durch die von manchen Ansätzen vorgenommene Trennung sog. Entity und Task Services wird in der Regel ein anderes Ergebnis bei der Serviceidentifikation erreicht als bei Ansätzen, die auf diese Unterscheidung verzichten. Hierauf ist bei der Auswahl eines Ansatzes ggf. zu achten.

Erheblicher Forschungsbedarf besteht insbesondere bei der offenen Frage, wie sich die Ansätze zur Identifikation von Services hin zu ausgereiften Methoden mit einer stärker formalisierten und detaillierten Vorgehensweise weiterentwickeln lassen. Nur so wird sich eine systematische Identifikation von Services im Sinne eines ingenieurmäßigen Vorgehens realisieren lassen. In diesem Zusammenhang ist auch der Einsatz von optimierenden Verfahren und Vorgehensweisen, die zumindest eine Abschätzung der Güte von erreichten Ergebnissen zulassen, stärker voranzutreiben.

Bezeichnend ist, dass bei der Identifikation von Services nicht auf bereits bestehende Ansätze der Komponentenorientierung zurückgegriffen, sondern ganz offenbar wieder von Neuem begonnen wird. Diese Beobachtung trifft für das Vorgehen im Kontext serviceorientierter Architekturen an vielen Stellen zu und wird in der Literatur deshalb auch zu Recht kritisiert [SGM02]. Bei genauerem Hinsehen lässt sich jedoch feststellen, dass zahlreiche Methoden, die im Kontext sog. Fachkomponenten (engl. Business Components) ebenfalls mit starkem fachlichen Bezug entwickelt wurden, durchaus bei der Gestaltung serviceorientierter Architekturen eingesetzt werden können. Eine synergetische Betrachtung beider Disziplinen könnte das Entstehen ausgereifter Methoden für die Serviceidentifikation deshalb nachhaltig beschleunigen.

Literatur

- [AD06] Antonia Albani und Jan L.G. Dietz. The Benefit of Enterprise Ontology in Identifying Business Components. In *IFIP World Computing Conference*, Santiago de Chile, Chile, August 2006.
- [ADZ05] Antonia Albani, Jan L.G. Dietz und Johannes Maria Zaha. Identifying Business Components on the basis of an Enterprise Ontology. In D. Konstantas, J.-P. Bourrieres, M. Leonard und N. Boudjida, Hrsg., *Interoperability of Enterprise Software and Applications*, Seiten 335–347, Geneva, Switzerland, 2005. Springer Verlag.
- [Aie06] Stephan Aier. How Clustering Enterprise Architectures helps to Design Service Oriented Architectures. In *2006 IEEE International Conference on Services Computing (SCC 2006), 18-22 September 2006, Chicago, Illinois, USA*, Seiten 269–272. IEEE Computer Society, 2006.
- [Ars04] Ali Arsanjani. Service-oriented modeling and architecture - How to identify, specify, and realize services for your SOA. IBM developerWorks Web services zone, Nov. 2004.
- [BKM08] Daniel Beverungen, Ralf Knackstedt und Oliver Müller. Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung - Eine Konzeptions-

methode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte. *Wirtschaftsinformatik*, 50(3):220–234, 2008.

- [Bro00] A. W. Brown. *Large-Scale, Component-Based Development*. Prentice Hall, Upper Saddle River, NJ, 2000.
- [BS04] Jörg Becker und Reinhard Schütte. *Handelsinformationssysteme - Domänenorientierte Einführung in die Wirtschaftsinformatik*. Redline, Frankfurt, 2004.
- [BS06] Hans-Jörg Bullinger und Peter Schreiner. Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. In Hans-Jörg Bullinger und August-Wilhelm Scheer, Hrsg., *Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen*, Kapitel 1, Seiten 53–84. Springer, 2006.
- [CD01] John Cheesman und John Daniels. *UML Components. A Simple Process for Specifying Component-Based Software*. Addison-Wesley, Upper Saddle River, NJ, 2001.
- [Dav93] A. M. Davis. *Software Requirements. Objects, Functions, and States*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [DW99] Desmond Francis D’Souza und Alan Cameron Wills. *Objects, Components, and Frameworks with UML. The Catalysis Approach*. Addison-Wesley, Upper Saddle River, NJ, 1999.
- [EAK06] Abdelkarim Erradi, Sriram Anand und Naveen Kulkarni. SOAF: An Architectural Framework for Service Definition and Realization. *Services Computing, 2006. SCC ’06. IEEE International Conference on*, Seiten 151–158, Sept. 2006.
- [Erl05] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [Erl07] Thomas Erl. *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2007.
- [HS00] P. Herzum und O. Sims. *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*. John Wiley & Sons, New York, NY, 2000.
- [IB07] Srikanth Inaganti und Gopala Krishna Behara. Service Identification: BPM and SOA Handshake. Bericht, BPTrends, March 2007.
- [IBM84] IBM Corporation. Business Systems Planning: Information Systems Planning Guide. Technical report ge20-0527-4, International Business Machines Corporation, 1984.
- [JCIR01] Hemant Jain, Naresh Chalimeda, Navin Ivaturi und Balarama Reddy. Business Component Identification - A Formal Approach. In *EDOC ’01: Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*, Seite 183, Washington, DC, USA, 2001. IEEE Computer Society.
- [Jun07] Dieter Jungnickel. *Graphs, Networks and Algorithms*. Springer, Berlin, 3. Auflage, 2007.
- [KL04] Donald Kossmann und Frank Leymann. Web Services. *Informatik Spektrum*, 26(2):117–128, 2004.
- [MDW02] Rainer Minz, Anthony Datel und Holger Wenzky. Web Services - nur eine Schimäre? *Information Management and Consulting*, 17(3):6–12, 2002.
- [Mil71] H.D. Mills. Top-down programming in large systems. In R. Ruskin, Hrsg., *Debugging Techniques in Large Systems*, Seiten 41–55. Prentice Hall, 1971.

- [MSJL06] James McGovern, Oliver Sims, Ashish Jain und Mark Little. *Enterprise Service Oriented Architectures*. Springer, 2006.
- [Nat03] Yefim V. Natis. Service-Oriented Architecture Scenario. Bericht ID Number: AV-19-6751, Gartner Research, 2003.
- [OT07] Sven Overhage und Klaus Turowski. Serviceorientierte Architekturen - Konzept und methodische Herausforderungen. In V. Nissen, M. Petsch und H. Schorcht, Hrsg., *Service-orientierte Architekturen. Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen*, Seiten 3–17. Deutscher Universitätsverlag, 2007.
- [Par72] D. L. Parnas. On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15(12):1053–1058, 1972.
- [PBFG03] Gerhard Pahl, Wolfgang Beitz, Jörg Feldhusen und Karl-Heinrich Grote. *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung - Methoden und Anwendung*. Springer, Berlin, Heidelberg, 2003.
- [PTD⁺06] Mike Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann und Bernd Krämer. Service-Oriented Computing: A Research Roadmap. In *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum für Informatik, 2006. Schloss Dagstuhl.
- [Rop99] Günter Ropohl. *Allgemeine Technologie: Eine Systemtheorie der Technik*. Hanser, München, Wien, 1999.
- [SAP05] SAP. Enterprise Services Architecture: Enterprise Services Design Guide. Bericht, SAP AG, 2005.
- [SBAK05] Bernhard Selk, Bettina Bazijanec, Antonia Albani und Sebastian Klöckner. Experience Report: Appropriateness of the BCI-Method for Identifying Business Components in large-scale Information Systems. In Klaus Turowski und Johannes Maria Zaha, Hrsg., *Component-Oriented Enterprise Applications (COEA)*, Jgg. LNI. Bd. P-70, Seiten 87–92, 2005.
- [Sch98] A. W. Scheer. *ARIS: Vom Geschäftsprozess zum Anwendungssystem*. Springer, Berlin, Heidelberg, 3.. Auflage, 1998.
- [SGM02] Clemens Szyperski, Dominik Gruntz und Stephan Murer. *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley, Harlow, 2.. Auflage, 2002.
- [Tur03] Klaus Turowski. *Fachkomponenten: Komponentenbasierte betriebliche Anwendungssysteme*. Shaker Verlag, 2003.
- [WH07] Thomas Wilde und Thomas Hess. Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung. *Wirtschaftsinformatik*, 49(4):280–287, 2007.
- [Win07] Veronica Winkler. Identifikation und Gestaltung von Services - Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich. *Wirtschaftsinformatik*, 49(4):257–266, 2007.
- [WXZ05] Zhongjie Wang, Xiaofei Xu und Dechen Zhan. A Survey of Business Component Identification Methods and Related Techniques. *International Journal of Information Technology*, 2(4):229–238, 2005.
- [ZKG04] Olaf Zimmermann, Pal Krogdahl und Clive Gee. Elements of Service-Oriented Analysis and Design - An interdisciplinary modeling approach for SOA projects. IBM developerWorks Web services zone, Jun. 2004.

Components for Growing the RESTful Enterprise

Andreas Heil^{1,2}, Johannes Meinecke¹, Martin Gaedke¹

¹Chemnitz University of Technology

09111 Chemnitz, Germany

²Microsoft Research Cambridge

CB3 0FB Cambridge, United Kingdom

¹{firstname.lastname}@cs.tu-chemnitz.de, ²v-aheil@microsoft.com

Abstract: For a modern enterprise, it is vital to be on the Web. Beyond offering human-readable Web sites, organizations increasingly use the Web as a media for machine-readable data about itself. With the help of technologies like XML feeds, RESTful Web services and semantic markup, new forms of enterprises models emerge in a bottom-up way. These models are easily consumable and facilitate the interaction with departments, partners and customers. Engineering good publishing systems is however extremely challenging. On the one hand, knowledge of many technologies is required; on the other hand, it must be easy to extend systems and data models in accordance to agile businesses. In this paper, we propose a framework of components for publishing dynamically growing enterprise models on the Web, present an implemented system and discuss its use in a case study.

1 Introduction

An enterprise model is a computational representation of an organization, covering aspects like its structure, activities or processes [FG98]. In recent years, there has been a tendency for enterprise data to grow in a bottom-up way rather than in a planned top-down way [TW06]. Simple lists, which were originally created to fulfil some personal organization need, are made available to others, get extended and become valuable assets for the enterprise. Technologically, this bottom-up form of collaboration is enabled through Web-based tools like Wikis, blogs, feeds and mashups. These technologies are suitable for both, in usage the Internet with public accessibility as well as usage in corporate networks and intranets. The dynamic growth of enterprise data in both scenarios is favoured by the principles of the Web, after which relevant resources have a URI and can thus be linked to and combined. In accordance to the Web's underlying Representational State Transfer (REST) architecture [Fi00], companies with such Web-enabled models could be called *RESTful enterprises*.

This way of exposing information comes with a number of advantages for the company. The data can be easily combined in end user-written mashups that are characterized by decreased implementation costs over traditional software development [An06]. Typical scenarios involve the combination of external sources, like geographical maps, with

internal data, like employee workplaces. The relative simplicity of standards like HTML and XML makes it easy to reuse the information in many places, as e.g. to display information on multiple web sites. This in turn improves consistency and lowers the cost to maintain the information, changing the future business behaviour [Yo07]. In addition, existing Web tools can be leveraged, as for example to provide search across the company's XML data sources [MM04].

In this paper, we concentrate on the question of how to build the necessary Web-based systems to expose enterprise models efficiently, rather than on the enterprise models themselves. In section 2, we examine the engineering challenges that arise in typical scenarios with the help of an example. From this examination, we derive the aim of our work: to identify reusable components for exposing enterprise models in a Web-compliant way. As our main contribution, we then specify a framework of components that can be instantiated to support a large number of standards and tasks in section 3. In section 4, we discuss our experience with the implementation of a number of these components, which we used to gradually build up a RESTful enterprise model. Section 5 contrasts our work with related approaches and section 6 summarizes the conclusions.

2 Exposing Bottom-Up Enterprise Models on the Web

For illustration purposes, we explain the general problem domain with an example scenario. The example covers the typical evolutionary bottom-up growth of information assets within organizations that start within small teams and turn into large-scale undertakings over time [TW06]. Figure 1 shows entities inside an organization that gradually emerge as Web-based representations, called *lists* in the following.

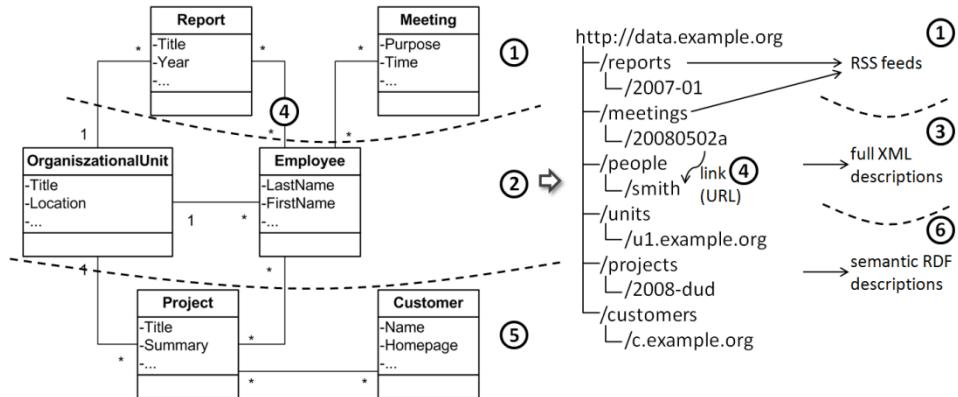


Figure 1: Example of a gradually growing enterprise model exposed on the Web.

In the beginning, a team sets up lists as ad-hoc solutions for managing reports and meetings. The content is published in the RSS format, which allows team members to subscribe to automatic notifications on any new list items and which enables other teams to integrate the lists in their homepages (1). Over time, this “model” is extended, as more

lists become available (2) and more data formats of the existing lists are offered (3). For example, in addition to the very generic RSS format, other XML formats might include geographical coordinates and domain-specific information that enable more powerful mashups. In order to improve the combined use of the data, multiple lists are then linked to each other (4). More precisely, URLs are included in the list items that point to other list items, as e.g. to point from reports the authoring employees. Data from existing project and customer management solutions is exposed in a similar way to be combined with the ad-hoc data sources (5). Future interoperability needs may demand the support for additional, semantically annotated formats (6). These can be searched, viewed and processed with a wide range of semantic tools, without the need for programming application-specific mappings.

The described way of exposing enterprise data can be seen as an application of the REST architecture style. REST comprises a set of design principles that are seen as the reason for the scalability and organic growth of the Web [Fi00]. These principles are applied here to achieve a similar advantage for enterprise models. A central requirement demands that every important entity must have an identifier so that it can be referred to. As shown in Figure 1, both lists and list items in the example are mapped to URIs. REST distinguishes between resources (the entities on the left side) and their various representations (the content types on the right side). The representation used depends on the demands of the requesting application. In the example, the same resource is reused in mashup engines, RSS readers, Web applications and Semantic Web browsers. All resources can be manipulated in a standardized way through the uniform interface of HTTP methods. In addition to read-access, this allows e.g. adding a new entry to the meeting list via an HTTP POST.

Whereas tools now become available for easily combining data sources on the Web [Ma08, Mi08, Ya07], exposing structured data beyond the simple publication of static XML files remains a largely unsupported task. [PZL08] list several design decisions to be made when programming RESTful services that include designing URIs, defining the interaction semantics of HTTP methods and choosing the data presentations. In this work we are interested in building systems that automate this process as much as possible to support the outlined target domain. We observe the following engineering challenges:

- **Gap between end users and technologies:** The initial data sources in the scenario are set up by users rather than a central IT-department. This corresponds to the findings of US Bureau of Labour Statistics [Us08], stating that 98% of users creating programs are end users without special IT-expertise who need to solve immediate non-technical problems. In contrast, the described final solution comprises a large number of technologies. While the benefits of e.g. Semantic Web formats may be highly desirable, the necessary knowledge to author them cannot be expected from the end user. Instead, technologies should be encapsulated in tools or parts of tools as much as possible.
- **Reusability of multiple resource representations:** As exemplified, the reusability of the exposed enterprise models requires the same resources to be accessible in many different representations. To free the user from the burden of managing differ-

ent representations, these should be generated automatically. For automatic mappings to work, structure must be imposed on the data. While unstructured content, as e.g. managed in Wikis, has the potential for organic growth, it cannot be reused easily and is therefore lost for applications beyond human browsing.

- **Dynamic growth of data structures:** The enterprise model in the example is gradually extended, as more data is added and linked to existing data. Corresponding systems must therefore support the effortless creation of new lists, including their structure and the mappings to their representations. Dynamic growth implies that this is possible at runtime and on the Web. Ideally, the process of extending the model can profit from the same simplicity and standard-conformance as used for manipulating individual resources.
- **Need for system extensibility:** In addition to the model, the system itself is subject to growth. In the example, this applies to the support for new formats (e.g. RDF), new data sources (e.g. customer management systems) and new client applications (e.g. search applications). Furthermore, the opening of the model to partners outside the enterprise may require adding security mechanisms that were unforeseen and unnecessary at the very beginning. The system architecture should reflect this need and allow for flexibility.

The observed challenges stress the need for solutions that comply with Web standards and that encapsulate these standards in reusable, extendable parts. Next, we therefore try to identify reusable components for exposing bottom-up enterprise models in a Web-compliant way.

3 The Data Grid Service Component Model

The WebComposition Data Grid Service (WebComposition/DGS) component model is a two-layer framework of reusable software artefacts for building customizable Web-based applications. A WebComposition/DGS component is a RESTful service for creating, managing and publishing lists of arbitrary data in the Web. It is based on a set of exchangeable sub-components to incorporate the latest technological developments. Exchanging these sub-components allows adopting the component itself to meet specific business needs. In the following we will first describe WebComposition/DGS component's building blocks (cf. Figure 2) before we will discuss the possible usage of the WebComposition/DGS component itself. Additional background information on the WebComposition/DGS component and its architectural description can be found in [HG08].

The Service Component (a) represents a central facet within the WebComposition/DGS component model. Incoming requests are accepted, processed and delegated irrespectively of their origination. Clients sending a request could be HTTP-based browsers, SOAP-based SOA components or legacy XML-RPC based clients that rely on so-called plain old XML (POX) invocations. Incoming data is processed by the Data Adapter (b). It is within the Data Adapter's responsibility to create and store data structures in form of lists using an external storage solution. Multiple Data Adapters could be specified within a WebComposition/DGS component, each responsible for a specific representa-

tion of the resources. Based on the requestor's needs, an appropriate Data Adapter is used to provide the corresponding representation of a resource. In case of a HTTP-based request this is achieved by HTTP content negotiation. Simultaneously to the creation of the data, metadata is saved along with the data. Unnoticeable to the user, a Meta Store component processes additional metadata (c). This includes e.g. creation and last update times, creator and other information that describes the actual data. The metadata already provided automatically by a Data Adapter can be completed by additional metadata added through the user. As the format of metadata varies from case to case, Input and Output Filter components (d) allow specifying the transformation of metadata formats into and from the internal representation within the Meta Store component. To be deployed in various business scenarios, an optional Access Control component incorporates the authentication and authorization functionality of external security components (e). Finally, optional sub-components can be added to a WebComposition/DGS component extending the Service Component's capabilities with additional unforeseen functionality (f).

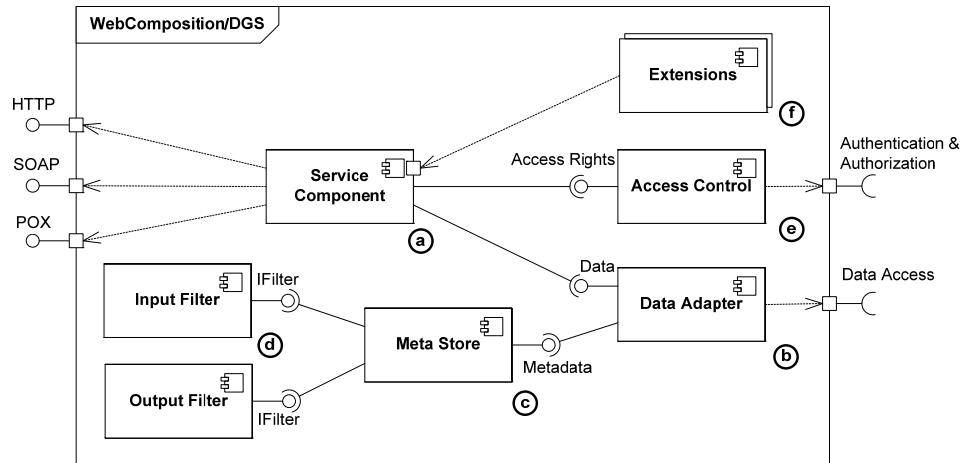


Figure 2: UML component diagram of WebComposition/DGS sub-components.

The components described thus far allow customizing a WebComposition/DGS component to a large extent. Once customized and configured, the WebComposition/DGS acts as a component by itself that can be seen as a black-box, hiding its internal functionality (cf. Figure 3). A core concept of the WebComposition/DGS component (i) is its capability of being accessible in a uniform way through different types of clients. This includes standard Web-based applications such as common Web sites (ii), RSS feed readers as well as Semantic Web browsers. In addition, specialized clients can be used to access more specific functionality of a WebComposition/DGS component (iii). One example of this type of clients, a list managing application, will be discussed in the section 4. Based on a generic SOAP interface, derived from our previous work [MMG07], SOA-based clients can also access the WebComposition/DGS component (iv). The way enterprise models are stored depends heavily on the particular requirements and constraints of a business. Hence, this is foreseen as self-subsistent component within the system (v) allowing to apply database management systems or simple file based solution

– based on the particular requirements. This also allows reacting to the growing needs of projects that originally start small and increase in size over time. Another form of potential evolution is foreseen by the security component that provides external, centralized authentication and authorization functionality to complement the access control enforced within the WebComposition/DGS component (vi). Possible security components applied for this reason include, but are not limited to the Identity Federation System (idFS) [MNG05], Shibboleth implementations [In08] or the Active Directory Federation Services (ADFS) [Pi05].

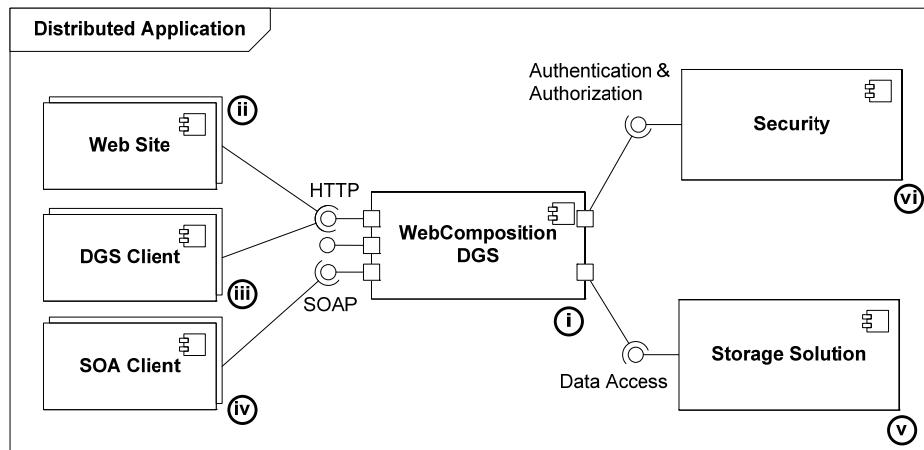


Figure 3: Distributed application based on a WebComposition/DGS component.

The proposed two-layer component architecture allows the disciplined and cost efficient engineering of Web-based application suitable to the current evolution stage. In the beginning, the out-of-the box functionality of a WebComposition/DGS component in conjunction with functionality of already available components could be sufficient to deal with scenarios where unstructured data is created ad-hoc. The bottom-up growth of this information as well as its increasing need for structures is tackled by exchangeable components addressing the particular needs. In the following section we will discuss this evolution of a system, based on a real data stock that grew over time.

4 Case Study

We now describe a project in which the presented component model was applied to bottom-up enterprise models. In the case at hand, the enterprise model represents the structure of a research group. Performed in the academic field, the nature of the exposed model is not restricted to the research domain, so that we consider the solutions applicable to enterprises in general. The study was conducted for 6 months in a production environment, using real, externally visible data. During this time, the model was gradually exposed and extended with new resources, new representations and new components according to the emerging needs of the group. A part of it was transformed from origi-

nally unstructured media, mainly managed with Wiki software. This old form of data management proved to be too hard to integrate and consume outside the Wiki itself. Therefore, an implementation of the specified WebComposition/DGS was used to successively expose data in accordance to Web standards and the REST principles. Over time, this included publications, courses, projects, student projects and people. The lists typically contain several hundred entries, describing both historical and currently active data. In addition, the model was extended several times to accommodate for new information needs (e.g. adding archive IDs to publications), to introduce links between different resources (e.g. relating publications and people) and to add new representations (cf. Figure 4).

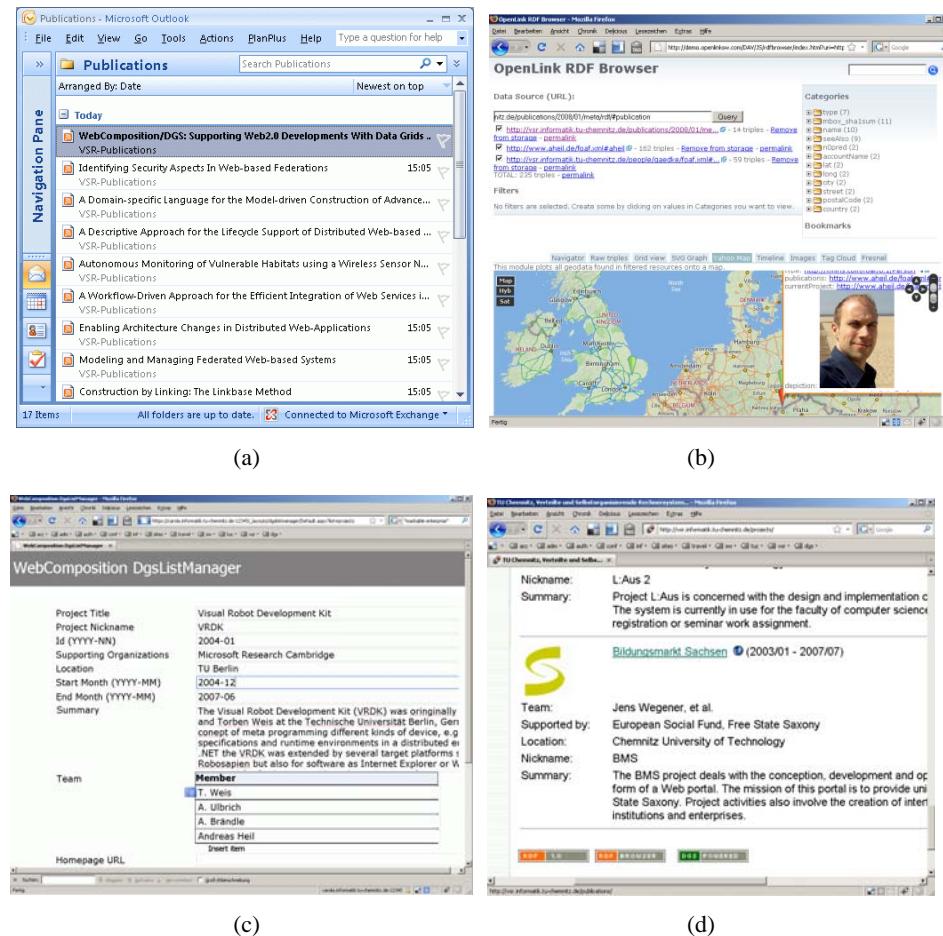


Figure 4: Representations of resources from the WebComposition/DGS in an RSS reader (a), in a Semantic Web browser (b), in the DGS List Manager (c) and on a Web site (d).

The technical realization included the following components:

- As the foundation, we used our standard implementation [HG08] of the **WebComposition/DGS** that is based on the .NET Framework. In this implementation, sub-components (cf. Figure 2) correspond to dynamic link libraries that are composed according to configuration files.
- The **XML Data Adapter** provided the necessary facilities for managing arbitrary XML lists. The adapter supports the creation of new lists at runtime as well as the enforcement of XML-Schemas to ensure that the data conforms to a given structure.
- In addition to the default file-based XML storage, a **Database Storage Solution** based on SQLite was developed to verify the exchangeability of storage solutions.
- Another component, the **Dynamic Transformation Adapter**, was used to generate new presentations from existing XML resources based on XSL stylesheets, as e.g. to generate RSS-feeds (cf. Figure 4a).
- Later, a **RDF Data Adapter** was developed to generate RDF representations of resources and metadata according to Linked Data guidelines [BCH07]. Once a list structure has been mapped to common ontologies, the model information can be re-used in Semantic Web tools together with other data from the Web (cf. Figure 4b).
- For managing the model stored in the WebComposition/DGS, a **DGS List Manager** was developed. This generic client component retrieves the XML schemas from the list manager and automatically generates forms for editing resources (cf. Figure 4c).
- The integration into the research group's Web site was supported with a **PHP List Client Component**. The component encapsulates the technical details of rendering the data and generating the necessary links to other representations (cf. Figure 4d).

We summarize the lessons learned during the case study with respect to the engineering challenges outlined in section 2. The concept of encapsulating technologies in components proved to be an important factor for end user support. With the developed system, new lists can be created ad-hoc and are automatically editable in Web forms, without any scripting or code deployment. They can also be integrated into Web pages without the need to know the involved internal components, transformations, protocols and formats. Whereas in our current system XML schemas and XSL stylesheets need to be specified when creating new lists, the architecture allows for adding more user-friendly components that automate this process further. The applied components also favored the reusability of the resources by automating the process of generating content representations. On the Web site alone, the data could be integrated at multiple locations for realizing different views on it, e.g. on personal homepages, on project pages and on central group pages. Furthermore, the study illustrated the system's potential for bottom-up data growth. Natural limits to flexible changes arose from the Web's need for stable URIs and from necessary updates of the corresponding mappings into formats and ontologies. The demand for system extensibility was met by the component-based architecture. As demonstrated, components were gradually added to the WebComposition/DGS, while the service was in productive use, i.e. integrated into the group's Web site.

The described systems are still in use and will be extended in future for further studies. A demo of the WebComposition/DGS as well as downloadable components can be found at <http://www.webcomposition.net/dgs>.

5 Related Work

The growing demand for creating, managing and publishing data within Web-based solutions is reflected in the high number of ongoing developments in this field. In this section we discuss approaches related to our work. We concentrate on solutions that follow the REST-style principles, allow building and hosting enterprise models and are applicable to our case study. The solutions are segmented into application-specific, database-driven and protocol-driven approaches.

Application-specific container solutions: This class of solutions includes Wikis, Weblogs and portals that maintain data in the form of lists. Wikis provide lists with extensive change-histories and references to list items that do not yet exist. Weblogs are also part of this class by providing chronologically ordered lists of entries and different views (e.g. by tags, month, week or day) on the data. While both follow REST-like architectures and gain increasing recognition within companies [Li07], they mainly focus unstructured content for human readers. An example of a commercial list-oriented application is Backpack [37Si08]. This easy-to-use Web-based service provides basic functionality to manage and share lists such as notes, images and files on the Web. The data model is however preset and not subject to growth. The more complex document management system Microsoft Office SharePoint [Mi07] provides the capability of maintaining lists of documents, tasks, contacts, appointments as well as self-defined lists. While this theoretically allows managing arbitrary models, the monolithic approach taken makes it hard to reuse the structure outside the portal itself in a standardized way. Generally, container solutions are suitable for dedicated tasks, offer a limited set of functionality and appear mostly as stand-alone solutions that are difficult to integrate into existing component-based environments.

Database-driven approaches: Currently, a number of services can be observed that provide traditional database functionality over Web standards. These services mostly target a commercial audience and are offered by particular providers. The Amazon SimpleDB [Am08] Web service includes a simple REST-style Web service interface that follows the traditional CRUD (Create, Read, Delete, Update) pattern [Ki90]. SimpleDB is designed to store relatively small amounts of data with the focus on fast data access. For growing needs, the commercial Amazon S3 service [Sh07] is offered. In contrast to WebComposition/DGS, these services form black-box components, lack the capability of customization and do not impose structure on the data. The Microsoft ADO.NET Data Services [Ca07] comprise an implemented set of patterns for data-centric services. The data is represented in various common formats such as XML, JSON or ATOM/RSS. All the characteristics of the REST-like architectural style appear as proposed in [Fi00]. Compared to the WebComposition/DGS component, this solution is designed for the Microsoft database management systems only and introduces dependencies to very specific platforms and technologies. None of the examined database-driven approaches supported creating and changing list structures via their Web interface. Their usage often requires extensive background knowledge of database management systems.

Protocol-driven approaches: The Atom standard [GD07, NS05] defines an XML-based format and an HTTP-based protocol for publishing and editing lists of related docu-

ments. For example, the POST method is used to create new resources within a collection, including an additional link entry for an extendable set of metadata about the particular resource. The most common usage of Atom is the syndication of news and Weblogs. While the format is extendable, there is no support for serving multiple representations of a resource. The Google Data APIs [Go08] provide simple protocols for reading and writing data on the Web based on RSS 2.0 and ATOM 1.0. The CRUD concept is fully supported through a REST-style interface using HTTP and the corresponding methods to access and query the XML-based data. Metadata is provided in the form of additional feeds containing referential information using e.g. the Google Base schema. However, this strategy is limited in terms of the fixed semantic information provided by the metadata feeds. According to their protocol-driven nature, the approaches do not cover component-based architectures for realizing extendable systems.

With respect to the aimed usage for bottom-up enterprise models, the related approaches were either too rigid for gradual growth, or too unstructured for providing the model data in multiple reusable representations.

6 Conclusion

In this paper we presented the WebComposition/DGS component model as a framework for the Web-compliant exposition of enterprise models. Our work focuses on the often observed bottom-up approach to creating valuable enterprise data sources that start out as simple lists on the Web. This process is supported by breaking down functionality and technology support into an extendable set of components. Within a case study, we showed the practical application of corresponding component implementations and demonstrated their ability to represent different representations of the model for maximum reusability. Upcoming work is targeted at finding ways to better support end user tasks, like e.g. the specification of new lists. Another interesting open problem is the transparent handling of URIs as references to other list entries or machine-readable information on the Web with corresponding user interfaces.

References

- [37Si08] 37signals LLC: Backpack - Website, 2008, <http://www.backpackit.com/> (02-19-2008).
- [Am08] Amazon Web Services LLC: Amazon SimpleDB Developer Guide, 2008.
- [An06] Anant, J.: Enterprise Information Mashups: Integrating Information, Simply, in 32nd International Conference on Very Large Data Bases, VLDB Endowment, Seoul, Korea; 2006, p. 3-4.
- [BCH07] Bizer, C.; Cyganiak, R.; Heath, T.: How to Publish Linked Data on the Web, 2007.
- [Ca07] Castro, P.: Project Astoria, The Architecture Journal, 2007(13): p. 12-17.
- [Fi00] Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures, University of California, 2000.
- [FG98] Fox, M.S.; Gruninger, M.: Enterprise Modeling, AI Magazine, 1998. 19(3): p. 109-121.
- [Go08] Google Inc.: Google Data APIs - Website, 2008, <http://code.google.com/apis/gdata/> (02-17-2008).

- [GD07] Gregorio, J.; Hora, B.D.: The Atom Publishing Protocol - Request for Comments: 5023, 2007, <http://www.ietf.org/rfc/rfc5023.txt> (02-19-2008).
- [HG08] Heil, A.; Gaedke, M.: WebComposition/DGS: Supporting Web2.0 Developments With Data Grids, in IEEE International Conference on Web Services (ICWS 2008), Beijing, China; 2008.
- [In08] Internet2: Shibboleth - Website, 2008, <http://shibboleth.internet2.edu/> (08-08-2008).
- [Ki90] Kilov, H.: From semantic to Object-oriented Data Modeling, in First international Conference on Systems Integration, Morristown, NJ, USA; 1990, p. 385-393.
- [Li07] Li, C.; Stromberg, C.: The ROI of Blogging, Forrester Research Inc., 2007.
- [Ma08] Markl, V. et al.: Data Mashups for Situational Applications, in Model-Based Software and Data Integration: First International Workshop, Mbsdi 2008., Berlin, Germany; 2008.
- [MMG07] Meinecke, J.; Majer, F.; Gaedke, M.: Component-Based Content Linking Beyond the Application in Seventh International Conference on Web Engineering (ICWE), Springer, Como, Italy; 2007, p. 427-441.
- [MNG05] Meinecke, J.; Nussbaumer, M.; Gaedke, M.: Building Blocks for Identity Federations, in Fifth International Conference for Web Engineering (ICWE2005), Springer, Sydney, Australia; 2005, p. 203-208.
- [Mi07] Microsoft: Microsoft Office SharePoint Server 2007 Homepage - Website, 2007, <http://office.microsoft.com/en-us/sharepointserver/> (02-01-2007).
- [Mi08] Microsoft Corporation: Microsoft Popfly - Website, 2008, <http://www.popfly.com/> (02-19-2008).
- [MM04] Mukherjee, R.; Mao, J.: Enterprise Search: Tough Stuff, Queue, 2004. 2(2): p. 36-46.
- [NS05] Nottingham, M.; Sayre, R.: The Atom Syndication Format - Request for Comments: 4287, 2005, <http://www.ietf.org/rfc/rfc4287.txt> (02-18-2008).
- [PZL08] Pautasso, C.; Zimmermann, O.; Leymann, F.: RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision, in 17th International World Wide Web Conference, ACM Press, Beijing, China; 2008, p. 805-814.
- [Pi05] Pierson, N.: Overview of Active Directory Federation Services in Windows Server 2003 R2, 2005.
- [Sh07] Shanahan, F.: Amazon.com MashupsEds., Birmingham, UK: Wrox Press Ltd., 2007.
- [TW06] Tapscott, D.; Williams, A.D.: Wikinomics: How Mass Collaboration Changes Everything, New York, NY: B&T, 2006.
- [Us08] U.S. Department of Labor: Bureau of Labor Statistics - Website, 2008, <http://www.bls.gov/> (07-20-2008).
- [Ya07] Yahoo! Inc.: Pipes: Rewire the web - Website, 2007, <http://pipes.yahoo.com/pipes/> (02-19-2008).
- [Yo07] Young, G.O.: IT Will Measure Web 2.0 Tools Like Any Other App, Forrester Research, Inc., Cambridge, MA, USA, 2007.

Text Driven Modeling Architecture - Modellierung im Zeitalter Serviceorientierter Architekturen

Mai, André

kommfinanz GbR
Elsternweg 6
53179 Bonn
andre.mai@kommfinanz.de

Abstract: Mit Serviceorientierte Architekturen können wiederverwendbare Modellstrukturen entwickelt werden. Bei der Modellierung über Unternehmensgrenzen hinweg sind Missverständnisse vorprogrammiert, die durch eine Transformation von natürlichsprachlichen Texten in semiformale Modelle vermieden werden können. Im Beitrag werden existierende Möglichkeiten zur Extraktion von UML-Klassenmodellen aus Benutzeranforderungen, statistisch-linguistische Analysten und Heuristiken untersucht und zu einer neuen Methode kombiniert.

1. Herausforderung Serviceorientierte Architekturen

Unter Serviceorientierten Architekturen (SOA) werden Hard- und Softwarelandschaften sowie Organisationsformen verstanden, die durch eine Ausrichtung auf Dienste (engl. services) gekennzeichnet sind. Der Begriff Service wird insofern gleichbedeutend mit dem Begriff Dienst verwendet [WikiSOA]. Dienste sind insbesondere – aber nicht nur – Softwareleistungen, die einem Client von einem Server zur Verfügung gestellt werden. In der Mathematik und Informatik wird zwischen zustandslosen Diensten und Zustandsautomaten unterschieden, unter technische Services fallen RPC, CORBA, RMI, Web-Service [EiPi07, S. 13 f.]. Organisatorisch gesehen ist ein Service ein gekapselter (Teil-)Prozess mit definierten Eingangsparametern und vorgegebener Funktionalität.

Die technischen Argumente, die für eine SOA sprechen, sind lose Kopplung von Diensten durch standardisierte Kommunikationsstrukturen (z. B. [EiPi07, S. 13], [Herz05, S. 10], [WikiSOA]), wobei jedoch die Frage zu stellen ist, ob eine Kopplung von Diensten, die von Servern an verschiedenen Standorten angeboten werden, tatsächlich eine Verbesserung in der Performanz und Stabilität von Anwendungen bietet [EiPi07, S. 3].

Für die Modellierung von Geschäftsprozessen, bspw. mit ereignisgesteuerten Prozessketten (EPK, vgl. [Sche98]) bietet das Konzept der Serviceorientierung jedoch den Vorteil, während der Modellierung Kapselungen von Prozess-Schritten vorzunehmen, die an geeigneter Stelle wieder verwendet werden können. [BKnM08]

Bei der Definition von Services treten technische Probleme auf. Zum einen ist die Granularität der Dienste oft nicht klar ([EiPi07, S. 11], [Herz05, S. 13]). Hinzu kommen sämtliche Probleme, die mit verteilten Systemen einhergehen [EiPi07, Se. 3].

Konzeptuell wird deutlich, dass SOA sowohl auf der technischen Ebene in Form von Webservices und XML [Herz05, S. 11] als auch auf der organisatorischen Ebene ([EiPi07, S. 3]) diskutiert wird. Neben der uneinheitlichen Verwendung der Konzepte stellt sich insbesondere bei der Modellierung auch das Problem einer uneindeutigen Verwendung von Begriffen in Modellen ([Ortn00a], [Ortn00b]).

Um dieses Problem zu lösen, wird ein Ansatz vorgestellt, mit dem Inhalte aus Dokumenten, existierenden Modellen und vorhandener Software automatisch in Modelle transformiert werden. Somit gelingt dem Modellentwickler ein schneller Überblick über die verwendeten Begriffe und deren Strukturen und die uneindeutige Verwendung von Begriffen in Modellen wird vermieden. Durch dieses Vorgehen wird erreicht, dass Modellentwickler, die den Fachraum nicht so gut kennen, mit den relevanten Begriffen konfrontiert werden. Kennen sie deren Bedeutung, finden sie leichter Zugang zu den fremden Informationen.

Der Beitrag ist wie folgt aufgebaut: Kapitel 2 enthält Ansätze zur Normierung von Modellbegriffen. In Kapitel 3 werden mit beispielhaften Methoden des Information Retrievals und Heuristiken Wege zum Strukturieren von Informationen angegeben. Diese Ansätze werden im 4. Kapitel zu einer neuen Methode des Text Driven Modeling Architecture verbunden, mit der unstrukturierten Textinformationen in UML-Klassenmodelle überführt werden können. Die beispielhafte Modellierung eines kommunalen Informationssystems zeigt in Kapitel 5 eine Einsatzmöglichkeit der Methode demonstriert, ehe im 6. Kapitel ein Ausblick auf die automatische Extraktion von Servicekandidaten gegeben wird.

2. Normierung von Modellbegriffen

2.1. Methodisches Vorgehen

In Softwareentwicklungsprojekten müssen häufig zu Beginn Benutzeranforderungen erhoben und dokumentiert werden ([Part98]). Die Effizienz dieses Vorgehens und die Akzeptanz der entstehenden Interaktionsanwendungen lässt sich steigern, wenn die späteren Benutzer als Fachexperten in die Entwicklung einbezogen werden ([Wiem03]).

Erfahrungsgemäß sind Fachexperten nicht ausreichend in den Methoden des Software-Engineerings geschult. Sie stellen ihre Anforderungen mit Hilfe natürlichsprachlicher Dokumente (vgl. [Rupp01b], [Rupp03]) oder mit Hilfe von Prozessmodellen dar (vgl. [Sche98]), während im Software-Engineering die UML Anwendung findet (vgl. u. a. [OMG00, S. 2], [OMG01, S. 6], [Jec+04a, S. 323], [Jec+04b, S. 10]).

Das gemeinsame Verständnis der dem zu entwickelnden System zu Grunde liegenden Fachterminologie [Heye02, S. 1] hilft, die Fachexperten in das Projekt zu integrieren, und Missverständnisse bei der Interpretation von Anforderungen durch die Entwickler zu vermeiden. Sowohl die natürlichsprachliche Darstellung als auch UML-Diagramme haben ihre Berechtigung. Natürlichsprachliche Darstellungen bieten eine höhere Verständlichkeit für ungeübte Analysten. UML-Diagramme verfügen über eine höhere Präzision.

Aus diesem Grund wäre eine Kopplung beider Methoden denkbar. Der Einsatz mehrerer Methoden führt jedoch regelmäßig zu Konsistenzproblemen, wenn die jeweiligen Dokumente oder Modelle per Hand abgeglichen werden müssen (vgl. z. B. [RuSc04, S. 2]).

Um die Konsistenzprobleme zu vermeiden und den Aufwand zur Überführung eines Dokuments in ein Modell bzw. eines Prozessmodells in ein UML-Modell zu verringern, bietet sich die automatische Transformation der Anforderungen in Modelle an. Mit der Model Driven Architecture definiert die OMG Standards zur Transformation von Modellen in andere Modelle (vgl. [OMG00], [OMG01], [FeLo03], [Fran03]). Die Transformation von unstrukturierten Dokumenten in Modelle wird von der MDA jedoch nicht abgedeckt. Mit der natürlichsprachlichen Informationsbedarfsanalyse (NIBA) existiert ein Ansatz zur Generierung von UML-Modellen aus Dokumenten, der im Abschnitt 0 vorgestellt wird.

2.2. Natürlichsprachliches Parsing als Mittel zur Transformation von Benutzeranforderungen in semiformale Modelle

Das natürlichsprachliche Parsing basiert auf einer grammatischen Analyse der Sätze in einem Dokument. Jeder Satz wird in seine einzelnen Worte zerlegt, deren Beziehungen anhand der verwendeten Verben und ihrer Stelligkeit extrahiert werden [FIKM00], [FIMM00].

Im NIBA-Projekt – der Natürlichsprachlichen InformationsBedarfsAnalyse – wurde ein Parser und Mapper implementiert, mit deren Hilfe die entsprechenden Beziehungen gewonnen und in ein UML-Klassenstrukturdiagramm überführt werden können. [FIWe00] Ein Ausschnitt ist in Abbildung 1 dargestellt.

Um dieses Verfahren für die gesamte deutsche Sprache anzuwenden, ist die komplette Abbildung aller deutschen Verben und mit ihren jeweiligen Stelligkeiten notwendig. Unter Stelligkeiten wird die Anzahl der Subjekte und Objekte verstanden, die im Kontext eines Verbs auftauchen. [FIWe00] Geeignete Algorithmen zur generischen Extraktion dieser Beziehungen sind für deutsche Verben nicht vorhanden. Damit wird das Verfahren bei umfangreichen Texten oder Spezialkontexten sehr aufwändig, wenn das entsprechende Verzeichnis manuell gepflegt werden muss.

In Kapitel 3 werden ein Verfahren zur statistisch-linguistischen Analyse von unstrukturiertem Texten sowie eine Methode zur Überführung von Benutzeranforderungen in UML-Modelle vorgestellt.

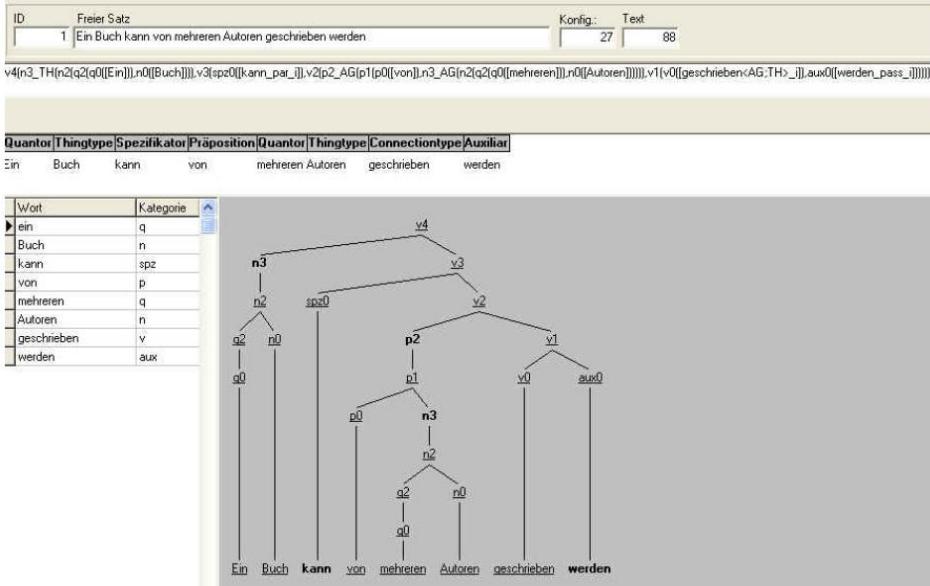


Abbildung 1: Natürlichsprachliches Parsing (aus [FIKM00])

3. Strukturierung von Benutzeranforderungen

3.1. Patternmatching und statistisch-linguistische Analysen

Der Ansatz des Pattern-Matchings beruht auf einer Matching-Operation `match[pattern, text]`, mit der entweder das Vorhandensein oder alle Positionen eines als Pattern definierten Strings in einem Text oder einem Dokument bestimmt werden [Sun90]. Es existieren verschiedene Verfahren zur Analyse von Texten, wie der Algorithmus von Boyer and Moore, das approximative Pattern-Matching, der Soundex-Algorithmus [Fuh96]. Ergebnis dieser Verfahren sind Begriff-Zitat- oder Begriff-Dokument-Beziehungen.

Als Beispiel für das Pattern-Matching soll der Ansatz der automatischen Gewinnung von Fachbegriffen dienen [Heye02], der in Form des Wortschatztools [Wort] implementiert wurde. Mit Hilfe des Wortschatztools lassen sich aus dem Deutschen Wortschatz¹ die häufigsten Worte, Kookkurrenzen² sowie linke und rechte Nachbarn aus einem Text extrahieren und die bestehenden Beziehungen grafisch aufbereiten (vgl. [Wort]).

¹ Im Deutschen Wortschatz werden "sorgfältig ausgewählte öffentlich zugängliche Quellen automatisch erhoben". [Wort, Hinweis unter einem Suchergebnis]

² Unter Kookkurrenzen wird das gemeinsame Auftreten von Wörtern in einem Dokument verstanden. Es wird angenommen, dass die beiden Wörter interdependent sind, wenn sie häufig gemeinsam in Dokumenten auftreten. (vgl. <http://de.wikipedia.org/Kookkurrenz>, [Heye02, S. 4, wobei in der Quelle auf Kollokationen, also das unabhängige gemeinsame Auftreten von Wortformen, abgestellt wird]

Neben den Worten werden im Wortschatztool Zitate angezeigt, in denen die gesuchten Worte vorkommen. Außerdem enthält die zugrunde liegende Technologie die Möglichkeit zur Identifikation von Aussagen, die im Zusammenhang mit den Worten besonders häufig auftauchen.

Eine Technologie, wie sie im Wortschatztool vorliegt, bildet somit die Grundlage für die Extraktion von Begriffen und Zitaten aus Dokumenten.

3.2 Heuristiken

1998 wurde von Mario Jeckle ein Ansatz zur Strukturierung von Anforderungen aus natürlichsprachlichen Dokumenten vorgestellt ([Jeck98]). Dieser Ansatz basiert auf drei Heuristiken, die auf einen natürlichsprachlichen Fachtext angewendet werden:

1. "Mark every noun as candidate class which is not a meaningless expletive or part of a description clarifying some previous expressed information." [Jeck98, S. 3]
2. "Check every identified noun (candidate class) if it can receive a value expressible using the basic data types of the intended implementation language. If so the consider expressing the noun as attribute of a class." [Jeck98, S. 4]
3. "A verb within the textual specification linking two nouns (i. e. concepts) together could be an association within the analysis model. If one of the stated nouns is previously identified as attribute, the verb expresses the implicit interrelation to the containing class. Else all the related nouns were identified as candidate classes, they will be interrelated using a UML association." [Jeck98, S. 5]

Der Modellentwickler soll also zunächst alle Substantive im natürlichsprachlichen Fachtext als Klassenkandidaten markieren, dann bewerten, ob die bedeutungslos oder Teil einer erklärenden Wiederholung sind. Kann ein gefundene Klassenkandidat einem Basisdatentyp der zu implementierenden Programmiersprache zugeordnet werden, wird er als Attribut modelliert. Mit den Verben werden Beziehungen im Modell angelegt.

In [Jeck98] ist ein Beispiel angegeben, in dem mit Hilfe dieser Heuristiken aus einer Prozessbeschreibung ein Klassenmodell entwickelt wird. Es wird deutlich, dass die Identifikation der Objekt- und Attributkandidaten sowie ihrer Beziehungen intellektuelle Fähigkeiten bei der Anwendung der Heuristiken erforderlich sind, da sich die Entscheidungsfindung über die Bedeutungslosigkeit einzelner Worte, ihre Zuordnungsmöglichkeit zu einem Datentyp oder selbst die Zugehörigkeit von Substantiven zu Verben nicht oder nur mit großem Aufwand (s. o. 0) automatisieren lässt.

In [GeMa03] sind ebenfalls mit Syntaxmustern, Patternorientierter Prozessmodellierung und Sprachnormierung Heuristiken zur Extraktion von Modellinformationen aus natürlichsprachlichen Texten vorgestellt, die mangels Toolunterstützung ebenfalls nur in Modellierungsregeln festgelegt werden können und durch manuell angewandt werden.

4. Kopplung der Verfahren

Die in den vorangegangenen Abschnitten vorgestellten Verfahren zeichnen sich entweder durch hohen Aufwand (NIBA, vgl. 0, bzw. Heuristiken, vgl. 0) oder durch Unvollständigkeit (Wortschatz-Technologie, vgl. 0) aus. Koppelt man die Wortschatz-Technologie mit den Heuristiken, die Jeckle vorgestellt hat, zu einer neuen Methode, eröffnet sich die Möglichkeit zur automatischen Extraktion von kontextspezifischen Fachwissen, das bei der Modellierung von Fachräumen genutzt werden kann.

Nachfolgend soll der dem gekoppelten Verfahren zu Grunde liegende Algorithmus an einem Beispiel erläutert werden, ehe in einem Ausblick die noch zu beantwortenden Fragen aufgeführt werden.

Grundlage des Algorithmus' sind Textdokumente D des der Software zu Grunde liegenden Fachkontexts. Diese werden mit Hilfe der Wortschatz-Technologie in Worte im Singular W und Zitate Z zerlegt. Außerdem wird eine Liste der Trivialworte und Eigennamen T sowie die Liste der attribut- und werttypischen Worte A gebildet. Unter einem "attributtypischen" Wort sind zu verstehen, die von der Bezeichnung her auf ein Attribut schließen lassen, z. B. ID, Name, Bezeichnung, Größe, Farbe. "Werttypische" Worte stellen Werte, wie true, false, Millionen, DM, Euro, cm, dar. Die attribut- und werttypischen Worte sind manuell zu bilden bzw. zu verwalten und werden in den Modellierungsprojekten fortgeschrieben. Es gilt: $D \supseteq Z \supseteq W \supseteq A, W \supseteq T$.

Um alle Zitate zu finden, die ein bestimmtes Wort $w \in W$ enthalten, wird eine Abbildung Z' definiert: $Z': W \rightarrow Z: w \in W \rightarrow z' \in Z'(W) \mid w \in z'$.

Das Wort und seine Zitate werden in einem Tupel $B(W_B, Z'(W_B))$ gespeichert, welches gemäß dem semiotischen Dreieck der Sprachwissenschaft (vgl. [HeSi80, S. 236]) den Begriff B widerspiegelt.

Zu jedem Begriff wird die Häufigkeitsklasse H_B gespeichert (vgl. [Heye02, S. 3]), mit der der Begriff im Kontext auftritt. Die häufigsten Begriffe, die keine Trivialworte T darstellen, werden relevante Begriffe B_r genannt. Es gilt: $\forall b \in B_r: H_B(W_b) > H_S \wedge W_b \notin T$, wobei H_S als signifikante Häufigkeit für Begriffe definiert ist.

Die Beziehungen zwischen Begriffen lassen sich durch die Stärke von Kookkurrenzen bestimmen (vgl. [Heye02, S. 4]). Die Stärke der Kookkurrenzen $Ko(B_1, B_2)$ zwischen den relevanten Begriffen lässt sich als Graph darstellen. Aus Übersichtlichkeitsgründen soll der Graph nur Kookkurrenzen enthalten, deren Häufigkeit einen bestimmten Wert H_K übersteigen. Die Menge der Kookkurrenzen mit signifikant hoher Häufigkeit stellt die Menge der relevanten Kookkurrenzen dar. $\forall k \in K_r(B_1, B_2): K(B_1, B_2) > H_K$.

In Anlehnung an die Heuristiken in [Jeck98] werden auf die Menge der relevanten Kookkurrenzen Heuristiken angewendet. Dazu werden die Menge der UML-Klassen Kl_B und UML-Attribute Att_B gebildet, wobei das tiefgestellte B die Bezeichnung der Klasse mit dem Begriff B andeutet. Die Heuristiken lauten:

1. Jeder Begriff, der relevante Kookkurrenzen zu mindestens zwei anderen Begriffen besitzt, wird Klassenkandidat.
 $\forall b \in W_B, B1, B2 \in W: \exists Ko(b, B1) > 0 \wedge \exists Ko(b, B2) \wedge B1 \neq B2 \Rightarrow b \rightarrow Kl_b$
2. Jeder Begriff, der relevante Kookkurrenzen zu nur einem anderen Begriff besitzt, wird Attribut: $\forall b \in W_B, B1, B2 \in W: \forall Ko(B, B1) > 0 \wedge \forall Ko(B, B2) > 0 \Leftrightarrow B1 = B2 \Rightarrow b \rightarrow Att_b$
3. Ein Begriff, der ein "attribut-" bzw. "werttypisches" Wort enthält, wird Attribut.
 $\forall b \in W_B: b \in A \Rightarrow \exists Att_b$

Im Gegensatz zu Jeckles Heuristik 2 (vgl. Abschnitt 0) werden die Worte automatisch gegen eine Liste geprüft, in der attribut- bzw. werttypische Worte enthalten sind.

Während bei [Jeck98, S. 5] die UML-Assoziation mit Hilfe von Verben gebildet wird und somit entweder die Infrastruktur von NIBA (vgl. Abschnitt 0) oder intellektuelle Leistungen erfordert, wird im vorgestellten Ansatz die UML-Assoziation mit Hilfe der relevanten Kookkurrenzen gebildet, die automatisch ausgewertet werden können.

5. Beispiel: kommunales Ratsinformationssystem

Der vorgestellte Algorithmus soll nun am Beispiel eines Ratsinformationssystems dargestellt werden. Diese Informationssysteme dienen der Arbeit der kommunalen Parlamente und zum Austausch von Terminen, Agenden und Protokollen der Sitzungen der Parlamente und Ausschüsse.

Im Wortschatztool wurde eine Suche nach den Begriffen *Gemeinderat*, *Sitzung* und *Tagesordnung* durchgeführt. Die Suche lieferte die in Abbildung 2 enthaltenen Kookkurrenzen.

im (939), der (625), beschlossen (480), hat (392), Verwaltung (364), Bürgermeister (355), Mehrheit (326), Sitzung (258), Stadt (241), Stadtverwaltung (238), Stuttgarter (219), für (212), Oberbürgermeister (211), Der (203), zugestimmt (186), einstimmig (151), dem (146), beschloß (133), Esslinger (131), jetzt (126), Stimmen (120), Bürgerentscheid (119), vom (114), Beschluß (112), Grünen (107), Fraktionen (105), Antrag (104)

Abbildung 2: Kookkurrenzen zu den Worten Gemeinderat, Sitzung und Tagesordnung

Werden die Trivialworte T ausgeblendet, bleiben die in Abbildung 3 genannten Kookkurrenzen. Bei der Analyse der Worte wird deutlich, dass die Begriffe *beschlossen*, *beschloß* und *Beschluß* sowie die Begriffe *Bürgermeister* und *Oberbürgermeister* synonym verwendet werden.

beschlossen (480), Verwaltung (364), Bürgermeister (355), Mehrheit (326), Sitzung (258), Stadt (241), Stadtverwaltung (238), Oberbürgermeister (211), zugestimmt (186), beschloß (133), Bürgerentscheid (119), Beschluß (112), Fraktionen (105), Antrag (104)

Abbildung 3: Kookkurrenzen zu den Worten Gemeinderat, Sitzung und Tagesordnung

Ermittelt man die Stärke der Kookkurrenzen, ergibt sich bei einem Signifikanzlevel $H_K=245$ der in Abbildung 4 dargestellte Graph. Die Dicke der Kanten gibt die Stärke der Kookkurrenzen an.

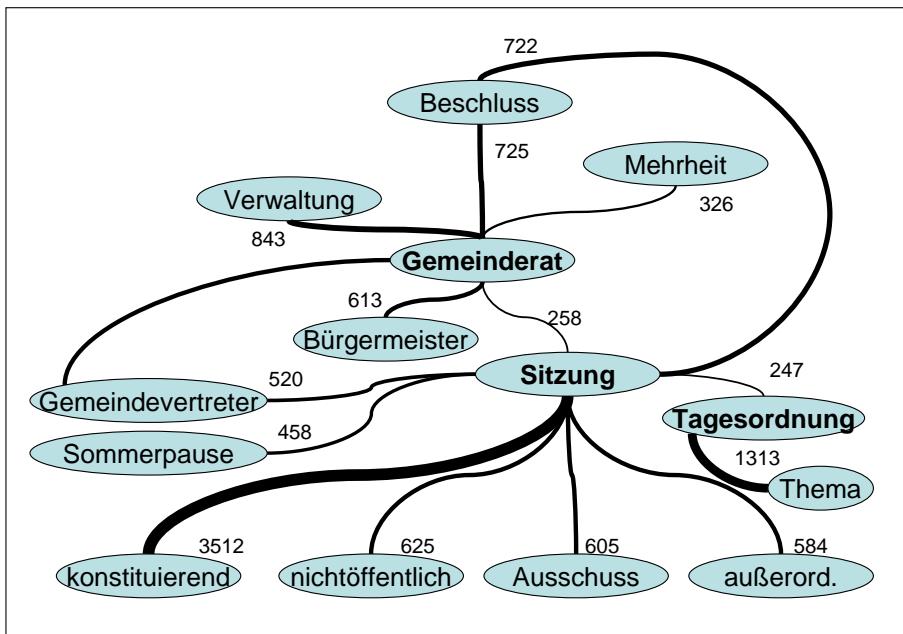


Abbildung 4: Graph über die signifikanten Kookkurrenzen

Aus Heuristik 1 folgt, dass alle Substantive im Graph Klassenkandidaten im UML-Klassenmodell werden. Da die Begriffe *konstituierend*, *nichtöffentliche*, *Ausschuss*, *außerordentlich*, *Thema*, *Mehrheit*, *Verwaltung*, *Bürgermeister* und *Sommerpause* nur Beziehungen zu einem anderen Begriff haben, werden nach Heuristik 2 Attribute. Somit lässt sich das in Abbildung 5 abgebildete UML-Modell generieren.

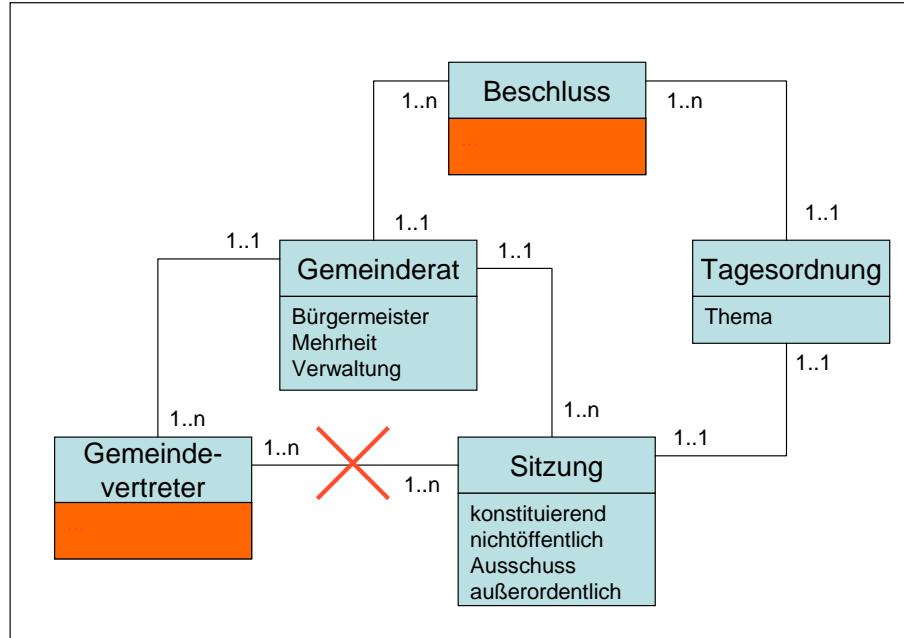


Abbildung 5: generiertes UML-Modell

Im generierten UML-Klassenmodell wurden die Klassen *Gemeinderat*, *Gemeindevertreter*, *Sitzung*, *Tagesordnung* und *Beschluss* generiert. Die Klassen *Gemeindevertreter* und *Beschluss* haben keine Attribute. Bei näherer Betrachtung ist die Beziehung zwischen den Klassen *Gemeindevertreter* und *Sitzung* nicht notwendig. Im Ratsinformationssystem werden Sitzungsprotokolle u. ä. Dokumente verwaltet. Die Anwesenheit einzelner Gemeindevertreter an konkreten Sitzungsterminen spielt in der Regel keine Rolle.

Die Diskussion mit Fachexperten ergibt, dass das Attribut *Bürgermeister* in der Klasse *Gemeinderat* seine Berechtigung hat, da der Bürgermeister sowohl Chef der Gemeindeverwaltung ist als auch den Vorsitz im Kommunalparlament hat und hier in der Rolle eines Gemeindevertreters auftritt. Das Attribut *Bürgermeister* weist somit auf den Gemeindevertreter, der den Vorsitz im Gemeinderat innehat.

Das aus dem Graph generierte UML-Modell bietet einen ersten Ansatz für ein Klassenmodell des zu entwickelnden Ratsinformationssystems. Die aufgetretenen Fehler, wie leere Klassen und überflüssige Beziehungen, sind nachvollziehbar, da als Grundlage für das Modell der Inhalt des Deutschen Wortschatzes, also öffentlich zugängliche Quellen, und keine Spezifikation aus dem entsprechenden Kontext verwendet wurde. Für ein Modell, dass den Anforderungen eines Kommunalpolitikers entspricht, müssen weitere Informationen, wie Zugangsberechtigungen, Sitzungsdokumente o. ä. mit abgebildet werden.

6. Stand des Projekts – Ausblick – Bezug zu SOA

Anhand eines Beispiels konnte der Ablauf des Algorithmus' demonstriert werden. Es verdeutlicht, dass theoretisch durch die Kopplung von Verfahren zur statistisch-linguistischen Analyse und Heuristiken UML-Klassendiagramme aus natürlichsprachlichen Dokumenten erzeugt werden können. Die Umsetzbarkeit dieses theoretischen Ansatzes ist durch eine prototypische Implementierung nachzuweisen.

Darüber hinaus wird zu zeigen sein, dass dieses Verfahren auch für umfangreiche Dokumente in akzeptabler Zeit Ergebnisse liefert, mit denen ein fachfremder Modellentwickler schnell einen Überblick über den Fachraum gewinnt und mit den Fachexperten die nötigen Änderungen durchführen kann. Insbesondere sind die Höhe der Signifikanzwerte H_S und H_K sowie die Gültigkeit der Heuristiken für Fachtexte zu prüfen.

Anschließend ist zu untersuchen, inwiefern sich Beziehungen zwischen Klassenkandidaten mit Hilfe der Zitate, in denen sie vorkommen, als Aggregationen oder Spezialisierungen identifizieren lassen. Gleiche Überlegungen sind für Verhaltensdiagramme zu stellen. Die Frage ist, ob aus sich Begriffsbeziehungen (dann auch zu anderen Wortarten, wie Verben und Adjektiven) oder Zitaten allgemeingültige oder kontextspezifische Aussagen zu Prozessabläufen ableiten lassen.

Für die Spezifikation von Services müsste das Verfahren und Heuristiken erweitert werden, mit denen auch Operationen der Klassen und deren Reihenfolge hintereinander ermittelt werden könnte. Aus den Operationen lassen sich dann mit Hilfe von UML-Activity-Diagrams oder auch Ereignisgesteuerten Prozessketten Abläufe definieren, die als Services gekapselt werden können. [GeMa03]

Hierfür bietet das Wortschatztool bereits eine Auswertung, bei der signifikante rechte und linke Nachbarn angezeigt werden können. [FlIMM00] Außerdem könnten die Kookkurrenzen auf den Worttyp *Verb* beschränkt erste Ansätze für mögliche Operationen bilden.

Literaturverzeichnis

- [BKnM08] Beverungen, Daniel/ Knackstedt, Ralf/Müller, Oliver: Entwicklung Serviceorientierte Architekturen zur Integration von Produktion und Dienstleistung – Eine Konzeptionsmethode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte
in Wirtschaftsinformatik, Heft 3/2008, S. 220-234
- [Chom65] Chomsky Noam: Aspects of the Theory of Syntax, 1965.
- [EiPi07] Eisele, Markus/Pizka, Markus: SOA² Report – State of the Art Service-Orientierter Architekturen, in: <http://www.software-kompetenz.de>, v. 19.04.2007
- [FeLo03] Fettke, Peter/Loos, Peter: Model Driven Architecture (MDA).
in: Wirtschaftsinformatik. 45 (2003), Nr. 5, S. 555-559
- [FeSi01] Ferstl, Otto K./Sinz, Elmar J.: Grundlagen der Wirtschaftsinformatik : Band 1. 4., überarb. und erw. Auflage. München : Oldenbourg, 2001

- [FIKM00] Linguistische Aspekte des Projekts NIBA, Klagenfurt 2000.
- [FIMM00] Fliedl, Günther/Mayerthaler, W/Mayr, Heinrich C. et. al.: The NIBA Workflow – From textual requirements specifications to UML-Schemata, Klagenfurt 2000.
- [FIWe00] Fliedl, Günther/Werner, Günther: NIBA-TAG – A tool for analyzing and preparing german texts, 2000.
- [Fran03] Frankel, David S.: Model Driven ArchitectureTM : Applying MDATM to Enterprise Computing. 1. Auflage, Indianapolis, Wiley Publishing, Inc. 2003
- [Fuh96] Norbert Fuhr: Skriptum Information Retrieval, Lecture Notes on Information Retrieval, Universität Dortmund, 1996
- [GeMa03] Gerber, Stefan/Mai, André: Modellieren mit Begriffen - ein Vorgehensmodell zur Wiederverwendung von Ergebnissen der Analysephase im Design, in: Petrasch, Roland/Wiemers, Manuela/Kneuper, Ralf (Hrsg): Praxistauglichkeit von Vorgehensmodellen : 10. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e. V. (GI), Aachen, Shaker 2003
- [HeSi80] Herberger, Maximilian/Simon, Dieter: Wissenschaftstheorie für Juristen : Logik, Semiotik, Erfahrungswissenschaften, Frankfurt am Main, Metzner 1980
- [Herz05] Herz, Richard Alan: Stichwort SOA – Serviceorientierte Architekturen, WIAI-Stammtisch 2.11.2005
- [Heye02] Heyer, Gerhard/Quasthoff, Uwe et al.: Möglichkeiten und Verfahren zur autormatischen Gewinnung von Fachbegriffen aus Texten in: Bullinger, H.-J. (ed.) .Proc. Innovationsforum Content Management –Digitale Inhalte als Bausteine einer vernetzten Welt, Stuttgart 2002.
- [Jeck98] Jeckle, Mario: On formalising OOA heuristics – Bridging the gap between analysis and design, 1998
- [Jec+04a] Jeckle, Mario/Rupp, Chris/Zengler, Barbara et. al.: UML 2.0 – Neue Möglichkeiten und alte Probleme, in: Informatik Spektrum, Bd. 27 (2004), Heft 4, S. 323 – 331
- [Jec+04b] Jeckle, Mario ; Rupp, Chris ; Zengler, Barbara et. al.: UML 2 glasklar. 1. Auflage. München. Carl Hanser, 2004
- [JKSK00] Jungings, S./Kühn, H./Strobl, R./Karagiannis, D.: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation- Adonis: Konzeption und Anwendung, Wirtschaftsinformatik, 5, 2000.
- [Oest04] Oestereich, Bernd: Objektorientierte Softwareentwicklung. Analyse und Design mit der UML 2.0. 3. Auflage, München : Oldenbourg, 2004.
- [OMG00] OMG: Model Driven Architecture : White Paper Draft 3.2.
<ftp://ftp.omg.org/pub/docs/omg/00-11-05.pdf> 25.01.2005 - OMG
- [OMG01] OMG: Model Driven Architecture (MDA).
<http://www.omg.org/cgi-bin/doc?ormsc/2001-07-01> 25.01.2005 - OMG
- [Ortn00a] Ortner, Erich: Wissensmanagement, Teil 1: Rekonstruktion des Anwendungswissens, in: informatik spektrum, vom 23.04.2000.
- [Ortn00b] Ortner, Erich: Wissensmanagement, Teil 2: Systeme und Werkzeuge, in: informatik_spektrum vom 23.06.2000.
- [Part98] Partsch, Helmuth: Requirements-Engineering systematisch : Modellbildung für softwaregestützte Systeme., o. Auflage., Berlin. Springer, 1998.
- [Rump05] Rumpe, Bernhard: Agile Modellierung mit UML : Codegenerierung, Testfälle, Refactoring. 1. Auflage. Berlin : Springer, 2005
- [Rupp01a] Requirements Engineering und Management, Professionelle, iterative Anforderungsanalyse für die Praxis, Hanser 2001; www.sophist.de
- [Rupp01b] Requirements Engineering – der Einsatz einer natürlich-sprachlichen Methode bei der Ermittlung und Qualitätsprüfung von Anforderungen, in: Proceedings der Net.ObjectDays 2001, net.objectdays.org
- [Rupp03] Rupp, Chris: Wie aus Wünschen Anforderungen werden.
in: InformationWeek Nr. 22 vom 30. Oktober 2003, S. 24-25

- [RuSc04] Rupp, Chris/Schimpfky, Nicole: Erfahrungsbericht über einen Spezifikationsprozess mit einem bunten Reigen an Methoden und Notationen, in: Softwaretechnik-Trends, Band 24, Heft 4, November 2004
- [Sche98] Scheer, August-Wilhelm: ARIS – Vom Geschäftsprozess zum Anwendungssystem, Berlin, Heidelberg, New York, Tokio, Springer 1998.
- [Sun90] D. Sunday: A Very Fast Substring Search Algorithm. Communications of the ACM (CACM), Volume 33, Number 8: S. 132-142 (1990)
- [Wiem03] Wiemers, Manuela: Usability und Vorgehensmodelle, in: Petrasch, Roland/Wiemers, Manuela/Kneuper, Ralf (Hrsg): Praxistauglichkeit von Vorgehensmodellen – 10. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e. V., Aachen, Shaker 2003
- [WikiSOA] Begriff SOA in Wikipedia, abgerufen am 11.08.2008
- [Wort] Wortschatztool der Universität Leipzig, <http://www.wortschatz.uni-leipzig.de>, zuletzt abgerufen am 27.06.06

