

Gesellschaft für Informatik (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into the fields of

- Seminars
- Proceedings
- Dissertations
- Thematics

current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English.

Information: <http://www.gi-ev.de/service/publikationen/lni/>

ISSN 1617-5468

ISBN 978-3-88579-254-3

"This volume contains papers from the workshops of the Software Engineering 2010 conference held in Paderborn from February 22<sup>nd</sup> to 26<sup>th</sup> 2010. The topics covered in the papers range from enterprise engineering, software architecture, software-intensive embedded systems, software quality management, product line engineering to reports discussing social software engineering, grid workflows and software systems for public safety."



Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.):  
Software Engineering 2010 – Workshopband

160

# GI-Edition

## Lecture Notes in Informatics

**Gregor Engels, Markus Luckey,  
Alexander Pretschner,  
Ralf Reussner (Hrsg.)**

## Software Engineering 2010 – Workshopband

**(inkl. Doktorandensymposium)**

**22.–26. Februar 2010  
Paderborn**

# Proceedings





Gregor Engels, Markus Luckey,  
Alexander Pretschner, Ralf Reussner (Hrsg.)

## **Software Engineering 2010**

—

### **Workshopband**

(inkl. Doktorandensymposium)

**Fachtagung des GI-Fachbereichs Softwaretechnik**

**22.-26.02.2010**

**in Paderborn**

Gesellschaft für Informatik e.V. (GI)

**Lecture Notes in Informatics (LNI) - Proceedings**  
Series of the Gesellschaft für Informatik (GI)

Volume P-160

ISBN 978-3-88579-254-3  
ISSN 1617-5468

**Volume Editors**

Gregor Engels

Markus Luckey

Universität Paderborn  
Institut für Informatik  
Warburger Str. 100  
33098 Paderborn  
Email: {engels,luckey}@upb.de

Alexander Pretschner

TU Kaiserslautern  
Fachbereich Informatik  
Erwin-Schrödinger-Str.  
67663 Kaiserslautern  
Email: pretschner@cs.uni-kl.de

Ralf Reussner

Karlsruher Institut für Technologie  
Am Fasanengarten 5  
76131 Karlsruhe  
Email: reussner@kit.edu

**Series Editorial Board**

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

Hinrich Bonin, Leuphana-Universität Lüneburg, Germany

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, SAP Research, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität Berlin, Germany

Ulrich Furbach, Universität Koblenz, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr, München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Ernst W. Mayr, Technische Universität München, Germany

Sigrid Schubert, Universität Siegen, Germany

Martin Warnke, Leuphana-Universität Lüneburg, Germany

**Dissertations**

Dorothea Wagner, Karlsruhe Institute of Technology, Germany

**Seminars**

Reinhard Wilhelm, Universität des Saarlandes, Germany

**Thematics**

Andreas Oberweis, Karlsruhe Institute of Technology, Germany

© Gesellschaft für Informatik, Bonn 2010

**printed by** Köllen Druck+Verlag GmbH, Bonn



## **Vorwort**

Die Tagung "Software Engineering 2010" (SE 2010) wurde vom 22. bis 26. Februar 2010 als sechste Veranstaltung einer inzwischen etablierten Reihe von Fachtagungen durchgeführt, deren Ziel die Zusammenführung und Stärkung der deutschsprachigen Softwaretechnik ist. Die SE 2010 bot ein Forum zum intensiven Austausch über praktische Erfahrungen, wissenschaftliche Erkenntnisse sowie zukünftige Herausforderungen bei der Entwicklung von Softwareprodukten bzw. Software-intensiven Systemen. Sie richtete sich gleichermaßen an Teilnehmer aus Industrie und Wissenschaft. Der Tagungsband wurde als LNI-Band 159 veröffentlicht.

Die Software Engineering-Tagungsreihe wird vom Fachbereich Softwaretechnik der Gesellschaft für Informatik e.V. getragen. Die SE 2010 wurde vom Lehrstuhl Datenbanken und Informationssysteme der Universität Paderborn veranstaltet.

Ein wichtiger Bestandteil der SE 2010 waren 8 Workshops zu innovativen und praxisrelevanten Themen im Software Engineering, die auf großes Interesse gestoßen sind. Die Workshops deckten thematisch das Spektrum vom Enterprise-Engineering bis hin zur Diskussion der zukünftigen Entwicklung Software-intensiver, eingebetteter Systeme umfassend ab. Hinzu kamen Workshops zu innovativen Themen in den Bereichen Qualitätsmanagement, Produktlinienentwicklung, Social Software Engineering, Grid Workflows, Enterprise Architecture Management und der softwaretechnischen Unterstützung der zivilen Sicherheit. Der vorliegende Workshopband wurde im Anschluss an die Tagung erstellt, sodass Autoren die Möglichkeit hatten, Anregungen aus Diskussionen aus den Workshops aufzunehmen. Er umfasst alle Workshops mit Ausnahme des Workshops Software-Qualitätsmodellierung und -bewertung (SQMB), der separate Proceedings vorlegt.

Zum ersten Mal wurde im Jahr 2010 ein Doktorandensymposium auf der Software Engineering-Tagung veranstaltet. Die vielversprechenden und interessanten Themenvorschläge der eingeladenen jungen Forscher sind ebenfalls in diesem Band abgedruckt.

Unser Dank geht an die Organisatoren der einzelnen Workshops für ihr besonderes Engagement sowie an Markus Luckey, der die Zusammenstellung der Beiträge für diesen Band übernommen hat. Darüber hinaus danken wir dem Organisationsteam für die Unterstützung bei der Planung und Durchführung der Workshops.

Wir wünschen Ihnen viele neue Einsichten und Erkenntnisse bei der Lektüre des vorliegenden Workshop- und Doktorandensymposiumbandes!

Ralf Reussner, Leitung Workshops und Tutorials  
Alexander Pretschner, Leitung Doktorandensymposium  
Gregor Engels, Tagungsleitung

Karlsruhe, Kaiserslautern und Paderborn, im April 2010



### **Tagungsleitung**

Gregor Engels, Universität Paderborn

### **Leitung Industrietag**

Wilhelm Schäfer, Universität Paderborn

### **Leitung Workshops und Tutorials**

Ralf Reussner, Karlsruher Institut für Technologie

### **Leitung Doktorandensymposium**

Alexander Pretschner, TU Kaiserslautern

### **Leitung SE FIT**

Stefan Sauer, Universität Paderborn

### **Tagungsorganisation**

Markus Luckey, Universität Paderborn

Friedhelm Wegener, Universität Paderborn

Beatrix Wiechers, Universität Paderborn

### **Programmkomitee**

Klaus Beetz, Siemens AG

Manfred Broy, TU München

Bernd Brügge, TU München

Jürgen Belz, Hella KGaA Hueck & Co.

Jürgen Ebert, Universität Koblenz-Landau

Martin Glinz, Universität Zürich

Michael Goedicke, Universität Duisburg-Essen

Klaus Grimm, Daimler AG

Volker Gruhn, Universität Leipzig

Wilhelm Hasselbring, Christian-Albrechts-Universität zu Kiel

Stefan Jähnichen, TU Berlin

Matthias Jarke, RWTH Aachen

Gerti Kappel, TU Wien

Udo Kelter, Universität Siegen

Roger Kilian-Kehr, SAP AG

Claus Lewerentz, BTU Cottbus

Horst Lichter, RWTH Aachen

Peter Liggesmeyer, TU Kaiserslautern

Oliver Mäkel, Siemens AG

Florian Matthes, TU München

Barbara Paech, Universität Heidelberg

Klaus Pohl, Universität Duisburg-Essen

Alexander Pretschner, TU Kaiserslautern

Andreas Rausch, TU Clausthal

Ralf Reussner, Karlsruher Institut für Technologie

Bernhard Rumpe, RWTH Aachen

Eric Sax, MBtech Group  
Wilhelm Schäfer, Universität Paderborn  
Andy Schürr, TU Darmstadt  
Rainer Singvogel, msg systems AG  
Markus Voß, Capgemini sd&m AG  
Andreas Winter, Universität Oldenburg  
Mario Winter, Fachhochschule Köln  
Heinz Züllighoven, Universität Hamburg  
Albert Zündorf, Universität Kassel

**Offizieller Veranstalter**

Fachbereich Softwaretechnik der Gesellschaft für Informatik (GI)

**Mitveranstalter**

s-lab – Software Quality Lab, Paderborn  
Universität Paderborn

**Unterstützt wird die Tagung zudem von**

Schweizer Informatik Gesellschaft (SI)  
Österreichische Computer Gesellschaft (OCG)

## Sponsoren

---

### SE 2010 Goldsponsoren



---

### SE 2010 Silbersponsoren



---

### SE 2010 Unterstützer



UNIVERSITÄT PADERBORN  
Die Universität der Informationsgesellschaft



# Inhaltsverzeichnis

## Enterprise Engineering meets Software Engineering (E2mSE)

Geschäftsregelmanagement, Geschäftsprozessmodellierung und Unternehmensmodelle <i>Jürgen Pitschke</i> .....	19
Comparison of business processes and agile software development processes <i>Stefan Eicker, Tim Gottschalk, Annett Nagel</i> .....	27
Ein begriffsbasierter Ansatz zur semantischen Extraktion von Datenbankschemata <i>Henri Mühle, Hannes Voigt, Wolfgang Lehner</i> .....	35
Unterstützung von BPMN durch Lösungen der SAP <i>Sascha Alber</i> .....	43
Optimierung der Modellierungsergebnisse im Bereich interaktiver Unternehmensapplikationen durch sprachkritische Generierung des Domainenwissens <i>Joachim Sternhuber</i> .....	51
Modellierung von Kennzahlensystemen mit BPMN <i>Nicole Zeise</i> .....	63
Interaktive Konfigurierung dynamischer Anwendungssysteme aus Komponenten <i>Tobias Grollius</i> .....	75

## Erster Workshop zur Zukunft der Entwicklung softwareintensiver, eingebetteter Systeme (ENVISION2020)

ENVISION 2020: Erster Workshop zur Zukunft der Entwicklung softwareintensiver, eingebetteter Systeme <i>Kim Lauenroth, Ernst Sikora</i> .....	87
Model-based Development Tools for Embedded Systems in the Industry – Results from an Empirical Investigation – <i>Markus Herrmannsdoerfer, Thomas Kofler, Stefano Merenda, Daniel Ratiu, Judith Thyssen</i> .....	93
[(Spacecraft Bus Controller) + (Automotive ECU)] / 2 = UltimateController <i>Sergio Montenegro, Frank Dannemann, Lutz Dittrich, Benjamin Vogel, Ulf Noyer, Jan Gacnik, Marco Hannibal, Andreas Richter, Frank Köster</i> .....	103

Ansätze zum Nachweis der Gleichwertigkeit von Software-Komponenten <i>Hans-Werner Wiesbrock</i> .....	115
Evaluation eines modellbasierten Requirements-Engineering-Ansatzes für den Einsatz in der Motorsteuerungs-Domäne <i>Ernst Sikora, Klaus Pohl</i> .....	127
A System for Seamless Abstraction Layers for Model-based Development of Embedded Software <i>Judith Thyssen, Daniel Ratiu, Wolfgang Schwitzer, Alexander Harhurin, Martin Feilkas, Eike Thaden</i> .....	137
Eine erweiterte Systemmodellierung zur Entwicklung von softwareintensiven Anwendungen in der Automobilindustrie <i>Jörg Holtmann, Jan Meyer, Wilhelm Schäfer, Ulrich Nickel</i> .....	149

## **Grid Workflow Workshop (GWW)**

3. Grid Workflow Workshop (GWW 2010) <i>Wilhelm Hasselbring, André Brinkmann</i> .....	161
Dynamische Prozesse und deren Abbildung auf ausführbare Workflows <i>Andreas Hoheisel, Michael Fellmann, Thorsten Dollmann</i> .....	163
Semantisch beschriebene Grid-Dienste in einem Semantik Webservice Framework <i>Dirk Mühlenberg, Sandro Leuchter</i> .....	169
Grid-Workflows in Molecular Science <i>Georg Birkenheuer, Sebastian Breuers, André Brinkmann, Dirk Blunk, Gregor Fels, Sandra Gesing, Sonja Herres-Pawlis, Oliver Kohlbacher, Jens Krüger, Lars Packschies</i> .....	177
Workflow Modeling for WS-BPEL-based Service Orchestration in SMEs <i>S. Gudenkauf, G. Scherp, W. Hasselbring, A. Höing, O. Kao</i> .....	185
Security, Fault Tolerance and Modeling of GridWorkflows in BPEL4Grid <i>Ernst Juhnke, Tim Dörnemann, Roland Schwarzkopf, Bernd Freisleben</i> .....	193
Ein modellgetriebener Ansatz zur Nutzung von WS-BPEL für Scientific Workflows <i>G. Scherp, W. Hasselbring</i> .....	201
The Missing Features of Workflow Systems for Scientific Computations <i>Mirko Sonntag, Dimka Karastoyanova, Frank Leymann</i> .....	209

## **2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM)**

Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes <i>Helge Buckow, Hans-Jürgen Groß, Gunther Piller, Karl Prott, Johannes Willkomm, Alfred Zimmermann</i> .....	219
A Method for Integrating EAM and BPM <i>Christoph Moser, Daniel Fürstenau, Stefan Junginger</i> .....	231
Interrelating Concerns in EA Documentation – Towards a Conceptual Framework of Relationships <i>Sabine Buckl, Florian Matthes, Christian M. Schweda</i> .....	243
Roadmaps for Enterprise Architecture Evolution <i>Alexander M. Ernst, Alexander W. Schneider</i> .....	253
EA Management Patterns for Future State Design <i>Chris Aitken</i> .....	267
I-Pattern for Gap Analysis <i>Philipp Gringel, Matthias Postina</i> .....	281
Business Driven SOA-Service Candidate Identification <i>Martin Weber, Markus Krieger</i> .....	293
Teaching Enterprise Architecture Management with Student Mini-Projects <i>Sabine Buckl, Alexander M. Ernst, Florian Matthes, Christian M. Schweda</i> .....	309

## **Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation (PIK2010)**

PIK2010 Workshop-Einleitung und Überblick <i>Andreas Birk, Klaus Schmid, Markus Völter</i> .....	323
Herausforderungen für Variantenmanagement in Anforderungsdokumenten <i>Ekaterina Boutkova</i> .....	329
Werkzeuge zur modellgetriebenen Entwicklung von Produktlinien: Ein Erfahrungsbericht am Beispiel von Versionskontrollsystemen <i>Thomas Buchmann, Alexander Dotor, Bernhard Westfechtel</i> .....	339

Projectional Language Workbenches as a Foundation for Product Line Engineering  
*Markus Voelter* ..... 353

Verknüpfung von kombinatorischem Plattform- und individuellem Produkt-Test für  
Software-Produktlinien  
*Andreas Wübbeke, Sebastian Oster* ..... 361

## **Innovative Systeme zur Unterstützung der zivilen Sicherheit (Public Safety)**

Workshop 'Innovative Systeme zur Unterstützung der zivilen Sicherheit: Architekturen  
und Gestaltungskonzepte' – Geleitwort  
*Rainer Koch* ..... 375

Smarte Systeme und Services für Freiheit, Recht und Ordnung  
*Margarete Donovan-Kuhlisch* ..... 377

Ein Konzept zur organisationsübergreifenden Integration von IT-Systemen für die zivile  
Sicherheit  
*Wolf Engelbach, Heiko Roßnagel, Sandra Frings* ..... 379

Rahmenbedingungen für Informationssysteme im Kontext der nicht-polizeilichen  
Gefahrenabwehr  
*Marco Plaß, Christina Schäfer* ..... 387

Architektur eines prozessunterstützenden Softwaresystems für den Rettungsdiensteinsatz  
bei einem Massenansturm von Verletzten  
*Carsten Wirth, Dirk Roscher, Paul Zernicke,  
Martin Schultz, Christine Carius-Düffel* ..... 397

Systemkonzept für ein Mobiles Informationssystem zur Prozessoptimierung in  
Feuerwehren und öffentlichen Verwaltungen  
*Anna Maria Japs, Stephan Prödel* ..... 405

Das Paradigma der Event-Driven Architecture als Grundlage für ein  
entscheidungsunterstützendes IT-System im Zivil- und Katastrophenschutz  
*Robin Marterer, Benedikt Birkhäuser, Rainer Koch* ..... 407

Die praxisorientierte Perspektive der Anforderungsanalyse aus Sicht der Endanwender  
*Jörg Rhode* ..... 415

Mobile Dienste für die Einbindung von Bürgern und Hilfeinsatzkräften  
*Heiko Roßnagel* ..... 425

Modellierung von Prozessen in der Feuerwehrdomäne zur Identifikation von Informationsbedarfen <i>Christian Lindemann, Stephan Prödel, Rainer Koch</i> .....	433
--	-----

### **3rd International Workshop on Social Software Engineering (SSE2010)**

3rd International Workshop on Social Software Engineering <i>Martin Ebner, Imed Hammouda, Hans-Jörg Happel, Walid Maalej, Wolfgang Reinhardt</i> .....	445
Towards Social Information Systems <i>Marc Quast, Jean-Marie Favre</i> .....	449
weHelp: A Reference Architecture for Social Recommender Systems <i>Swapneel Sheth, Nipun Arora, Christian Murphy, Gail Kaiser</i> .....	461
Switch! Recommending Artifacts Needed Next Based on Personal and Shared Context <i>Alexander Sahn, Walid Maalej</i> .....	473
Adding Semantics to Social Software Engineering: (Re-)Using Ontologies in a Community-oriented Requirements Engineering Environment <i>Steffen Lohmann, Thomas Riechert</i> .....	485
Diagnosing Inconsistent Requirements Preferences in Distributed Software Projects <i>A. Felfernig, M. Schubert, M. Mandl, P. Ghirardini</i> .....	495
User-Driven Requirements Engineering for Mobile Social Software <i>Norbert Seyff, Florian Graf</i> .....	503

### **Doktorandensymposium**

Doktorandensymposium der SE 2010 <i>Alexander Pretschner</i> .....	514
Variantenmanagement in Anforderungsdokumenten <i>Ekaterina Boutkova</i> .....	517
Ein Ansatz zur Entwicklung operationalisierter Usability-Anforderungen <i>Holger Röder</i> .....	525
Design Decisions in Model-driven Software Development <i>Patrick Könemann</i> .....	531



Semi-automatic Matching of Heterogeneous Model-based Specifications <i>Konrad Voigt</i> .....	537
Product Certification of Component Performance Specifications <i>Henning Groenda</i> .....	543
Towards Information Flow Auditing in Workflows <i>Claus Wonnemann</i> .....	549

SE | 10  
SOFTWARE ENGINEERING

**Enterprise Engineering meets Software Engineering**  
**(E<sup>2</sup>mSE)**



# Geschäftsregelmanagement, Geschäftsprozessmodellierung und Unternehmensmodelle

Dr. Jürgen Pitschke

BCS – Dr. Jürgen Pitschke  
Bautzner Str. 79  
01099 Dresden  
jpitschke@enterprise-design.eu

**Abstract:** Die OMG hat eine Vielzahl von Standards für die Modellierung und Softwareentwicklung hervorgebracht. Neben den populären Standards UML und BPMN ist der Standard „Semantics of Business Vocabulary and Rules (SBVR)“ in den Fokus gerückt.

Der Beitrag stellt einen Ansatz vor, SBVR und BPMN-Modelle zu verknüpfen, um bessere Prozessmodelle als Grundlage für die Entwicklung von Unternehmensmodellen zu erstellen. Weiterhin wird die Verbindung von Geschäftsregelanalyse, Anforderungsanalyse, und Geschäftsprozessmodellierung diskutiert. Abschließend werden Möglichkeiten der Werkzeug-Unterstützung des Vorgehens aufgezeigt.

Die Trennung von Fachmodell und ausführbarem Modell ist dabei Grundgedanke des Modellierungsansatzes.

## 1 Semantics of Business Vocabulary and Rules (SBVR)

Der OMG Standard „Semantics of Business Vocabulary and Rules“ standardisiert ein Metamodell für die Entwicklung semantischer Modelle für die Darstellung eines Geschäftsvokabulars und von Geschäftsregeln. *Nicht* Gegenstand des Standards ist die Definition einer einzelnen Sprache oder Diagrammkonvention für die Darstellung von Vokabular und Regeln. Ebenso ist eine Methode nicht im Standard enthalten.

Verkürzt gesagt folgt der Standard dem „Mantra der Geschäftsregeln“: „Regeln basieren auf Fakten. Fakten bauen auf Konzepten auf. Konzepte werden durch Begriffe repräsentiert“. Dieses ist nicht völlig exakt, drückt aber den Zusammenhang zwischen den grundlegenden Konzepten – Konzept, Begriff, Fakttyp und Fakt, Regel – aus.

Wichtiges Grundprinzip ist die Trennung von Bedeutung (meaning) und Darstellung (expression).

Die Trennung von Bedeutung und Darstellung ist Grundlage der in SBVR definierten „Communities“. Verschiedene Communities verwenden verschiedene Begriffe für dasselbe Konzept. Unterschieden werden Sprach-Communities und Fach-Communities. Charakterisierende Eigenschaft einer Community ist der gemeinsame und einheitliche Gebrauch eines Vokabulars durch die Mitglieder der Community. Abbildung 1 zeigt das Metamodell eines Vokabulars nach SBVR.



Abbildung 1: Metamodell Vokabular nach [SBVR08]

Faktypen und Fakten sind nicht nur Grundlage für Geschäftsregeln, sondern werden auch in weiteren Modellen genutzt. Die Definition von Faktypen und Fakten basiert auf dem „Object-Role Modeling“ wie in [Halpin06] dargestellt. Halpin definiert „State of Affairs“ als Oberbegriff für „Events“ und „Situations“. „Events“ haben momentanen Charakter, während „Situations“ von einer bestimmten Dauer sind.

Faktypen etablieren Beziehungen zwischen Konzepten und können sowohl „Events“ als auch „Situations“ beschreiben. Z.B.

- Nutzer berichtet Incident. (Situation)  
Nutzer hat Incident berichtet. (Event)

Für unseren eigenen Modellierungsansatz bevorzugen wir die Darstellung von „Events“ durch Faktypen, um die Abgrenzung zu Geschäftsaktivitäten zu erleichtern.

Neben Faktypen wird in Unternehmensmodellen auch die Beschreibung von Fakten, d.h. die Beschreibung von Beziehungen zwischen individuellen Konzepten benötigt. Beispiel:

- Faktyp: Land benutzt Währung.
- Fakten: Deutschland benutzt Euro. Schweiz benutzt Schweizer Franken.

Faktenmodelle können sowohl textuell als auch grafisch dargestellt werden. Faktenmodelle sind Grundlage für die Definition von Ausdrücken die „Führung“ (Guidance) beschreiben. Abbildung 2 zeigt das Metamodell „Guidance“ des SBVR-Standards.



Abbildung 2: Metamodell Guidance nach [SBVR08]

Führung wird in Form von Geschäftsregeln und Empfehlungen beschrieben. Geschäftsregeln sind Regeln, die sich in der Verantwortung des „Geschäfts“ (des Unternehmens) befinden. Eine Geschäftsregel schränkt dabei immer einen Freiheitsgrad ein. Empfehlungen sind Klarstellungen, die gewünschtes Verhalten beschreiben.

## 2 Geschäftsregeln in natürlicher Sprache darstellen

Die Diskussion der Darstellung von Vokabular und Geschäftsregeln erfolgt oft vorrangig im Kontext von Rule-Engines. Schwerpunkt ist dabei die Implementation von Geschäftsregeln, die meist in Form von „If ... then ...“ Statements dargestellt werden.

Dabei werden mehrere Tatsachen übersehen:

- Bevor Regeln in einer Rule-Engine implementiert werden können, müssen sie im Fachmodell erfasst und systematisiert werden.
- Nicht alle Geschäftsregeln lassen sich in der „If ... then ...“-Form darstellen.
- Fachanwender kommunizieren Geschäftsregeln untereinander in vielen verschiedenen Formen, aber nur selten in Form von „If ... then ...“-Statements.
- Menschliche Kommunikation ist nicht immer logisch formalisierbar.

In [SBVR08] heißt es: “Business models, including the models that SBVR supports, describe businesses and not the IT systems that support them.” Für die Darstellung von Geschäftsregeln auf der Implementationsebene hat die Object Management Group (OMG) den Standard “Production Rules Representation (PRR)” geschaffen.

## 2.1 RuleSpeak® Deutsch – Geschäftsregeln in natürlicher deutscher Sprache darstellen

Im Anhang F des SBVR-Standards ist die RuleSpeak® - Notation für die Darstellung von Geschäftsregeln in natürlicher englischer Sprache enthalten. Dieser Ansatz wurde von Ron Ross entwickelt und gestattet die Darstellung von Geschäftsregeln in reglementierter, natürlicher Sprache. Schwerpunkt des Ansatzes ist dabei die Kommunikation zwischen Fachanwendern. Der Ansatz hat damit bis heute ein Alleinstellungsmerkmal. Die Abbildung auf andere Sichten (z.B. Geschäftsprozessmodell) und andere Perspektiven (z.B. Implementierung in einer Rule Engine) sind dabei unverzichtbar. RuleSpeak stellt dabei eine Menge von Best-Practices dar. RuleSpeak ist keine formale Sprache.

Der RuleSpeak®-Ansatz wurde vom Autor 2009 gemeinsam mit Ron Ross ins Deutsche übertragen. Damit steht ein Ansatz zur Darstellung von Geschäftsregeln in natürlicher deutscher Sprache zur Verfügung. Um das Management der Geschäftsregeln zu ermöglichen, wird die Konstruktion der Geschäftsregeln reglementiert.

Grundaussagen des Ansatzes sind:

- Jede Regel muss ein Regelschlüsselwort enthalten.
- Regelschlüsselworte sind:
  - „muss“
  - „darf nicht“
  - „nur“ (nur, wenn).
- Jede Regel muss mit dem Regelsubjekt beginnen.
- Jede Empfehlung muss ein Empfehlungsschlüsselwort enthalten.
- Empfehlungsschlüsselworte sind:
  - „kann“
  - „braucht nicht“

Beispiele:

- Ursprungsregel: *Für eine Bestellung ist es unbedingt erforderlich, den Kunden anzuzeigen, der sie platziert hat.*  
Revidierte Regel: *Eine Bestellung muss den Kunden anzeigen, der sie platziert hat.*
- Regel: *Ein Kunde darf eine Bestellung nur platzieren, wenn dieser Kunde ein Konto besitzt.*
- Empfehlung: *Eine Kreditprüfung braucht nicht ausgeführt werden, wenn der Bestellwert weniger als 1.000 Euro beträgt.*

Ein Geschäftsregelausdruck muss der Form der Fakten folgen.

Beispiel:

- Fakten: Kunde platziert Bestellung  
Kunde besitzt Konto
- Regel: Ein Kunde darf eine Bestellung nur platzieren, wenn der Kunde ein Konto besitzt.

## 2.2 Geschäftsregelanalyse mit RuleSpeak®

Neben den Basis-Richtlinien existieren weitere Vorgaben für die Darstellung von Geschäftsregeln. Regeln sollen keine eingebetteten Regeln beinhalten, die Wiederverwendung von Regelbestandteilen wird unterstützt, usw. Folgt man diesen Richtlinien entsteht eine Methode zur Darstellung und Verwaltung von Geschäftsregeln. Die schrittweise Formalisierung der Regeln entsprechend der Richtlinien stellt zugleich eine Anleitung für die textuelle Analyse von ungeordneten Informationen dar. Durch die Anwendung der Richtlinien erfolgt eine *Normalisierung* von Regeln. Durch diese Normalisierung ist eine heuristische Überprüfung der Konsistenz der Regelmenge möglich. Wichtige Kriterien sind z.B. Widerspruchsfreiheit und Überlappung von Regeln.

Um Geschäftsregeln in anderen Modellen nutzbar zu machen, werden weitere Mechanismen benötigt. Neben den im SBVR-Standard definierten „Communities“ ist die Strukturierung der Regelmenge durch die Definition von Regelgruppen ein wichtiger Aspekt.

Mit dem Produkt „RuleXpress“ der Firma RuleArts steht ein Werkzeug zur Verfügung, dass die praktische Umsetzung des RuleSpeak-Ansatzes unterstützt.

## 3 Geschäftsregeln und Geschäftsprozessmodelle

Vokabular und Geschäftsregeln sind essentiell für die Gestaltung von Unternehmensmodellen. Besonders eng sind Geschäftsregeln mit Geschäftsprozessmodellen verbunden.

Abbildungen 3 und 4 zeigen Beziehungen zwischen verschiedenen Artefakten im Modellierungsprozess.



Abbildung 3: Relationen zwischen Artefakten im Unternehmensmodell - I



Bevor wir die Zusammenhänge zwischen Geschäftsregeln und Geschäftsaktivitäten analysieren können, müssen wir sicher stellen, dass Prozessmodell und Regelbasis dasselbe Vokabular benutzen.

Eine in der Literatur übliche Benennungsmethode für Geschäftsaktivitäten ist die Verwendung von <<Verb>> <<Substantiv>> bzw. <<Substantiv>><<Verb>>-Kombinationen. Die Substantive präsentieren Begriffe, die im Vokabular definiert sein müssen. Die verwendeten Verben müssen als Faktensymbol definiert sein. Hier ist die gängige Namenskonvention zu hinterfragen. Z.B. wir kennen einen Fakttyp „Ware ist verladen in Container“. Entsprechend der gängigen Praxis würde die Aktivität im Prozess mit „Ware verladen“ benannt. Das reflektiert den Fakttyp jedoch unvollständig. Korrekt wäre die Bezeichnung „Ware in Container verladen“. Ergebnis der Aktivität ist der Fakt „Ware ist in Container verladen“. Wir ändern daher die Namenskonvention für die Benennung von Aktivitäten zu „Die Benennung einer Geschäftsaktivität muss den zugrunde liegenden Fakttyp widerspiegeln.“

### 3.1 Geschäftsregeln und Geschäftsprozessmodelle

Geschäftsregeln und Geschäftsprozesse sind zwei Seiten derselben Medaille. Geschäftsregeln detaillieren Aktivitäten und Elemente des Geschäftsprozesses, Geschäftsprozessmodelle ohne Geschäftsregeln sind unvollständig.

Es ist wichtig, Geschäftsprozesse und Geschäftsregeln unabhängig voneinander zu analysieren und darzustellen, da es sich um unabhängige Konzepte handelt. Prozesse sind vergleichsweise stabil, Regeln ändern sich schneller und öfter als Prozesse. Zugleich sind beide eng verknüpft. Abbildung 4 zeigt die Beziehungen zwischen Regeln, Regelgruppen und anderen Artefakten eines Unternehmensmodells.

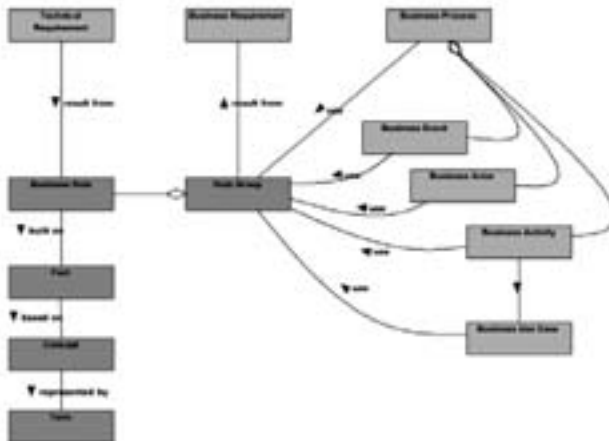


Abbildung 4: Relationen zwischen Artefakten im Unternehmensmodell - II

Eine systematische Betrachtung der Verbindung von Geschäftsregeln und Elementen im Geschäftsprozessmodell resultiert oft in erheblich verbesserten Prozessmodellen. Oft denken Prozessmodellierer in einem ersten Ansatz an die Gleichung „Gateway == Regel“. Das ist vollständig falsch und resultiert aus einem falschen Verständnis des BPMN-Elements „Gateway“ und einer oberflächlichen Betrachtung von Geschäftsregeln.

Abbildung 5 zeigt einen Ausschnitt eines in der Praxis typischen Prozessmodells.

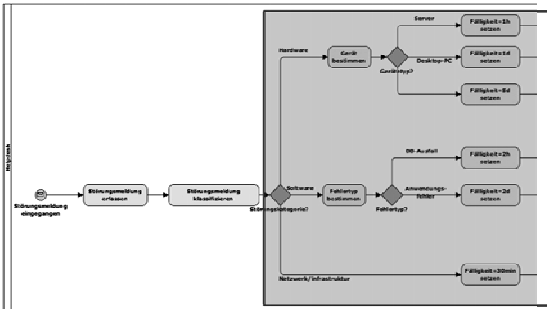


Abbildung 5: Ausschnitt eines Prozessmodells

Der blau markierte Teil des Prozessmodells stellt einen Entscheidungsbaum zur Bestimmung der Fälligkeit einer Störungsmeldung dar, nicht jedoch einen Geschäftsprozess. Änderungen in der *Regelmenge* zur Bestimmung der Fälligkeit führen zu einer Änderung des Prozessmodells. Abbildung 6 zeigt den korrekten Geschäftsprozess.



Abbildung 6: Verbessertes Prozessmodell

Um diesen Ansatz praktisch umsetzen zu können, benötigen wir Werkzeuge, die die Verwendung des Vokabulars in Prozessmodellen und die Verknüpfung von Regeln bzw. Regelmengen mit Aktivitäten zulassen. Eine Möglichkeit ist hier die Integration von RuleXpress und Visual Paradigm.

Abbildung 4 zeigt weitere Verknüpfungen zwischen Modellartefakten. Eine konsequente Trennung der einzelnen Konzepte resultiert in stabileren und besseren Modellen.

## 4 Geschäftsregeln, Geschäftsprozessmodelle, Unternehmensmodelle

Vokabular, Faktenmodell, Geschäftsregeln und Geschäftsprozessmodell sind wichtige Bestandteile jedes Unternehmensmodells. Natürlich existieren weitere essentielle Sichten und Perspektiven auf ein Unternehmen. Hier helfen Frameworks wie das Zachman-Framework [Zach01] relevante Sichten und Perspektiven zu identifizieren und zu organisieren.

Die OMG hat weitere Standards sowohl für die Implementation (z.B. UML, PRR) als auch die Fachmodellierung (z.B. BMM) freigegeben. Die Zusammenhänge zwischen den Standards rücken immer mehr in den Fokus der Standardisierung. Abbildung 7 ordnet verschiedene Standards in eine Unternehmensarchitektur ein.

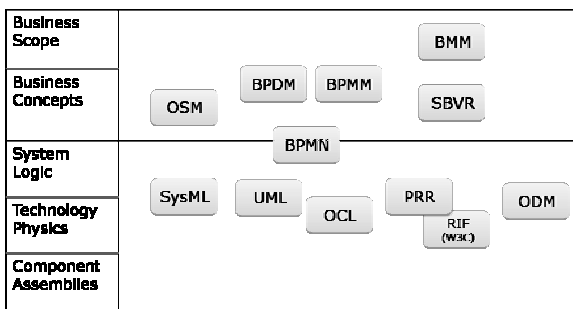


Abbildung 7: Standards für die Unternehmensmodellierung

## Literaturverzeichnis

- [BPMN09] Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0, OMG, August 2009, OMG Document Number: dtc/2009-08-14
- [Halp06] Halpin, Terry, Objectification of Relationships in Advanced Topics in Database Research, Volume 5, Seiten 106-123, Idea Group Pub, 2006
- [Ross09] Ross, Ronald G. , Business Rule Concepts, Third Edition, Business Rule Solutions, LLC., 2009
- [Pit09A] Pitschke, Jürgen, RuleSpeak® Satzformen, Business Rules in natürlich sprachlichem Deutsch spezifizieren, 2009
- [Pit09B] Pitschke, Jürgen, RuleSpeak® Guidelines- Grundlagen, Geschäftsregeln in natürlich sprachlichem Deutsch darstellen – was man tun sollte und was nicht, 2009
- [Pit09C] Pitschke, Jürgen, RuleSpeak®, Kommentare zu den Basisdokumenten, Version 1.2
- [SBVR08] Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, OMG Document Number: formal/2008-01-02
- [Zach01] Zachman, John, The Zachman Framework for Enterprise Design: Primer for Enterprise Design and Manufacturing, Zachman Int., 2001

# Comparison of business processes and agile software development processes

Stefan Eicker, Tim Gottschalk, Annett Nagel

Institute for Computer Science and Business Information Systems  
University of Duisburg-Essen  
Universitätsstraße 9  
45141 Essen

{Stefan.Eicker; Tim.Gottschalk; Annett.Nagel}@icb.uni-due.de

**Abstract:** Well-defined software development processes are rarely established. This especially is true for agile software development processes, which become more and more popular. The paper evaluates how far agile software development processes can be seen as a kind of business processes. This forms the basis for the selective transferring of methods and tools from the field of business processes to the field of software development processes.

## 1 Introduction

Software development is a process embedded within a dynamic or even chaotic environment. Additionally the software as the core object of software development becomes more and more complex and comprehensive (Ha07, p.89f). Providing success of software development projects, enterprises have to optimize the development processes by using effective software development tools on the one hand and agile approaches on the other hand (MR08, p. 1). The first step is the integration of agile software development processes within the organization. Those agile processes do not only target a higher quality of the software itself, but also a higher effectiveness and efficiency of the development teams. Furthermore agile software development processes provide approaches to cope with increasing complexity and dynamic of different kinds of software development projects. These approaches are based on agile management and method frameworks (LM04, p. 29ff).

The paper evaluates how far agile software development processes can be seen as a kind of business processes. This evaluation forms the prerequisite for the transfer of methods and tools from the field of business processes to the field of agile software development processes. Scrum and eXtreme Programming (XP) are exemplary chosen for the presentation of agile software development processes as they are very popular and often used in the field of agile software development (Ve08, p. 5). The research method used is a combination of an analysis of related literature and argumentative-deductive reasoning.

The goal of using those methods is the warranty of a systematic analysis of commonness's and differences concerning the fields of business processes and software development processes. On this base methods can wittingly be transferred from one field to the other.

The software development process can be separated into three different levels: The first one is the project management level, the second one the process level and the third one the level of practices and activities. Each agile approach addresses one or more of these levels (ASR02, p. 95). Also different approaches addressing different levels can be combined to an overall approach for all levels. An example is XP@Scrum, where the approaches of Scrum for the levels of (project) management and process (process definition) (1st and 2nd level) are combined with practices and activities defined by XP for the 3rd level.

## 2 Added value in software development processes

Porter has strongly differentiated between primary activities and supporting activities concerning the added value of an organization (Po00, p. 1ff). His approach has to be extended in the context of software development as there are additional secondary activities (cf. Figure 1). The primary activities are corresponding to the actual software development process and therefore to the added value. The secondary activities on the one hand are supporting the primary added value, but on the other hand they are even a part of it. Thus the supporting activities not only provide a service to the primary but also to the secondary activities - not creating a part of the added value directly.

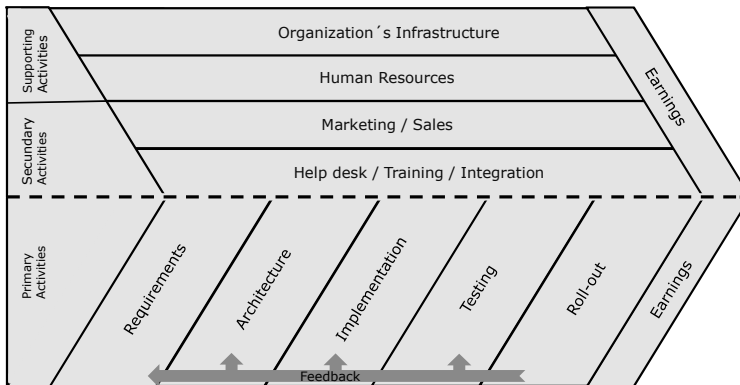


Figure 1: Added value of software development processes (According to Po00)

Therefore the actual development of the software within the development process is the primary activity. Secondary activities are the help desk, training of the customers, support or activities by the sales department by analyzing the market in form of studies and at the end sell them or using the results within the own organization. Those activities are supported by activities of the departments of infrastructure or human resources.

The structure of the activities and business processes of a software development organization therefore is comparable to those of a classic organization: The processes are separated into primary and secondary business processes supported by standardized processes (e.g. human resources). The actual development - the primary business processes - instead is individual in every software development project. That is useful as the development process can fulfill the specific requirements and conditions of each individual project. This aspect - the flexibility of processes - is called "Tailoring" and is the crucial factor of success within a dynamic environment since a non-dynamic process model is pointless in any kind of software development processes (MSV97, p. 1f).

### **3 Aspects of the business process management-oriented organization in the context of agile software development**

Schmelzer and Sesselmann specified an organization based on business processes upon eight aspects (SS08, p. 83). These aspects can also be found in the basic approaches of agile software development processes. Principally the agile processes are based on the approaches of the agile manifest (Ma01): Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. The approaches of the agile manifest are obviously an antagonism to the basic process orientation respectively process organization. The following presentation of the core aspects in the context of agile software development indicates, that the structure of agile processes is explicitly based on the same approaches as business processes are. The aspects are referenced to the three levels of the software development process in the context of agile software development.

#### **3.1 Stronger customer orientation**

One of the main goals of an organization is to solve the problems of their customers by providing solutions, i.e. products or services, to fulfill their needs and finally satisfy the customers. Depending on their grade of satisfaction customers buy the products or services and therewith the consistency of an organization can be assured.

Using the mechanisms and methods of business process management (BPM) those aspects become more important. Especially in the field of software development the focus is on the requirements of the customers and the ability to react quickly and flexible to changes of requirements. The term "customer" is general for a person, an organizational unit or an organization who/which ordered the development of software. The aspect of a stronger customer orientation is being accounted in the different agile development approaches in miscellaneous ways.

On the management and process level Scrum defines a product owner who is part of the development team. He always is available in case of questions or ambiguities on the part of the software developers. (Pi08, p. 9; Sc07, p. 114) On the level of practices and activities XP defines different approaches for this aspect.

E.g. the on-site customer is similar to the product owner defined by Scrum, the planning game as well as short releases. The customer is directly involved in many of the activities which results in a high customer orientation.

### **3.2 Less interfaces**

One of the core elements of business process management is the reduction of interfaces within an organization. This is done by using a process-oriented organization instead of a function-oriented one. Focusing the process-oriented organization means that an organization is no longer diverted into vertical functions where everybody fulfills only one task without knowledge what goes on behind or before his working station. So there won't be bottlenecks anymore and everyone has the final product or service in mind.

The reduction of interfaces actually relates to a function-oriented organization. The basic thought of agile development is that the development team iterates through all process steps several times until the software is finished. Due to that fact that there is e.g. no dedicated requirements manager or architect there is no function-oriented but a process-oriented organization. Thus there are less interfaces than there would be if the development process would be organized in a function-oriented manner.

Furthermore this aspect has to be seen in two different contexts: On the one side the interfaces apply to the different tools used in the process of software development like the integrated development environment (IDE) or a source code management system. On the other hand interfaces between the stakeholders are referenced here. The amount of interfaces between the heterogeneous systems is a question of the integration scale.

Scrum as an approach on the management and process level defines a daily scrum meeting. This meeting is compulsive and ensures that information is transported directly from the source to the target and not passing any interstations in form of other people which generally distort information. Therefore on the level of practices and activities the agile development approaches require a physical adjacency of the workplaces of the developers and the customer representative (product owner / on-site customer). Together with the required daily meetings this ensures a face to face communication which is the most efficient way of communicating for human beings (VS07, p. 9).

### **3.3 Less coordination effort**

In the context of process-oriented organizations the existence of coordination should be rather less. Not only there are fewer interfaces needed but also communication is much more efficient, because everybody knows the whole process and therefore knows who and when to ask for special concerns.

This aspect deals with the necessary coordination in processes. Agile software development approaches are based on a self coordinating team.

The teams run through the iterations several times and thus a concrete coordination is only necessary for a short period of time which is on the one side easier and requires less effort than coordinating a long period on the other side, e.g. due to less complexity. This aspect focuses the management and the process level of the agile development approach. Regarding the practices and activities level this aspect takes care of the fact that teams plan and discuss the next iteration together with the customer and thus the teams coordinating themselves. Another aspect on this level is that there is not more than one customer representative supposed to be. The teams exactly have one contact person. Hence on the side of the customer there is no necessity for coordination between several contact persons. These practices and activities reduce the coordination effort of the agile software development process (Pi08, p. 17f).

### **3.4 Clear responsibilities for the process result**

The result of a process is based on the working results of all involved people. The goal of business process management is that every one of them shows responsibility and interest in the final product or service. Therefore everybody involved has to reach a certain stage of quality to satisfy the customer and even shows a responsibility for this.

In the context of agile software development the whole team is responsible for the result of the process - in this case the developed software. On the management and process level this means that the members of the team act like a team e.g., they support each other. This aspect requires team building which is a process of learning. It is described in detail under the aspect “organizational learning”. Due to matter of fact that they are planning their activities themselves as well as how they accomplish these activities for the entire process they are responsible for the output. On the practice and activity level the practice of collective ownership is defined (LRW02, p. 94 ff).

### **3.5 Systematic increase of the process performance**

This aspect can be inherited from the previous ones. As the involved people not only try to save the quality of the products and services but also try to satisfy the customers, they are also interested in an increasing process outcome.

The increase of the process performance is closely linked to the adoption of the required and suggested practices in the context of agile software development. This means the team has to develop skills in using the practices and especially has to modify the practices to the requirements of the organization and the special needs of individual projects respectively project kinds. Due to the fundamental role of the team an important aspect is to optimize the composition of the team and their skills in different areas. To identify where this is necessary and how this can be done the agile principles suggest reflecting on “*how to become more effective, then tunes and adjust its behavior accordingly*” (Manifesto 2001).

Another aspect is the integration regarding the interaction of the used systems to support the development process like Computer Aided Software Engineering (CASE) tools.



Increasing process performance can be reached by optimizing the use of the suggested practices and also by increasing the scale of integration towards the optimum (SA05, p. 201).

### **3.6 Multidimensional control based upon the target figures customer satisfaction, time, quality and costs**

Besides the primary goal of the satisfaction of the customer's needs, financial aspects shall not be forgotten. Also the time of the production is important, because a customer can't wait to long - even for a "big" product. And the costs of course have to be taken into account, too. So these factors all together have to be matched to the big picture: assuring the organization's consistency.

The management of agile software development regarding the figures customer satisfaction, time, quality and costs is supported by several practices on the three levels of the development process. The first level project controlling is based upon classical approaches of project management like milestones. Regarding the process level and the customer satisfaction one aspect is, that the customer is directly involved into the development due to the on-site customer. Another aspect is that the customer is able to "see" the iteration (process) result very early due to the short time frame of an iteration and the short release cycles. The figure time on the process level focuses on the fact to not exceed the planned time for the project on the one hand for the entire project and on the other hand for each iteration. This is reached by the fact that the team, including the customer, are planning the next iteration and thus the team knows which functionalities can be implemented in the given time. The daily meetings required by the process level where everybody gives a status uncover possible delays in a matter of days.

Quality has to be ensured on the level of practices and activities. The focus here is on quality which is visible to the customer and not on internal quality attributes like code quality (nevertheless XP provides a practice for this aspect too). The quality is mainly ensured by two different activities. One is the automated testing of the code and the other one is a short release cycle. The automated testing helps to avoid errors in releases because a wide range of tests can be accomplished within a short time. Short release cycles ensure the quality regarding the usability and functionality of the software in its actual or final using environment (LEW02, p. 41ff).

Based upon the project triangle the costs are depending on the quality and the time and thus are not explained here in detail (OF03, p. 17).

### **3.7 Decentralized controlling**

Not only will every process be controlled by another department using e.g. activity-based costing, due to the fact that everybody is involved in the process and has to have the end quality of the product or service in his mind, the controlling can be done on different stations and levels. So besides the central controlling there is always a decentralized one e.g. using the results of a special service hotline.

The controlling of the software development process is decentralized because the development team defines at the beginning of an iteration itself, which functionality has to be implemented. How far this target is going to be reached is monitored by the daily status meeting. Possible delays due to problems can be identified early and necessary arrangements can be made to avoid any kind of delay. If the defined targets are not reached, the reasons are analyzed after the iteration but before starting the next one. The results are used to improve the process based upon organizational learning. This controlling approach is described by Scrum on the process level. Nevertheless it is not contemplated how this should be done. (Sc04, p. 9)

### **3.8 Organizational Learning**

The continuous process improvement (CPI) is one of the most important aspects within the BPM. Reaching CPI special approaches like the BPM cycle consisting of four phases (strategic process management, process design, process implementation, process controlling) and corresponding methods can be used to enhance the business processes (Al05, p. 1ff). To provide “better” processes - that are cheaper and faster and have a higher quality - all people involved in the process will therefore increase their knowledge by learning to use special methods and even find new ways on their own.

In the context of agile software development this aspect primary refers to the specific practices and proceedings and how they are used and learned by the developers working in the team. One of the basic thoughts in agile developing is naturally spoken that the developer team works as a team. Based on the fact that no defined interfaces between different processes do exist (e.g. there are no subassemblies), another aspect is the knowledge about the characteristic of the team members, i.e. which capabilities and qualifications do they have. According to Tuckman (Tu65, p. 76ff) a team hereby passes through four phases Forming, Storming, Norming and Performing. Many years experience show, that new teams - who’s members worked separately before - usually need 2-3 months reaching the phase Norming (Pi08, p. 18f). Changes of the team - equal if small or big ones - result in a re-cycling of all four phases. Therefore the composition of the team ideally should not be changed (Tu65, 78f).

## **4 Conclusion**

The comparison of agile software development processes and the classical primary business processes using their central aspects shows the similarity of the software development process and the classical product development process. Thus the software development process is one of the primary business processes of software development organizations. Although software development processes differ from classical business processes, their macro structures (core processes and supporting processes) are equal (SS08; TW07). Therefore it can be followed - regardless to the named differences - that special approaches of BPM are transferable to software development processes.

In a further step it will be researched if and in which way the approach of flexible BPM is transferable to agile software development processes. Connected to this it will be analyzed in which way the approaches of flexible BPM need to be tailored to the software development environment. Furthermore an evaluation of classical business processes and agile development processes with the focus on the paradigms of flexibility and service-orientation will be done.

## References

- [ASR02] Abrahamsson, P.; Salo, O.; Ronkainen, J.: Agile Software development methods - Review and analysis. Research report, VTT Publications, Espoo, 2002.
- [Al05] Allweyer, T.: Geschäftsprozessmanagement. Strategie, Entwurf, Implementierung, Controlling. W3L GmbH, Herdecke, 2005.
- [Ha07] Hamilton, P.: Dynaxity - Management von Dynamik und Komplexität. Springer, Heidelberg, 2007.
- [LM04] Lindvall, M.; Muthig, D.: Agile Software Development in large Organizations. IEEE - Computer 12, 2004, pp. 26-34.
- [LRW02] Lippert, M.; Rook, S.; Wolf, H.: Software entwickeln mit eXtreme Programming. d.Punkt, Heidelberg, 2002.
- [Ma01] Manifesto for Agile Software Development: Principles behind the Agile Manifesto, 2001, <http://www.agilemanifesto.org/principles.html>. Retrieved 2010-01-07
- [MR08] Meisinger, M.; Rittmann, S.: A comparison of service-oriented development approaches. Institut für Informatik der Technischen Universität München, 2008.
- [MSV97] Münch, J.; Schmitz, M.; Verlage, M.: Tailoring großer Prozeßmodelle auf der Basis von MVP-L. In: Montenegro S et al. (publ.): Vorgehensmodelle - Einführung, betrieblicher Einsatz, Werkzeugunterstützung und Migration. Beiträge zum 4. Workshop. Sankt Augustin: Forschungszentrum Informationstechnik, GMD-Studien (311), 1997, pp. 63-71.
- [OF03] Osterloh, M.; Frost, J.: Prozessmanagement als Kernkompetenz. Wie Sie Business Reengineering strategisch nutzen können. Betriebswirtschaftlicher Gabler, Wiesbaden, 2003.
- [Pi08] Pichler, R.: Scrum-Agiles Projektmanagement erfolgreich einsetzen. dPunkt, Heidelberg, 2008.
- [Po00] Porter, M. E.: Wettbewerbsvorteile. Campus, Wiesbaden, 2000.
- [SA05] Salo, O.; Abrahamsson, P.: Integrating agile software development and software process improvement: a longitudinal case study. Empirical Software Engineering, 2005, pp. 10ff.
- [SS08] Schmelzer, H. J.; Sesselmann, W.: Geschäftsprozessmanagement in der Praxis. Carl Hanser, München, 2008.
- [Sc07] Schwaber, K.: The Enterprise and Scrum. Microsoft Press, Redmond, 2007.
- [Sc04] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Redmond, 2004.
- [TW07] Thränert, M.; Werner, A.: A Process Family Approach for the reuse of development processes. In: Sobh T (publ.) Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. Springer, Netherlands, 2007.
- [Tu65] Tuckman, B. W. Developmental Sequence in Small Groups. Psychological Bulletin: 384-399, 1965.
- [Ve08] VersionOne: 3rd Annual Survey: 2008 "The State of Agile Development", 2008, [http://www.versionone.com/pdf/3rdAnnualStateOfAgile\\_FullDataReport.pdf](http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf). Retrieved 2010-01-07
- [VS07] Vigerschow, U.; Schneider, B.: Soft Skills für Softwareentwickler. dpunkt, Heidelberg, 2007.

# Ein begriffsbasierter Ansatz zur semantischen Extraktion von Datenbankschemata

Henri Mühle, Hannes Voigt, Wolfgang Lehner

Database Technology Group  
Technische Universität Dresden

**Abstract:** Die durch das rasante Anwachsen digitaler Datenbestände in Volumen und Vielfalt notwendig gewordene effiziente Verwaltung der erhobenen Datenbestände, bringt herkömmliche Datenbankmethoden an ihre Grenzen. Ein modelliertes Datenbankschema zur Grundstrukturierung der Datenbank kann längst nicht mehr statisch rigide modelliert werden. Vielmehr werden schemaflexible Datenbanken benötigt, die ihr Schema entsprechend an Änderungen im Datenbestand anpassen können. Da das Datenbankschema basierend auf einer konzeptuellen Datenbanksicht modelliert wird, präsentieren wir einen Ansatz, der die Formale Begriffsanalyse als Modellierungsmethode einsetzt. Die Formale Begriffsanalyse greift genau diese begriffsorientierte Welt-sicht auf. Damit können wir Schemaextraktion und weiterführende Problemstellungen mit wohl verstandenen und gut untersuchten Mechanismen behandeln. Im Rahmen dieses Beitrages stellen wir ein begriffsbasiertes Verfahren zur Schemaextraktion vor, das sich genau diese konzeptuelle Welt-sicht zu Nutze macht.

## 1 Einleitung

Mit dem rasanten Anwachsen digitaler Datenbestände geht die Anforderung nach einer effizienten Verwaltung der Bestände in immer neuen Anwendungsfeldern einher. Im Allgemeinen wird dafür auf die über Jahrzehnte bewährte Datenbanktechnologie zurückgegriffen. So sind Datenbanksysteme einer steten Diversifizierung ihres Anwendungsgebiets unterworfen. Dabei treten immer wieder die Grenzen ihrer zu-grundeliegenden Konzeption zu Tage.

Das Basiskonzept eines jeden bewährten Datenbanksystems ist ein modelliertes Datenbankschema, welches die Grundstrukturierung für Datenablage und -anfrage vorgibt. Bei der Modellierung eines Datenbankschemas wird eine Begriffsbildung durchgeführt, indem gleichartige Datenwerte zu einem strukturellen Merkmal abstrahiert und diese zu Begriffen zusammen gefasst werden. In der Entity-Relationship-Modellierung erfolgt die Begriffsbildung mittels Attributen und Entitytypen; im konstruktiven Schemaentwurf mittels Neben- und Hauptprädikatoren. Ein Datenbankschema fasst nun die zur Beschreibung der zu verwaltenden Daten gebildeten Begriffe zusammen und macht sie dem Datenbanksystem verständlich. Das Datenbanksystem orientiert sich dann beim Aufbau seiner physischen Ablage stark am Datenbankschema, also an den gebildeten Begriffen. Damit erreicht man Redundanzfreiheit zur Vermeidung von Änderungsanomalien, sowie eine Eingrenzung der zu lesenden Daten für die Beantwortung von Anfragen an diese Begriffsstruktur. Wird zum Beispiel durch eine Anfrage nach einer Person gesucht, so liest das Datenbanksystem ausschließlich Datensätze die dem Begriff „Person“ genügen. Das Datenbankschema lässt sich so als eine vereinbarte Begriffswelt zwischen Anwendung und Datenbanksystem verstehen.

Entscheidend ist, dass die Begriffsbildung außerhalb des Datenbanksystems stattfindet, die Ratio der Begriffsbildung dem System also verborgen bleibt. Dies hat zum einen zur Folge, dass eine Begriffsumbildung stets ein Eingreifen des Modellierers erfordert und zum anderen, dass das entwickelte Datenbankschema als fix betrachtet wird. Änderungen und Anpassungen am Schema sind der Sonderfall und in der Durchführung meist aufwändig, da stets auch abhängige Daten angefasst und angepasst werden müssen. Das ist aber unproblematisch solange die Begriffsbildung vorab stattfinden kann und eine stabile Begriffswelt als Vereinbarung zwischen Anwendung und Datenbanksystem zum Resultat hat.

In vielen neuen Anwendungsgebieten gestalten sich jedoch beide Bereiche, die Begriffsbildung im Vorhinein und die Vereinbarung einer stabilen Begriffswelt, als schwierig bis unmöglich. Die Bildung von Begriffen vor dem eigentlichen Betrieb einer Datenbank setzt voraus, dass alle Daten vorab strukturell bekannt sind und sich zu Begriffen vereinheitlichen lassen. Für die Vereinbarung einer stabilen Begriffswelt ist zwingend Voraussetzung, dass die Anwendung über ein stabiles, also wenig veränderliches, Weltverständnis verfügt. In vielen Anwendungen sind jedoch weder die Daten vorab vollständig strukturell bekannt, noch existiert ein stabiles Weltverständnis seitens der Anwendung.

Ein Beispiel für solche Anwendungen, sind sogenannte Multi-Tenant-Systeme. Diese hosten eine Anwendung für unterschiedliche Mandanten (Tenants) mit unterschiedlichsten Bedürfnissen. Zwar ergibt sich vorab aus der Anwendung eine gemeinsame Basis-Begriffswelt, jedoch wird diese oft an die Bedürfnisse einzelner Mandanten angepasst. Zudem sind die Bedürfnisse späterer Mandanten nur in eingegrenztem Maße bekannt, so dass sie sich begrifflich schwer im Vorhinein erfassen lassen. Jeder Mandant bringt ein Stück weit sein eigenes Weltverständnis in die Gesamtanwendung mit ein. Gerade bei Geschäftsanwendungen ist das Weltverständnis durch sich verändernde gesetzliche Vorgaben und Rahmenbedingungen ständigen Anpassungen unterworfen. Ein Multi-Tenant-System kumuliert dies und ist so einer sehr instabilen Begriffswelt ausgesetzt. [AGJ<sup>+</sup>08, For08]

Als zweites Beispiel sollen hier Anwendungen zur Unterstützung von Wissensarbeitern dienen. Wissensarbeiter erkunden Datenbestände nach neuen Erkenntnissen. Ihr Vorgehen folgt nicht immer festen Pfaden und Algorithmen. Aus einer Erkenntnis entstehen neue Fragestellungen, denen der Wissensarbeiter nachgeht. In jedem Schritt zieht er, in Abhängigkeit von Verfügbarkeit und Eignung für die Fragestellung, neue Daten heran. Welche Daten der Wissensarbeiter verwendet und welche strukturelle Form diese haben kann vorab nicht bekannt sein, da es sich erst im Laufe des Arbeitsprozesses ergibt. Ziel eines Wissensarbeiters ist es gerade ein Weltverständnis aufzubauen bzw. auszuweiten, dementsprechend ist das Weltverständnis seitens der Anwendung per se instabil. [End08]

Um Datenbanksysteme zu einem effizienten Umgang mit einer flexiblen Begriffswelt zu befähigen, sehen wir es als unerlässlich an, das Datenbanksystem selbst zur Begriffsbildung zu befähigen. Mit der Formalen Begriffsanalyse stehen wohl verstandene und gut untersuchte Konzepte, Formalismen und Algorithmen bereit, um automatisiert eine Begriffsbildung vorzunehmen. In dieser Arbeit betrachten wir als einen ersten Schritt, wie die Formale Begriffsanalyse grundsätzlich zur strukturellen Organisation von Daten in einem Datenbanksystem eingesetzt werden kann. Darauf aufbauend können dann weiterführende Mechanismen entwickelt werden, die diese Begriffsbildung im Zuge einer Schemaevolution ausnutzen.

Dazu stellen wir in Abschnitt 2 die notwendigen Begrifflichkeiten der Formalen Begriffsanalyse vor. Das Verfahren selbst gliedert sich dann in drei Schritte: einen Abstraktionsschritt (Abschnitt 2.1), einen Kollabierungsschritt (Abschnitt 2.2 und Abschnitt 2.3) und einen Extraktionsschritt (Abschnitt 2.4). Abschließend geben wir eine Zusammenfassung (Abschnitt 3) und einen Ausblick auf nachfolgende Arbeiten (Abschnitt 4).

## 2 Finden von Schemakandidaten mit Hilfe Formaler Kontexte

Die Formale Begriffsanalyse ist ein mathematisches Teilgebiet, das sich der Mathematisierung von „Begriff“ und „Begriffshierarchie“ widmet [GW96]. Zentrale Elemente der Formalen Begriffsanalyse sind sogenannte **formale Kontexte**. Darunter versteht man Tripel  $(G, M, I)$ , bestehend aus einer Menge  $G$  von **Gegenständen**, einer Menge  $M$  von **Merkmalen** und einer Inzidenzrelation  $I \subseteq G \times M$ , die beschreibt, ob ein Gegenstand  $g \in G$  ein Merkmal  $m \in M$  **hat**. Zur intuitiven Veranschaulichung formaler Kontexte werden Kreuztabellen verwendet, also Tabellen, deren Zeilen Gegenstände und deren Spalten Merkmale repräsentieren und in deren Zellen ein Kreuz steht, wenn der korrespondierende Gegenstand das korrespondierende Merkmal aufweist.

In diesen sehr allgemeinen Strukturen lassen sich nun **formale Begriffe** bilden. Das sind Paare  $(A, B)$  maximaler Teilmengen  $A \subseteq G, B \subseteq M$ , sodass jeder Gegenstand in  $A$  jedes Merkmal in  $B$  besitzt und gleichermaßen jedes Merkmal in  $B$  jedem Gegenstand in  $A$  besessen wird. Formal findet man diese Begriffe mit Hilfe der folgenden Ableitungsoperatoren

$$A' := \{m \in M \mid \forall g \in A : gIm\}$$

$$B' := \{g \in G \mid \forall m \in B : gIm\}$$

sodass für einen Begriff  $(A, B)$  stets  $A' = B$  und  $B' = A$  gilt. Man nennt  $A$  den **Begriffsumfang** und  $B$  den **Begriffsinhalt**. Auf der Menge aller Begriffe  $\mathfrak{B}(G, M, I)$  eines Kontextes  $(G, M, I)$  lässt sich eine Ordnungsrelation wie folgt definieren:

$$(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2 \quad (\Leftrightarrow B_1 \supseteq B_2)$$

Mit dieser Ordnung bilden die Begriffe eines Kontextes einen vollständigen Verband, den **Begriffsverband**  $\mathfrak{B}(G, M, I)$  des Kontextes  $(G, M, I)$ . Unter allen Begriffen von  $(G, M, I)$  seien noch die Begriffe der Form  $\gamma g := (g'', g')$  für  $g \in G$  und  $\mu m := (m', m'')$  für  $m \in M$  ausgezeichnet, die sogenannten **Gegenstands-** bzw. **Merkmalbegriffe**. [GW96]

### 2.1 Überführung der Datenbank in einen formalen Kontext

Um eine schemabezogene Ablage der Datensätze einer Datenbank zu realisieren, ist es hilfreich das Datenbankschema zu kennen. Wird die Datenbank von vornherein sauber modelliert, liegt das Schema explizit vor und die Datenablage kann dementsprechend strukturiert werden. Moderne Anwendungen erzeugen allerdings zunehmend Datenmengen, die nicht explizit strukturiert sind.

Das **Datenbankschema** beschreibt die semantische Struktur der Datensätze und besteht aus einer Überdeckung der Datenbankattribute<sup>1</sup> durch *semantische Einheiten*.

Überführen wir eine Datenbank derart in einen formalen Kontext, dass wir jeden Datenbankeintrag, der verschieden von NULL ist, durch ein Kreuz repräsentieren, bieten uns die Begriffsinhalte des zugehörigen Begriffsverbandes gerade einen strukturierten Suchraum für eine solche Überdeckung. Betrachten wir als Beispiel den bereits abstrahierten Datenbestand aus Datensätzen der freien Datenbank Freebase<sup>2</sup> in Abbildung 1. Im zugehörigen Begriffsverband (Abbildung 2) findet man durch die Begriffe

<sup>1</sup>Eine **Überdeckung** einer Menge  $M$  ist eine Familie  $\{M_t \mid t \in T\}$  von Teilmengen  $M_t \subseteq M$  für eine beliebige Indexmenge  $T$ , so dass  $\bigcup_{t \in T} M_t = M$ .

<sup>2</sup><http://www.freebase.com>

	Also known as	Date of Birth	Country of Nationality	Height	Weight	Position	Religion	President Number	Date founded	Country	Time Zone(s)	Population
Michael Jordan	×	×		×		×						
LeBron James	×	×		×	×	×						
Arnold Schwarzenegger	×	×		×	×		×					
Michael Schumacher		×	×									
Barack Obama	×	×					×	×				
Leeds	×										×	×
Berlin	×								×	×	×	×
New York City	×								×			×
Chicago	×								×		×	×

Abbildung 1: Ein formaler Kontext basierend auf Freebase-Datensätzen

eine Aufteilung der Datenbank in achtzehn logisch-strukturelle Einheiten. Der Begriffsverband bietet zudem eine visuelle Darstellung der Beziehungen zwischen diesen strukturellen Einheiten. Man sieht z. B., dass der grau markierte Begriff zum Datensatz *Chicago* ein Oberbegriff zu *Berlin* ist. Er prägt eine Teilmenge der Attribute seines Unterbegriffes aus, verallgemeinert diesen also.

Die Beschriftung des Verbandes ergibt sich so, dass ein Begriff ein Merkmallabel erhält, wenn er der größte Begriff ist, dessen Inhalt dieses Merkmal umfasst. Dual erhält er ein Gegenstandslabel, wenn er der kleinste Begriff ist, zu dessen Umfang dieser Gegenstand gehört. Die Merkmallabel oberhalb und die Gegenstandslabel unterhalb eines Begriffes ergeben dessen Zusammensetzung. Der markierte Begriff umfasst also gerade die Datensätze *Berlin* und *Chicago*, auf denen die Attribute *Population*, *Time Zone(s)*, *Date founded* und *Also known as* gemeinsam ausgeprägt sind.

Die Datenbank aus Abbildung 1 besitzt offensichtlich zwei semantische Einheiten, *Person* und *Stadt*. Natürlich bietet die Gesamtheit aller Begriffsinhalte eine Überdeckung der Merkmalmenge. Diese ist allerdings potentiell viel zu groß<sup>3</sup>, als dass sie zur Strukturierung einer schemabezogenen Ablage in Frage kommt. Zudem bietet diese Überdeckung keine semantische Trennung der Datensätze, da die einzelnen Datensätze im Normalfall zu verschiedenen Begriffsumfängen gehören. Unser Ziel ist es also, mit Hilfe einer Merkmalüberdeckung  $\mathcal{M} := \{M_t \mid t \in T\}$  die Gegenstände so zu gruppieren, dass man jedem  $M_t$  eine Menge  $G_t$  von Datensätzen zuordnen kann, so dass  $\mathcal{G} := \{G_t \mid t \in T\}$  eine Partition<sup>4</sup> der Gegenstandsmenge ist.  $T$  ist hierbei eine beliebige Indexmenge.

Bei der Erzeugung des Kontextes aus der Datenbank setzen wir nur dann Kreuze, wenn die Attributausprägung des jeweiligen Datensatzes explizit bekannt ist. Das bedeutet aber *nicht*, dass ein Datensatz ein Attribut, zu dem kein Kreuz existiert *nicht* hat. Es kann auch sein, dass er dieses Attribut zwar semantisch besitzt, der Attributwert aber nicht bekannt ist und der Datensatz das Attribut somit mit NULL ausprägt. Man spricht hierbei auch von *Unknown NULL-Values* bzw. von *Non-Applicable NULL-Values*. Auf einer logischen Ebene sind diese Datensätze natürlich als eigenständig zu betrachten, auf einer darüber stehenden semantischen Ebene können sie aber durchaus als Einheit angesehen werden. Genau solche semantischen Einheiten suchen wir.

<sup>3</sup>Zu einem formalen Kontext  $(G, M, I)$  kann es höchstens  $2^{|M|}$  Begriffsinhalte geben.

<sup>4</sup>Eine **Partition** einer Menge  $G$  ist eine Überdeckung in disjunkte Mengen.

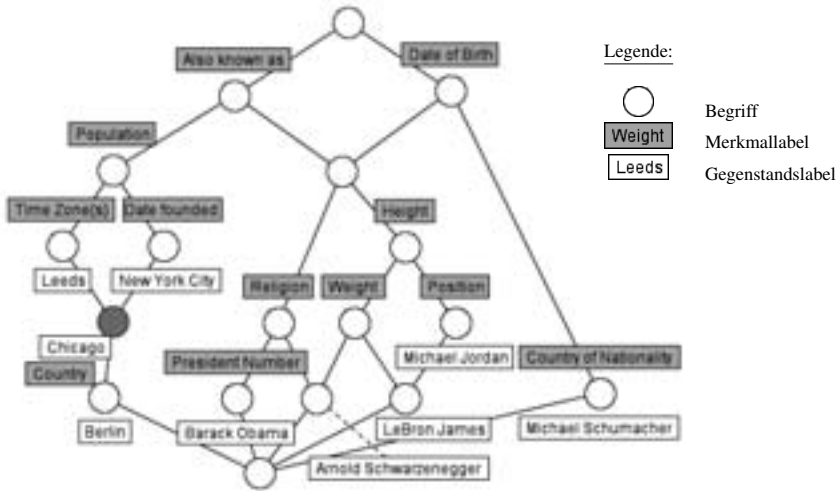


Abbildung 2: Der Begriffsverband zum Kontext aus Abbildung 1

## 2.2 Eine umfangsbezogene Ähnlichkeitsrelation auf den Merkmalen

Wir wollen unserem Ansatz die Annahme zu Grunde legen, dass ein Merkmal  $n \in M$  eines Kontextes  $(G, M, I)$  mit einem gegebenen Merkmal  $m \in M$  eine **semantische Einheit** bildet, wenn die Summe der Kreuze aus  $m'$  und  $n' \cap m'$  einen gewissen Prozentsatz der Rechteckfläche  $m' \times (m'' \cup n)$  überdeckt. Formal definieren wir damit auf den Merkmalen eines formalen Kontextes  $(G, M, I)$  eine Ähnlichkeitsrelation  $\sim$  zu einem Schwellwert  $t \in [0, 1]$ , so dass für  $m, n \in M$  gilt

$$m \sim n :\Leftrightarrow m' \cap n' \neq \emptyset \wedge \frac{|m'| \cdot |m''| + |m' \cap n'|}{|m'| \cdot |m'' \cup n|} \geq t$$

Außerdem schließen wir eine Ähnlichkeit zweier Merkmale aus, wenn ihre Merkmalumfänge disjunkt sind (sie also auf keinem Datensatz gemeinsam ausgeprägt sind). Eine Erweiterung auf Merkmalmengen (insbesondere Begriffsinhalte) erfolgt direkt.

Diese Art der Vergrößerung begünstigt „große Begriffe“. Zu diesen lassen sich wesentlich einfacher ähnliche Merkmale finden, da die bereits ausgefüllte Kreuzfläche durch den Begriff beliebig groß werden kann. Haben wir zu einem Begriff  $(A, B)$  ein Merkmal  $n \in M$  gefunden, mit  $B \sim n$ , dann erzeugen wir einen „größeren Kontext“  $(G, M, \tilde{I})$ , wobei

$$\tilde{I} := I \cup \{(g, n) \mid g \in A\}$$

Offenbar ist  $(A, B \cup \{n\})$  dann ein Begriff von  $(G, M, \tilde{I})$ .

Basierend auf dieser Ähnlichkeit stellen wir im nächsten Abschnitt eine Kollabierung des Begriffsverbandes durch Vergrößern der Begriffe vor.

## 2.3 Kollabierung der Begriffswelt

Um den in Abschnitt 2.4 beschriebenen Extraktionsschritt eindeutig durchführen zu können, müssen wir einen irreduziblen Kontext voraussetzen. Ein Kontext heißt



---

## Algorithmus 1 Der Algorithmus zur Kontextvergrößerung

---

**Require:** Kontext  $(G, M, I)$ , Threshold  $t$

```
1:  $(G, M, \tilde{I}) := (G, M, I)$ 
2: repeat
3:    $(G, M, \hat{I}) := (G, M, \tilde{I})$ 
4:    $\mathcal{B} := \emptyset$ 
5:   for all  $(A, B) \in \text{SEARCHSPACE}(G, M, \tilde{I})$  do
6:     for all  $m \in M \setminus B$  do
7:       if  $m' \cap A \neq \emptyset \wedge \frac{|A| \cdot |B| + |A \cap m'|}{|A| \cdot |B \cup \{m\}|} \geq t$  then
8:          $\mathcal{B} := \mathcal{B} \cup (A, B \cup \{m\})$ 
9:       end if
10:    end for
11:  end for
12:   $(G, M, \tilde{I}) := \text{CREATEFROMCONCEPTS}(\mathcal{B})$ 
13: until  $(G, M, \tilde{I}) = (G, M, \hat{I})$ 
14: return  $\text{CREATESCHEMACONTEXT}(G, M, \tilde{I})$ 
```

---

**irreduzibel**, wenn es keinen Gegenstand gibt, dessen Inhalt sich als Durchschnitt anderer Gegenstandsinhalte darstellen lässt und die duale Forderung für die Merkmalumfänge gilt.

Binden wir dieses Vorgehen nun in einen iterativen Algorithmus ein, dann vergrößern wir sukzessive die Inzidenzrelation und führen dabei eine Art semantisches Clustering der Gegenstände durch. Unser Algorithmus folgt dabei einem Greedy-Ansatz, so dass in jedem Iterationsschritt möglichst viele Merkmale an die Begriffe angeheftet werden.

Ein Algorithmus, der alle Begriffe durchläuft und zu jedem Begriff die ähnlichen Merkmale herausfindet und darauf basierend sukzessive den Kontext vergrößert (Algorithmus 1), bietet in jedem Schritt eine neue Konfiguration von logischen Einheiten an, die einer Kollabierung der vorherigen Konfiguration entspricht. Im Idealfall konvergiert dieses Verfahren auf eine Konfiguration semantischer Einheiten hin. Der Algorithmus bricht ab, wenn keine derartige Vergrößerung des Kontextes mehr möglich ist, also wenn entweder  $\tilde{I} = G \times M$  oder wenn die gefundenen Merkmalinhalte, die echt kleiner als der größte Begriff  $\top := (\emptyset', \emptyset'')$  sind, paarweise disjunkt sind.

Der Aufruf  $\text{CREATEFROMCONCEPTS}(\mathcal{B})$  (Zeile 12) erzeugt aus einer Menge  $\mathcal{B}$  von Paaren  $(A, B)$  mit  $A \subseteq G, B \subseteq M$  einen formalen Kontext  $(G, M, \tilde{I})$  mit  $\tilde{I} := \{(A, B) \mid (A, B) \in \mathcal{B}\}$ . Die gewünschte, minimale Merkmalüberdeckung findet man dann über die Atome des Begriffsverbandes zum größten Kontext. Ein Begriff heißt **Atom**, wenn er direkter oberer Nachbar des kleinsten Begriffes  $\perp := (\emptyset'', \emptyset')$  ist. Die Inhalte der Atome eines Verbandes bilden offenbar stets eine minimale, nicht-triviale Merkmalüberdeckung. Hierüber lässt sich auch ein alternatives Abbruchkriterium definieren, indem man eine maximale Anzahl semantischer Einheiten festlegt und den Algorithmus abbricht, wenn die Anzahl der Atome diesen Wert erreicht oder erstmals unterschreitet. Der Aufruf  $\text{CREATESCHEMACONTEXT}(G, M, \tilde{I})$  in Zeile 14 erzeugt aus dem kollabierten Kontext  $(G, M, \tilde{I})$  den Kontext der semantischen Einheiten, dessen Beschreibung in Abschnitt 2.4 folgt.

Aus Komplexitätstheoretischer Sicht ist dieser Algorithmus auf dem naiven Suchraum *aller* Begriffe allerdings äußerst unangenehm, da dies exponentiell viele sein können. Wir schlagen daher vor, als Suchraum nur die Merkmalbegriffe heranzuziehen. Anschaulich prüfen wir damit zunächst die Merkmalbegriffe untereinander auf semantische Ähnlichkeit und versuchen so den Begriffsverband von oben herab zu kollabieren. Da jeder Begriff  $(A, B)$  Unterbegriff aller Merkmalbegriffe  $(m', m'')$  mit  $m \in B$  ist, prüft das eingeschränkte Verfahren also zunächst, ob ein Begriff überhaupt Teil einer semantischen Einheit in unserem Sinne ist, ehe dieser Begriff um weitere

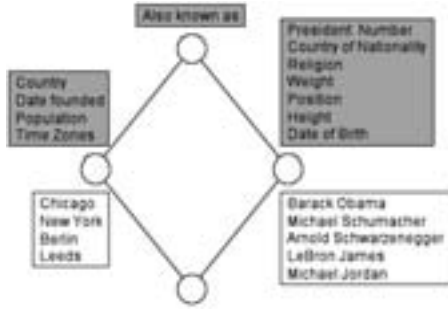


Abbildung 3: Der kollabierte Begriffsverband zum Kontext aus Abbildung 1 für  $t = 0.8$

Merkmale angereichert wird. Wir erhalten also eine semantisch striktere Suche. Dadurch kann es allerdings passieren, dass man nicht den gleichen kollabierten Kontext erhält, wie im Ansatz über alle Begriffe. Dies wird durch die Bildung des Kontextes der semantischen Einheiten aber wieder ausgeglichen. Mit dieser Eingrenzung des Suchraumes verbessern wir die Komplexität dieses Algorithmus erheblich. Allgemein hat jede Iteration eine Komplexität von  $O(|\text{SEARCHSPACE}| \cdot |M| \cdot |G|)$ . Da es höchstens  $|M|$  Merkmalbegriffe gibt, verringern wir die Komplexität also von  $O(2^{|M|} \cdot |G| \cdot |M|)$  auf  $O(|G| \cdot |M|^2)$ .

## 2.4 Extraktion der semantischen Einheiten

Allgemein müssen wir zunächst festlegen, wie wir aus dem kollabierten Kontext  $(G, M, \tilde{I})$  die gewünschten Überdeckungen  $\mathcal{G}$  und  $\mathcal{M}$  erhalten. Dazu wählen wir  $\mathcal{M} := \{M_t \subseteq M \mid (M'_t, M_t) \text{ Atom von } (G, M, \tilde{I})\}$ . Für die Gegenstandsüberdeckung  $\mathcal{G}$  wählen wir zu jedem Atom des kollabierten Kontextes genau die Gegenstände, deren Gegenstandsbegriffe oberhalb des Atoms liegen. Formal bedeutet das:

$$\mathcal{G} := \{G_t \subseteq G \mid t \in T\}, \text{ wobei } G_t := \{g \in G \mid \gamma g \geq (M'_t, M_t)\}$$

$T$  ist hierbei eine beliebige Indexmenge und  $\gamma g$  bezeichnet den am Anfang von Abschnitt 2 eingeführten Gegenstandsbegriff zum Gegenstand  $g$ . Anschließend erzeugt man den **Kontext der semantischen Einheiten**  $(G, M, S)$  mittels

$$S := \{G_t \times M_t \mid t \in T\}$$

Wenn der kollabierte Kontext irreduzibel ist, also kein Gegenstandsbegriff als Supremum anderer Gegenstandsbegriffe darstellbar ist (für Merkmalbegriffe dual), ist auch der Kontext der semantischen Einheiten irreduzibel und liefert über seine Atomumfänge eine Partition der Gegenstandsmenge. Zudem erzeugt auch Algorithmus 1 aus irreduziblen Kontexten stets wieder irreduzible Kontexte, da keine neuen Begriffe zum Begriffsverband hinzugefügt, sondern lediglich vorhandene Begriffe miteinander vereinigt werden.

Auf die Datenbank aus Abbildung 1 angewendet, erhalten wir für einen Schwellwert von  $t = 0.8$  mit unserem Algorithmus nach nur drei Iterationen den Begriffsverband der semantischen Einheiten in Abbildung 3. In diesem erkennen wir eine exakte Partitionierung der Gegenstände entsprechend der eingangs genannten semantischen Schemaelemente. Durch die Verwendung der Formalen Begriffsanalyse erhalten wir also direkt eine leicht verständliche Visualisierung der extrahierten semantischen Einheiten.

### 3 Zusammenfassung

Wir haben in diesem Artikel einen begriffsbasierten Ansatz zur semantischen Schemaextraktion aus nicht explizit strukturierten Datenbanken vorgestellt. Im Gegensatz zu anderen, graphen- oder logikbasierten Extraktionsverfahren, wie [BDFS97, LMP00, NAM98], nutzen wir explizit die strukturellen Informationen der Datenbank um den Suchraum von vornherein einzuschränken. Die Formale Begriffsanalyse bietet uns in diesem Zusammenhang einen theoretisch fundierten Ansatz um die vorausgesetzte instabile Begriffswelt zu modellieren und basierend auf einer erlaubten Unschärfe zusammenzufassen. Wir kollabieren dazu den Begriffsverband der Datenstruktur sukzessive, bis wir eine vorgegebene Anzahl von semantisch verschiedenen Schemaelementen unterschreiten. Anschließend extrahieren wir aus dem kollabierten Verband die Datensätze und Attribute, die den jeweiligen Schemaelementen genügen. Um dies zu erreichen, müssen wir eine Irreduzibilität des Datenbestandes voraussetzen, sodass es keine Datensätze gibt, deren Struktur aus anderen Datensätzen herleitbar ist. Insbesondere betrifft das Vererbungshierarchien auf den Typen der Datensätze. (Z. B. ist die Struktur eines Supertypen stets aus dem strukturellen Durchschnitt all seiner Subtypen herleitbar.) Da diese Reduzierung aber lediglich Datensätze entfernt, deren Struktur bereits implizit in anderen Datensätzen enthalten ist, ändert sich der Begriffsverband und damit der Suchraum für unseren Algorithmus *nicht*.

Unser Ansatz bietet zudem eine Erkennung von beliebig unstrukturierten Datenbeständen, indem der Algorithmus einen Kontext mit vollständig ausgefüllter Kreuztabelle zurück gibt. In diesem Fall gehören die Datensätze alle der gleichen semantischen Einheit an, bzw. sind diesbezüglich nicht unterscheidbar.

### 4 Ausblick

Eine Stärke unseres Ansatzes liegt in der sehr allgemeinen Modellierung durch die Formale Begriffsanalyse. Damit können wir nicht nur die in Abschnitt 1 genannten Anwendungsfälle in einer einheitlichen Sprache formulieren, sondern auch den gesamten Formalisierungsapparat auf datenbankspezifische Probleme anwenden. Somit erhalten wir eine neue Sicht auf die Problemstellung und damit auch einen gänzlich neuen Lösungsraum.

### Literatur

- [AGJ<sup>+</sup>08] Stefan Aulbach, Torsten Grust, Dean Jacobs, Alfons Kemper und Jan Rittinger. Multi-Tenant Databases for Software as a Service: Schema-mapping Techniques. In *SIGMOD'08*, 2008.
- [BDFS97] Peter Buneman, Susan B. Davidson, Mary F. Fernandez und Dan Suciu. Adding Structure to unstructured Data. In *ICDT'97*, 1997.
- [End08] Endeca. Endeca Information Access Platform, 2008.
- [For08] Force.com. The Force.com Multitenant Architecture, 2008.
- [GW96] Bernhard Ganter und Rudolf Wille. *Formale Begriffsanalyse: Mathematische Grundlagen*. Springer, 1996.
- [LMP00] Pierre-Alain Laur, Florent Masseglia und Pascal Poncelet. Schema Mining: Finding Structural Regularity among Semistructured Data. In *Principles of Data Mining and Knowledge Discovery*, 2000.
- [NAM98] Svetlozar Nestorov, Serge Abiteboul und Rajeev Motwani. Extracting Schema from Semistructured Data. In *SIGMOD'98*, 1998.

# Unterstützung von BPMN durch Lösungen der SAP

Sascha Alber

Bruchhäuser Weg 12  
69124 Heidelberg  
Sascha.Alber@gmx.de

**Abstract:** Business Process Model and Notation (BPMN) ermöglicht die Modellierung von Geschäftsprozessen und hat das Ziel, technische und fachliche Sicht auf Geschäftsprozesse in einer gemeinsamen Notation zu vereinen. Die SAP AG bietet eine Lösung an, welche die Geschäftsprozessmodellierung in BPMN und die Transformation der Modelle zu direkt ausführbaren Geschäftsprozessmodellen unterstützt. BPMN-Lösungen anderer Software-Hersteller bieten ähnliche Funktionalitäten, jedoch werden dort die ausführbaren Modelle durch Generierung von WS-BPEL Coding aus den BPMN-Modellen erzeugt. Die SAP-Lösung verfolgt stattdessen einen von der SAP entwickelten Ansatz für das Erzeugen ausführbarer Geschäftsprozessmodelle, der sich eng an das Prozessfluss-Konzept von BPMN hält. Diese Besonderheit ist Motivation, um eine Untersuchung der Unterstützung von BPMN durch SAP-Lösungen durchzuführen. Der Artikel beginnt mit einer kurzen Einführung in BPMN. Es folgt ein Überblick des Lösungsportfolios der SAP für das Geschäftsprozessmanagement. Dieser Überblick zeigt, dass nur SAP NetWeaver Business Process Management (BPM) die Erstellung von BPMN-Modellen und deren Transformation in ausführbare Geschäftsprozessmodelle unterstützt. In der darauf folgenden Analyse werden Praxisrelevanz und Restriktionen dieser SAP-Lösung untersucht. Der Artikel endet mit einem Ausblick auf die weitere Entwicklung von BPMN und SAP NetWeaver BPM.

## 1 Grundlagen

### 1.1 Business Process Modeling Notation / Business Process Model and Notation

Die Business Process Modeling Notation (BPMN) wurde von der Business Process Management Initiative (BPMI) entwickelt und ist seit Februar 2006 ein Standard der Object Management Group (OMG). Das Ziel von BPMN ist es, die Lücke zwischen der fachlichen und der technischen Sicht auf die Modellierung von Geschäftsprozessen zu schließen [Omg06]. Die aktuell freigegebene BPMN-Spezifikation ist in der Version 1.2 [Omg09a] seit März 2009 verfügbar. Für Juni 2010 ist die Freigabe für die Version 2.0 [Omg09b] geplant, welche bereits in einer Betaversion auf der OMG Webseite erhältlich ist.

Geschäftsprozesse können aus fachlicher Sicht im Rahmen der Prozessmodellierung betrachtet werden. Hierbei wird ein fachliches Modell erstellt, um ein Verständnis des grundlegenden Prozessablaufs<sup>1</sup> zu erhalten. Ein Geschäftsprozess kann hierfür z.B. in Form eines Flussdiagramms oder als ereignisgesteuerte Prozesskette (ePK) dargestellt werden. Selten auftretende Fälle oder Details, die für das grundlegende Verständnis des Prozessablaufs nicht von Bedeutung sind, müssen nicht ausmodelliert werden. Sie können entfallen, als Anmerkung im Modell aufgeführt oder außerhalb des Modells als Klartext ausformuliert werden [Al09]. Personen mit fachlicher Sicht auf Geschäftsprozesse sind beispielsweise der Prozessmodellierer oder die operativ ausführenden Mitarbeiter der Geschäftsprozesse.

Bei der technischen Sicht steht die technische Umsetzung der modellierten Geschäftsprozesse im Vordergrund [Omg09a]. Diese Sicht besitzt z.B. ein Software-Programmierer, der ein ausführbares Geschäftsprozessmodell implementieren möchte. Eine Möglichkeit zur technischen Beschreibung von Prozessen bietet die Business Process Execution Language (BPEL). BPEL unterstützt die Beschreibung von Prozessen, die im Rahmen einer serviceorientierten Architektur (SOA) durch Konsumieren von Services realisiert werden. Nicht durch BPEL beschreibbar ist die organisatorische Verantwortung bei den Prozessschritten [SI07]. Ein Standard für BPEL ist die Standard Web Service Business Process Execution Language (WS-BPEL) von der Organisation for the Advancement of Structured Information Standards (OASIS) [Oas07]. WS-BPEL ist eine auf der extensible Markup Language (XML) basierende Sprache zur Spezifikation des Verhaltens von Geschäftsprozessen auf Basis von Webservices. Webservices können über eine Webservice-Schnittstelle angesprochen werden, wodurch eine Interaktion zwischen verschiedenen Webservices ermöglicht wird. WS-BPEL ermöglicht die Koordination der Interaktion verschiedener Webservices, um einen Geschäftsprozess durchzuführen. Da diese Sprache auf XML basiert, erfolgt die Beschreibung von Geschäftsprozessen in einem Format, dass für Personen ohne Programmierkenntnisse nicht intuitiv verständlich ist.

Ein BPMN-Modell verbindet die fachliche und technische Sicht miteinander, indem eine aus fachlicher Sicht intuitiv verständliche grafische Notation zur Verfügung gestellt wird und die daraus entstehenden Modelle mit technischen Konstrukten verknüpft werden können. Aus fachlicher Sicht werden Geschäftsprozesse bei BPMN in einem Flussdiagramm beschrieben, dem Business Process Diagram (BPD). Das BPD stellt hierfür verschiedene grafische Objekte zur Verfügung, nachfolgend BPMN-Elemente genannt. BPMN-Elemente sind z.B. Aktivitäten, Ereignisse, Pools oder Gateways [Omg09a]. Mit Hilfe dieser Elemente ist die Modellierung von Geschäftsprozessen möglich und die Zuordnung von Prozessschritten zu Rollen, Personen oder anderen Organisationseinheiten kann abgebildet werden. Die zuvor angesprochene Verknüpfung der fachlichen mit der technischen Sicht erfolgt, indem technisch ausführbare Konstrukte wie Webservices, den BPMN-Elementen zugeordnet werden.

---

<sup>1</sup> Im Rahmen dieses Artikels ist der Begriff Prozess gleichbedeutend mit Geschäftsprozess.

Die BPMN-Spezifikation 1.2 [Omg09a] umfasst die Beschreibung und Bedeutung der BPMN-Elemente. Nicht enthalten ist die Ausführungssemantik der BPMN-Elemente oder anders formuliert, das Verhalten der BPMN-Elemente, wenn die BPMN-Modelle in ausführbare Geschäftsprozessmodelle transformiert werden. Des Weiteren fehlt die Definition einer Schnittstelle zum Austausch von BPMN-Modellen, beispielsweise für den Import oder Export zwischen zwei verschiedenen BPMN-Modellierungswerkzeugen. Einige der angebotenen BPMN-Werkzeuge verfügen über eine Schnittstelle für den Import oder Export von Modellen auf Basis des XML Process Definition Language (XPDL) Formats, jedoch ist die Umsetzung des XPDL Formats in den Werkzeugen nicht einheitlich [A109].

Die aktuell in Arbeit befindliche BPMN-Spezifikation Version 2.0 [Omg09b] soll die zuvor erwähnten Lücken schließen. Hierfür wird neben der Definition einer Schnittstelle für den Austausch von BPMN-Modellen das Business Process Definition Metamodel (BPDM) eingeführt. BPDM soll festlegen, wie die BPMN-Modelle bzw. die einzelnen BPMN-Elemente bei der Ausführung zu interpretieren sind. Sollten sich die Softwarehersteller an diese Vorgaben halten, wäre bei Nutzung verschiedener Anwendungen eine einheitliche Transformation der Modelle in ausführbare Modelle möglich. Eine weitere Neuerung der Version 2.0 ist die Umbenennung von BPMN. Durch die Aufnahme des Metamodells ist BPMN mehr als eine Notation, was sich in der künftigen Bezeichnung Business Process Model and Notation widerspiegeln soll [A109]. In diesem Artikel ist unter der Abkürzung BPMN die grafische Notation für Geschäftsprozessmodelle zu verstehen, die durch Zuweisung technischer Konstrukte in ausführbare Geschäftsprozessmodelle umgewandelt werden kann.

## **2 BPMN @ SAP**

### **2.1 Überblick der SAP-Lösungen für das Business Process Management**

Die SAP AG untergliedert Geschäftsprozesse in Standard-Prozesse und kundenindividuelle Prozessen [Sap08]. Die Standard-Prozesse werden als Application Core Prozesse bezeichnet. Sie repräsentieren nach Ansicht der SAP Geschäftsprozesse, die sich in der Praxis bewährt haben, sich selten ändern und die ohne größere Anpassungen bei einer Vielzahl von Kunden eingesetzt werden können. Sie umfassen demnach eine Unterstützung der Geschäftsprozesse, durch die sich Unternehmen nicht oder nur wenig voneinander unterscheiden, beispielsweise bei der Rechnungsabwicklung. Die Application Core Prozesse sind in der Auslieferung der SAP Business Suite enthalten, z.B. in SAP ERP, und können in begrenztem Maße kundenindividuell durch Änderung des Customizings angepasst werden [Si09].

Bei kundenindividuellen Geschäftsprozessen kann weiter zwischen Integrations- und Composite Business Prozessen unterschieden werden [Sap08]. Beide können von den SAP-Kunden nach deren individuellen Bedürfnissen zusammengestellt werden und sind deshalb nicht im Rahmen der Auslieferung von SAP-Anwendungen enthalten. Dies ist notwendig, um die Geschäftsprozesse abzubilden, die von Unternehmen zu Unternehmen sehr unterschiedlich sein können, beispielsweise im Bereich der Produktentwicklung. Die Integrationsprozesse dienen der system- und unternehmensübergreifenden Automatisierung von Prozessen. Der Schwerpunkt liegt hierbei auf der Integration verschiedener Systeme, weshalb ein systemzentrischer Blick auf die Geschäftsprozesse eingenommen wird. Composite Business Prozesse beschreiben dagegen system- und organisationsübergreifende Prozesse. Im Gegensatz zu den Integrationsprozessen können Geschäftsprozesse mit automatisierten und menschlichen Aktivitäten abgebildet werden. Der Fokus von Composite Business Prozessen ist auf die menschliche Zusammenarbeit in Prozessen gerichtet, weshalb ein menschenzentrischer Blickwinkel auf die Geschäftsprozesse eingenommen wird. Für jeden der drei beschriebenen Geschäftsprozessstypen stellt die SAP eine Lösung zur Verfügung, um Prozessmodelle und ausführbare Prozessmodelle erstellen zu können. Im Folgenden werden diese Lösungen kurz erläutert.

Die Geschäftsprozessmodelle der Application Core Prozesse sind als ePK-Modelle bei Auslieferung der SAP Business Suite enthalten. Mit Hilfe der Lösung SAP Enterprise Modeling Applications by IDS Scheer, können diese Modelle aufgerufen und verändert werden [Si09]. Eine Transformation der ePK-Modelle in ausführbare Geschäftsprozessmodelle wird nicht unterstützt. Stattdessen werden die ePK-Modelle in eine weitere Lösung, den SAP Solution Manager, importiert. Der SAP Solution Manager stellt das Business-Objekt Repository (BOR) zur Verfügung. In diesem Verzeichnis sind die Beschreibungen aller verfügbaren Funktionalitäten der SAP Business Suite aufgeführt und nach Business-Objekten gegliedert. Unter einem Business-Objekt ist ein reales Objekt zu verstehen, wie z.B. das Material oder der Kundenauftrag. Die importierten Geschäftsprozessmodelle werden in dem Business Process Repository (BPR) abgelegt und ihnen können Funktionalitäten aus dem BOR zugeordnet werden [SM09]. Durch die Zuordnung der Funktionalitäten entsteht eine Dokumentation, die im nächsten Schritt als Anleitung für die Implementierung eines ausführbaren Geschäftsprozessmodells verwendet werden.

Die Implementierung eines Application Core Prozesses erfolgt aus technischer Sicht als Programm innerhalb der Business Suite [Si09]. Für Prozessschritte mit menschlicher Interaktion können Benutzeroberflächen verschiedener Technologien eingesetzt werden, z.B. ABAP Web Dynpro. Zur Unterstützung der Koordination menschlicher Zusammenarbeit innerhalb eines Prozesses wird der in die SAP Business Suite integrierte SAP Business Workflow genutzt. Dieser koordiniert beispielsweise die Zuweisung von Aufgaben, indem ein zuständiger Bearbeiter mittels Rollendefinition dynamisch zur Laufzeit des Workflows ermittelt werden kann [MB00].

Zur Modellierung und Umsetzung der Integrationsprozesse wird die Lösung SAP NetWeaver Process Integration (PI) [Sap10a] eingesetzt. SAP NetWeaver PI ermöglicht die Interaktion von Anwendungen in heterogenen Informationstechnologie-Systemlandschaften. Dadurch ist es möglich, dass Anwendungen auf verschiedenen Systemen zusammenarbeiten können. Beispielsweise könnte eine SAP-Anwendung Informationen mit einer anderen Anwendung auf einem Legacy System austauschen.

Die Geschäftsprozessmodelle werden bei SAP NetWeaver PI in einer SAP-eigenen Notation mittels einer proprietären grafischen Oberfläche erstellt. Der Prozessablauf wird durch Aktionen beschrieben, die über synchrone oder asynchrone Verbindungen miteinander verbunden sind. Die menschliche Interaktion und Zusammenarbeit in einem Geschäftsprozess kann nicht modelliert werden [Si09]. Um von einem Modell zu einem ausführbaren Geschäftsprozessmodell zu gelangen, können den Aktionen eines Modells u.a. Webservices zugewiesen werden. Anschließend ist die Transformation des Modells in ein ausführbares Prozessmodell möglich, da z.B. eine Transformation in das WS-BPEL 2.0 Format unterstützt wird [Sap10a]. Für ein einfacheres Auffinden von Services stellt SAP NetWeaver PI ein Serviceverzeichnis zur Verfügung, das Enterprise Service Repository (ESR). Das ESR enthält neben kundeneigenen Services zusätzlich über 2800 Services der SAP Business Suite, mit denen auf einen Teil der Funktionalitäten der Application Core Prozesse zugegriffen werden kann [Sap08].

Für die Erstellung und Anpassung von Composite Business Prozessen wird das SAP NetWeaver Composition Environment (CE) [Sap10b] eingesetzt. Darin ist wiederum das SAP NetWeaver Business Process Management (BPM) enthalten, welches das Management von Geschäftsprozessen unterstützt. SAP NetWeaver BPM setzt sich aus den Bestandteilen Process Composer, Process Desk und Process Server zusammen. Der Process Composer ermöglicht die Erstellung von Geschäftsprozessmodellen nach BPMN und ist technisch ein Eclipse-Plugin. Den BPMN-Elementen in den Modellen können beispielsweise Services aus dem ESR für automatisierte Aktivitäten oder Benutzeroberflächen für menschliche Aktivitäten zugeordnet werden.<sup>2</sup> Anschließend sind diese Geschäftsprozesse auf einer Java Enterprise Edition (JEE) basierten Prozess-Engine direkt ausführbar, dem Process Server. Komplettiert wird die SAP NetWeaver BPM Lösung durch den Process Desk, der sowohl die menschliche Interaktion bei Durchführung der Geschäftsprozesse unterstützt als auch Monitoring- und Analysewerkzeuge für die Verwaltung und Kontrolle von Prozessen zur Verfügung stellt [Sap10b].

Das SAP NetWeaver Business Rules Management (BRM) ist ein weiterer Bestandteil des SAP NetWeaver CE. SAP NetWeaver BRM ermöglicht es, Entscheidungsregeln in Geschäftsprozessen festzulegen. Dadurch wird die Entscheidungslogik beschrieben, die unabhängig von der Prozesslogik ist [Si09]. Durch eine Integration kann SAP NetWeaver BPM direkt auf die Entscheidungsregeln des SAP NetWeaver BRM zugreifen [Sap10b].

---

<sup>2</sup> An dieser Stelle sei angemerkt, dass SAP NetWeaver CE und SAP NetWeaver PI eigenständige SAP Lösungen sind. Beide Lösungen beinhalten jeweils ein eigenes ESR. Sollte ein Kunde beide Lösungen einsetzen, wird nur das ESR von SAP NetWeaver PI verwendet.



## 2.2 Analyse der BPMN Unterstützung durch Lösungen der SAP

Die nachfolgende Analyse der Unterstützung von BPMN durch SAP-Lösungen konzentriert sich auf SAP NetWeaver BPM. Der Grund hierfür ist, dass nur diese SAP-Lösung die Modellierung von Geschäftsprozessen in BPMN und eine anschließende Transformation der Modelle zu ausführbaren Geschäftsprozessmodellen ermöglicht.

SAP Enhancement Package 1 for SAP NetWeaver Composition Environment 7.1 ist die Bezeichnung für die aktuelle Version von SAP NetWeaver CE, welches SAP NetWeaver BPM beinhaltet [Sap10b]. Diese Version von SAP NetWeaver BPM unterstützt eine auf BPMN 1.2 basierende Modellierung von Geschäftsprozessen. Es gibt jedoch Einschränkungen, da nicht alle BPMN-Elemente für die Modellierung zur Verfügung stehen. Beispielsweise fehlt das BPMN-Element für die Modellierung von Schleifen mit mehrfachen Instanzen.

Zur Erinnerung sei erwähnt, dass in BPMN 1.2 keine formalisierte Ausführungssemantik der BPMN-Elemente enthalten ist. Um die Modelle in ausführbare Modelle transformieren zu können, definieren die Software-Hersteller von BPMN-Lösungen eigene Ausführungssemantiken. Die SAP bildet an dieser Stelle keine Ausnahme und hat eine eigene Ausführungssemantik für die BPMN-Elemente definiert. Darauf aufbauend werden von SAP NetWeaver BPM die BPMN-Modelle interpretiert und in ausführbare Geschäftsprozessmodelle transformiert. Da die BPMN-Lösungen auf unterschiedliche Ausführungssemantiken zurückgreifen, werden die BPMN-Modelle unterschiedlich interpretiert [Si09]. Daraus folgt, dass die Transformation eines BPMN-Modells in verschiedenen BPMN-Lösungen zu verschiedenen ausführbaren Modellen führt.

Bei BPMN-Lösungen anderer Software Hersteller entstehen ausführbare Geschäftsprozessmodelle, indem BPEL-Coding aus den BPMN-Modellen generiert wird. Dieses Vorgehen ist fragwürdig, da BPMN auf einem flussorientierten Konzept, BPEL jedoch auf einem blockorientierten Konzept basiert. Dadurch entsteht das Problem, dass aufgrund der unterschiedlichen Konzepte die Transformation von Modellen zu falschen ausführbaren Modellen führen kann. Bei SAP NetWeaver BPM besteht dieses Problem nicht, da die Transformation von Modellen zu ausführbaren Modellen auf einem flussorientierten Konzept basiert, dass sich an dem Konzept von BPMN orientiert [St09].

Das Fehlen einer Import- und Export-Funktion von BPMN-Modellen ist eine weitere Einschränkung von SAP NetWeaver BPM. Ein Austausch von BPMN-Modellen zwischen SAP NetWeaver BPM und einer anderen BPMN-Lösung ist daher nicht möglich. Ein Hauptgrund hierfür ist, dass in BPMN 1.2 kein einheitliches Austauschformat für BPMN-Modelle definiert wird. Einige der angebotenen BPMN-Lösungen behelfen sich, indem sie den Import und Export von Modellen auf Basis des XML Process Definition Language (XPDL) Formats ermöglichen. Ein Austausch der BPMN-Modelle ist dennoch nur eingeschränkt möglich, da deren Umsetzung in das XPDL Format in den BPMN-Anwendungen nicht einheitlich ist [Al09].

Die Möglichkeiten und Einschränkungen von SAP NetWeaver BPM wurden bereits aufgeführt, offen ist jedoch die Frage der Praxistauglichkeit von SAP NetWeaver BPM. Zur Beantwortung dieser Frage wird auf die Ergebnisse einer Untersuchung der Anwendung von BPMN in der Praxis zurückgegriffen [MR08]. Diese Untersuchung zeigte, dass für die BPMN-Modellierung in der Praxis nur etwa 20% der verfügbaren BPMN-Elemente häufig eingesetzt werden. Diese häufig eingesetzten BPMN-Elemente werden von der aktuellen SAP NetWeaver BPM Version bereits unterstützt, weshalb dieser Lösung eine hohe Praxistauglichkeit unterstellt werden kann.

## **2.3 Weitere Entwicklung der BPMN Unterstützung durch SAP-Lösungen**

Mit der Standardisierung der Ausführungssemantik sowie der Definition eines Austauschformates für BPMN-Modelle in BPMN 2.0 [Omg09b] wird die Wiederverwendbarkeit der Modelle erhöht werden. Mit hoher Wahrscheinlichkeit führt dies dazu, dass BPMN bei Unternehmen verstärkt eingesetzt wird und damit das Marktpotential für BPMN-Lösungen steigt. Es bleibt jedoch abzuwarten, ob die genannten Lücken mit BPMN 2.0 tatsächlich geschlossen werden können und in welchem Maße, bzw. wie schnell die Software Hersteller von BPMN-Lösungen die in BPMN 2.0 gestellten Anforderungen in ihren Produkten umsetzen.

Seit Ende 2009 befindet sich SAP NetWeaver CE 7.2, inklusive einer neuen SAP NetWeaver BPM Version, in der Ramp-Up Phase und ist aktuell für Pilottester erhältlich [Sap10c]. Aus den bereits zugänglichen Release Notes ist ersichtlich, dass für die BPMN-Modellierung weitere, bisher fehlende BPMN-Elemente verfügbar sein werden. Ein Beispiel ist das bereits oben angesprochene BPMN-Element für die Modellierung von Schleifen mit mehrfachen Instanzen. Dies zeigt, dass die BPMN-Unterstützung dieser SAP-Lösung weiter ausgebaut wird.

Der Ausbau der BPMN-Unterstützung durch SAP NetWeaver BPM und die aktive Mitarbeit von SAP an der BPMN 2.0 Version [Sap10c, Omg09b] deuten darauf hin, dass die SAP ein großes Potential in BPMN sieht. Es ist daher davon auszugehen, dass der in der SAP NetWeaver BPM Roadmap [Sap08] skizzierte sukzessive Ausbau der BPMN-Unterstützung tatsächlich umgesetzt werden wird.

## **Literaturverzeichnis**

- [AI09] Allweyer, Thomas: BPMN 2.0 Business Process Model and Notation. Books on Demand, Norderstedt, 2009; S. 13-15.
- [MB00] Mende, Ulrich; Berthold, Andreas: SAP Business Workflow, 2. aktualisierte Auflage, Addison-Wesley Verlag, München, 2000; S. 19-28.

- [MR08] zur Muehlen, Michael; Recker, Jan: How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In (Bellahsene, Zohra; Léonard, Michel Hrsg.): Advanced Information System Engineering, 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008), Montpellier 2008, Springer Verlag, Berlin, Heidelberg, 2008; S. 465-479.
- [Oas07] OASIS: Web Services Business Process Execution Language Version 2.0, OASIS Standard, April 2007, verfügbar online unter <http://www.oasis-open.org/committees/wsbpel/>, letzter Abruf 15.01.2010.
- [Omg06] Object Management Group: Business Process Modeling Notation 1.0, OMG Final Adopted Specification, Februar 2006, verfügbar online unter <http://www.bpmn.org/>, letzter Abruf 15.01.2010.
- [Omg09a] Object Management Group: Business Process Modeling Notation 1.2, Januar 2009, verfügbar online unter <http://www.bpmn.org/>, letzter Abruf 15.01.2010.
- [Omg09b] Object Management Group: Business Process Modeling Notation 2.0 Beta 1, August 2009, verfügbar online unter <http://www.bpmn.org/>, letzter Abruf 15.01.2010.
- [Sap08] SAP AG; Business Process Management Roadmap, Version 6.0, Mai 2008, verfügbar online unter <http://www.sdn.sap.com/irj/sdn/nw-bpm>, letzter Abruf 15.01.2010.
- [Sap10a] SAP AG; SAP NetWeaver Process Integration 7.1 EhP1 Dokumentation, verfügbar online unter <http://help.sap.com/>, letzter Abruf am 15.01.2010.
- [Sap10b] SAP AG; SAP NetWeaver Composition Environment 7.1 EhP1 Dokumentation, verfügbar online unter <http://help.sap.com/>, letzter Abruf am 15.01.2010.
- [Sap10c] SAP AG; SAP NetWeaver Composition Environment 7.2 Dokumentation, verfügbar online unter <http://help.sap.com/>, letzter Abruf am 15.01.2010.
- [SI07] Stein, Sebastian; Konstantin Ivanov: EPK nach BPEL Transformation als Voraussetzung für praktische Umsetzung einer SOA. In (Bleek, Wolf-Gideon; Rassch, Jörg; Züllighoven, Heinz, Hrsg.): Software Engineering 2007, Lecture Notes in Informatics (LNI) 105, Gesellschaft für Informatik (GI), Hamburg 2007; Seite 75-80.
- [Si09] Silver, Bruce: SAP Netweaver BPM White Paper. Industry Trend Report, August 2009, verfügbar online unter <http://www.sap.com/>, letzter Abruf am 15.01.2010.
- [SM09] Schäfer, Marc Oliver; Melich, Matthias: SAP Solution Manager Enterprise Edition, 2. Auflage, Galileo Press, Bonn, 2009, S. 49-53.
- [St09] Stiehl, Volker: BPMN - Vermittler zwischen den Welten. In: Java Magazin, Ausgabe 09/2009, S. 72-77, verfügbar online unter <http://it-republik.de/jaxenter/>, letzter Abruf am 15.01.2010.

# Optimierung der Modellierungsergebnisse im Bereich interaktiver Unternehmensapplikationen durch sprachkritische Generierung des Domainwissens

Joachim Sternhuber

FG Entwicklung von Anwendungssystemen  
TU Darmstadt  
Hochschulstraße 1  
64289 Darmstadt  
sternhuber@winf.tu-darmstadt.de

**Abstract:** Aufbau einer optimierten Erfassungsmethode und des zugehörigen Prozesses zur Verbesserung der Ausgangsbasis zur Modellierung interaktiver wissensintensiver Unternehmensapplikation. Dabei erfolgt ein Überblick und Einordnung in den Gesamtrahmen eines prototypischen Projekts.

## 1 Einleitung

Die genaue Modellierung im Bereich Unternehmensapplikationen wird im Zuge der zunehmenden Aufgabenteilung im Bereich des Softwareengineering immer wichtiger. Nur eine fundierte und vollständige Modellierung ermöglicht einem „entkoppelten“ Team die Applikation im Sinne der Anforderung vollständig und „korrekt“ zu implementieren. Eine zusätzliche Herausforderung stellt die Integration der verschiedenen Interaktionsformen (vereinfacht: Mensch-Mensch, Mensch-Maschine, Maschine-Maschine) [DE07] dar. Dabei gewinnt der umgebende Kontext noch eine weit höhere Rolle als in der isolierten Betrachtung einer Applikation.

Im Rahmen eines Forschungsprojekts mit dem Ziel, User interaktiv mit Expertenwissen (i.S.v. Detailwissen und Erfahrungswissen über die Domäne, das weit über das Level eines Standard-Anwenders hinausgeht) zu unterstützen ist eine Fragestellung, wie die zu modellierende Inhalte möglichst gut generiert und erfasst werden. Dabei hat sich gezeigt, dass gerade für die interaktiven Bestandteile genauere und offenere Mechanismen bereitgestellt werden müssen.

Im vorliegenden Paper wird ganz kurz der Rahmen des Projekts hinsichtlich der Anforderungen exemplarisch skizziert. Das prinzipielle methodische Vorgehen wird anhand einer Beispieldomäne gezeigt, der Schwerpunkt liegt dann aber auf der Vorstellung einer Methode um zu einer optimierten Ausgangsbasis für die Modellierung, die Kontext und Dialogkomponenten berücksichtigt, zu gelangen. Im Rahmen des Papers kann dies leider nur in einer verkürzten Form dargestellt werden. Wo möglich wurden Graphiken zur Veranschaulichung und besseren Übersicht eingesetzt

## 2 Szenario und Kontext

### 2.1 Begriffsbestimmung Wissen und Wissensgenerierung im Projektrahmen

Wissen verstehen wir hier sowohl im klassischen Sinn des Wissensmanagements [Pr06, S. 246] [He02] als auch im Sinn der modellierungsrelevanten schematisierten Informationen über Kontext, Vorgehen, Strukturen und unternehmensspezifischer Bestandteile .

Die Entwicklung von Wissen [Am03][RK96] innerhalb eines Unternehmens ist ein sehr dynamischer Prozess. So entsteht neues Wissen in einem Unternehmen durch das Zusammenspiel von explizitem und implizitem Wissen, der Verwendung von vorhandenen Wissensbeständen und der Transformation von implizitem in explizites Wissen. [Pp98, S. 25] Durch diese bewusste Umwandlung wird ein Verständnis geschaffen, mit dem neue Wissensaspekte und der relevante Kontext integriert werden können. Zudem muss abgewogen werden, in welchem Fall es effektiver ist, Wissen selbst zu entwickeln oder Wissen extern zu beziehen. Hier soll in einem zweiten Schritt der Prototyp generiertes und modelliertes Wissen auf einer geeigneten Plattform zur Verfügung stellen.

Nonaka und Takeuchi haben zur Veranschaulichung der Dynamik die bekannte Theorie der Wissensspirale entwickelt, die in der Forschung hohe Akzeptanz findet. Abbildung 1 stellt die vier einzelnen Phasen, den Verlauf sowie die Methoden des Phasenübergangs der Wissensspirale dar.

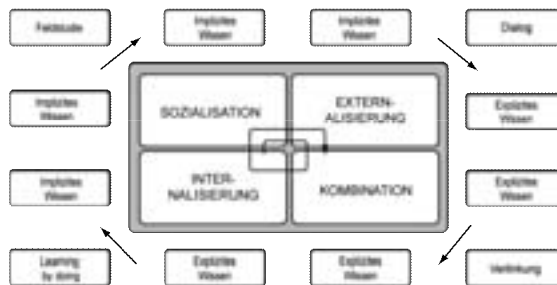


Abb. 1: Phasen der Wissensspirale (in Anlehnung an Nonaka/Takeuchi [NT95, S. 71] [He02, S. 105])

## 2.2 Projektszenario und Struktur

Das Projekt zur interaktiver Unterstützung von Anwendern mit Expertenwissen im Rahmen von Unternehmensanwendungen wurde prototypisch am Beispiel einer typischen spezialisierten Domäne durchgeführt. Gewählt wurde der Rückabwicklungsprozess (Rückgabe von Waren) innerhalb einer mittelständischen Unternehmensstruktur am Beispiel der DEG (Einkaufsgenossenschaft der Dachdecker). In einem ausführlichen Aufnahmeprozess wurden das prinzipielle Vorgehen sowie die Unterschiede in den einzelnen Filialen in Hessen untersucht und dokumentiert. Speziell für die Fragestellung der Userunterstützung wurden mehrere Erfassungsvorgänge durchgeführt um hier die unterschiedlichen Methodiken zu testen und zu optimieren. Die zugrundeliegende Struktur findet sich in Abbildung 1. Dabei wird der Begriff „Interview“ stellvertretend für unterschiedliche Erfassungsformen verwendet.



Abbildung 2: Methodische Schritte bei der Erfassung und Transformation der Ergebnisse

In Abbildung 2 sind die Teilschritte mit vereinfachten exemplarischen Inhalten aus der Erfassung im Bereich der DEG gefüllt.



Abbildung 3: exemplarische Ergebnisse, vereinfacht

Im Fokus für die Optimierung steht die erste Stufe „Interview“. Hier geht es darum Wissen und die Ausgangsbasis für die Modellierung zu generieren und optimieren, unter anderem durch den Einsatz von Experten. [Ra07, S. 439 f.] [Kh07, S. 46] Um einen optimierten Übergang vom Teilschritt „Interview“ zur Aussagensammlung und Terminologie zu ermöglichen, ist es notwendig die erfassten Inhalte sprachkritisch zu verarbeiten und zu normieren (i.S.v. Begriffsklärung, Einigung auf eine einheitliche Verwendung bestimmter Begriffstypen, Klären von vagen Bezeichnern etc.). Im Rahmen des zugrundeliegenden Projekts wurde eine vereinfachte Sprache zur maschinenlesbaren Darstellung verwendet: erweiterte Form von RDFS N3 (Details würden leider den Rahmen des Papers sprengen). Führt man die in Abbildung 2 skizzierte Vorgehensweise mit dem sprachkritischen Ansatz weiter, gelangt man zu folgenden Aussagearten und Sprachstufen innerhalb der verwendeten Sprachartefakte und Schematisierung (Abbildung 4).


SPRACHSTUFE	BEISPIEL	AUSSAGEART
Sprachstufe 2: Metasprache Informationen über Objekt- information	rdf:property rdf:type rdfs:Class	allgemeine
	eg:hatKundennummer rdf:type rdf:property	singuläre
Sprachstufe 1: Objektsprache Informationen über Objekte	eg:hatKundennummer rdfs:domain eg:Kunde	allgemeine
	eg:PersonXY eg:hatKundennummer „12345678“	singuläre
Objekte		Welt von Einzeldingen

Abbildung 4: Aussagearten und Sprachstufen übertragen auf RDFS

(in Anlehnung an Ortner [On00 S. 104])

### 3 Modellierung des optimierten Interviewprozesses

Ausgehend von dem skizzierten Vorgehen und Projektrahmen hat sich die Notwendigkeit für ein optimiertes Erfassen des Ausgangsmaterials zur Modellierung, speziell für den Einsatz im Unternehmen, ergeben. Im nachfolgenden Abschnitt erfolgt ein kurzer Überblick über die bekannten Verfahren, da nach wird ein eigenes optimiertes Modell (integrative Interview), das am Fachgebiet entwickelt wurde, und seine Bewertung vorgestellt.

#### 3.1 Bekannte Verfahren

Für den Einsatz in Unternehmen sind prinzipiell folgende Verfahren geeignet. Eine detaillierte Betrachtung findet sich unter anderem in [L05][Kk05]

- Receptives Interview: Das Wissen wird ohne jeglichen Einfluss durch den Interviewer in Reinform erfasst wird. Diese Art wird weitgehend in Kombination mit einer teilnehmenden Beobachtung eingesetzt und erfasst dadurch hauptsächlich implizites Wissen. Der Experte wird bei seinen Tätigkeiten beobachtet und kommuniziert von sich aus relevante Punkte. [Ls05, S. 381]

- Episodisches Interview: Es setzt durch seinen Wechsel von Erzählanreizen und direkten Fragen an der Schwäche durch die Einseitigkeit von erzählorientierten Verfahren an. Dieses Vorgehen ermöglicht die bewusste Aufnahme von explizitem und implizitem Wissen. Zur besseren Strukturierung werden Leitfäden eingesetzt.

- Problemzentriertes Interview: Diese Form startet mit einer allgemeinen Sondierungsphase und setzt dort gezielt einen Erzählanreiz. In der nächsten Phase werden gezielte Verständnisfragen gestellt und anschließend auch leitfadensorientierte direkte Fragen. Der Vorteil ist in der Kombination aus dem induktiven und dem deduktiven Vorgehen zu sehen.

- Tiefeninterview: Das Tiefeninterview ist angelehnt an das episodischen Interview, wobei die Abschnitte der freien Erzählungen durch ein Gespräch mit offenen Fragen, orientiert an einem Leitfaden, ersetzt sind. Direkten Fragen sind auch hier vorhanden, jedoch aufgrund des direkten Bezugs zu den offenen Fragen wesentlich detaillierter. Zudem werden Aussagen direkt in Frage gestellt und vorher erstellte Hypothesen gezielt gesucht.

- Convergent Interviewing: Ein wesentliche Erweiterung ist der zyklische Ablauf der verschiedenen Tiefeninterviews. Durch die strukturierten Interviews, die präzisen Zusammenfassungen und Auswertungen entsteht schrittweise ein tieferes Verständnis für die Materie. Ein zusätzlicher Vorteil ist, dass Divergenzen und Konvergenzen direkt nachgegangen werden kann und so nach und nach ein schrittweise Verfeinerung erreicht wird. [Ra07, S. 445]

Allerdings kann keine der Methoden wirklich den vollständigen dynamischen Prozess innerhalb der Wissensentwicklung, siehe Abbildung 1 dargelegt, berücksichtigen. Für Einzelgespräche sind je nach Situation die verschiedenen Interviewformen geeignet und helfen explizites sowie implizites Wissen der Experten zu erfassen. Den meisten Methoden fehlt dennoch eine übergreifende Schematisierung, die die Betrachtung über einen längeren Zeitraum ermöglicht und dadurch nachhaltig Wissensartefakte für den Erfasser schafft. Ebenso fehlt die Einbeziehung der individuellen Situationen im Unternehmen. Lediglich das Convergent Interviewing setzt bereits an diese Herangehensweise an. Es ist durch die bewusst geplante, langfristige Auseinandersetzung zwar in den einzelnen Tiefeninterviews sehr zielstrebig und dadurch oft nicht ausreichend geöffnet, jedoch im Gesamtablauf der dynamischste Prozess. Das Problem hierbei ist, dass es durch den ausschließlichen Einsatz des Tiefeninterviews oft zu wenig flexibel und anpassbar an die Experten herangeht. Der Erfassungsbereich in der Wissensspirale ist hauptsächlich die Externalisierung und die Kombination.



Die Methodik eignet sich somit ideal für das Erfassen von Detailwissen für einen bestimmten, im Laufe des Zyklus herauskristallisierten, Themenbereich. Die große Schwäche ist jedoch, dass das Vorgehen wenig geöffnet und wenig dynamisch ist. Schon vor dem Beginn des ersten Durchlaufs wird eine grobe Theorie entwickelt, deren Fundierung in den meisten Fällen das angestrebte Ziel ist.

Aufbauend auf diese Erkenntnisse soll also ein Erfassungsmodell definiert werden, das den schematischen und dynamischen Prozess des Convergent Interviewing aufnimmt, jedoch auch Vorteile der anderen vorgestellten Interviewverfahren im Einsatz im Unternehmensumfeld berücksichtigt.

### 3.2 Das integrative Interview

Als Grundlage für den Aufbau dieses neuen Modells soll erneut die Wissensspirale herangezogen werden, da sie die Wissensentstehung innerhalb des Unternehmens darstellt und die Dynamik der Wissenserfassung und -entstehung am besten verdeutlicht. Der Prozess des integrativen Interviews ist keine wirklich neue Herangehensweise innerhalb eines Interviewablaufs, sondern vielmehr eine erweiterte Kombination der geeigneten Interview-/Erfassungsverfahren. Es steht also das systematische Zusammenspiel als Prozeß im Vordergrund, dadurch auch die Bezeichnung integrativ. Hierbei wurden von den vorgestellten Verfahren hauptsächlich das rezeptive Interview (in Verbindung mit der teilnehmenden Beobachtung), das episodische Interview und das Tiefeninterview sowie ergänzend der Fragebogen herangezogen. Folgend wird zunächst eine Übersicht, die Begründung der Teilelemente und die Grundidee des Modells in Orientierung an der Wissensspirale vorgestellt. Darauf aufbauend wird auf die detailliertere Vorgehensweise eingegangen und abschließend das Modell in die Zyklen der Wissenserfassung eingeordnet.

#### 3.2.1 Idealtypische Struktur

Eine grobe Einordnung der verwendeten Verfahren für den neuen Interviewprozess des integrativen Interviews ist in der Abbildung 5 grafisch erfasst und stellt darüber hinaus ihre Einsatzpunkte innerhalb der Wissensspirale dar.

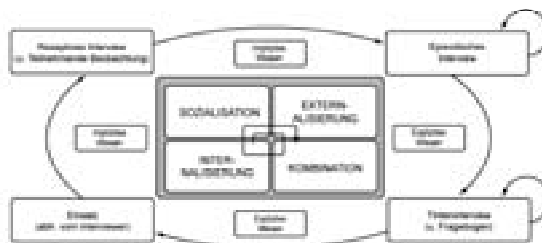


Abbildung 5: Struktur und Übersicht des kombinierten integrativen Interviews

Wichtig ist hier zu beachten, dass die einzelnen Interviewmodelle nicht exakt nur auf einen Bereich der Wissensspirale angewandt werden können, da die Grenzen unscharf sind. Der Gesamtprozess des neuen Modells ist somit, wie auch die Wissensspirale, als ein zyklischer Prozess zu verstehen, dessen Potential nicht mit einem einzigen Durchlauf vollständig erschlossen werden kann. Der Kreislauf beginnt in der Phase der Aufnahme von implizitem Wissen (Sozialisation). Dafür wurde hier das offenste Verfahren, das rezeptive Interview, gewählt. In diesem Modell wird es grundsätzlich in Verbindung mit einer eher kürzeren teilnehmenden Beobachtung gesehen, wobei sowohl die Erzählungen als auch die Beobachtungen eine wichtige Rolle spielen.

Um an das implizite Wissen des Experten durch Explikation in Erzählungen und der Beantwortung direkter Fragen zu gelangen (Externalisierung), wird hier das episodische Interview herangezogen. Die Wahl dieses Vorgehens ist dadurch begründet, dass es nicht zu problem- oder theorieorientiert ist und so noch ausreichend Freiräume für den Interviewten existieren. Eineweitere Besonderheit dieses Modells zeigt sich schon in diesem Schritt durch kleinere Zyklen innerhalb des Gesamtzyklus. Diese sind in Abbildung 5 durch die linksdrehenden Rundpfeile symbolisiert. Die Durchführung des episodischen Interviews erfolgt somit bei Bedarf schon beim ersten Gesamtprozess mehr als einmal. Diese Wiederholungen können mit demselben oder mit anderen Experten durchlaufen werden, um so ein breiteres Vorwissen zu erlangen. Generell sollte hier ein Durchlauf von etwa drei Interviews nicht überschritten werden. Auch das Tiefeninterview, das in diesem Modell hauptsächlich zur Verbindung von explizitem Wissen (Kombination) herangezogen wird, kann für besseres Detailverständnis hintereinander angewandt werden. Somit handelt es sich eher um den Einsatz einer kleineren Ausführung des Convergent Interviewing als um ein Tiefeninterview. Doch da sich auch hier die Anzahl der Wiederholungen in Grenzen halten sollten, ca. 3-7 Durchgänge, wurde hier bewusst das Tiefeninterview aufgeführt. Zur weiteren Fundierung der Ergebniss wird gegen Ende der Phase des Tiefeninterviews ein dynamisch generierter Fragebogen erstellt.

### **3.2.2 Detaillierte Vorgehensweise**

Nach der Vorstellung des Gesamtzusammenspiels der Teilprozesse des integrativen Interviews, wird in diesem Abschnitt das neue Modell noch einmal in einer detaillierteren Prozeßansicht dargestellt. Die Abbildung 6 stellt die 15 Schritte für den idealtypischen, einmaligen Durchlauf des integrativen Interviews dar. Grundlage sind hier die Prozessabläufe der verwendeten Interviewverfahren. Die Anordnung der Prozesse als Kreislauf wurde bewusst gewählt, damit die Parallelen zu Abbildung 5 deutlicher werden. Der Prozess beginnt mit einer ausgewählten oder zufälligen Situation, bei der durch aktives Zuhören und Beobachtungen implizites Wissen aufgenommen sowie dem Interviewer ein Einblick in die Thematik ermöglicht wird. Auf dieser Wissensgrundlage können eine ausführliche Zusammenfassung erstellt und erste Analysen vorgenommen werden. Diese erste Auseinandersetzung mit der Problematik löst Fragen aus und ermöglicht ein lockeres Festlegen der Richtungen zum Lösungsprozess.

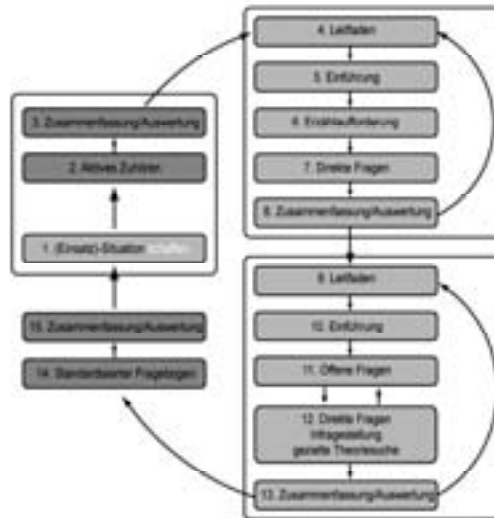


Abbildung 6: Detaillierte Prozessstruktur zur Gewinnung einer optimierten Modellierungsbasis

Diese werden im folgenden Schritt in einem Leitfaden festgehalten. Das Ziel des nächsten Teilprozesses ist es, festzustellen, ob die ersten Eindrücke auch in anderen Zusammenhängen auftreten, und ggf. modifiziert werden müssen. Bevor erste Erzählanreize gegeben werden, erfolgt eine Beschreibung des Gesprächsablaufs - hier dem des episodischen Interviews. Nach den Erzählungen, die oft schon auf die vorher beobachtete Situationen angesetzt wurden, befragt der Interviewer den Erzähler direkt über das Berichtete, um so sein Verständnis zu festigen. Gegebenenfalls werden weitere Experten, aufbauend auf die neuen Wissensbereiche des Interviewers, erneut zu einem episodischen Interview geladen. Aus den neuen Erkenntnissen und dem nun auch erfassten expliziten Wissen ergibt sich ein besseres Verständnis für den Interviewer, der auch schon vorsichtige Theorien entwickelt. Diese dienen als Grundlagen für den Leitfaden des nächsten Teilabschnitts. Auch hier erfolgt zunächst eine kurze Vorstellung des Interviewablaufs - dem des Tiefeninterviews. Das Gespräch beginnt direkt mit offenen Fragen, die sich stärker als zuvor an dem aufgestellten Leitfaden orientieren. Der Interviewer erhält hier innerhalb des integrativen Interview initial Detailinformationen des Interviewpartners. Aus diesem Grund sollte der Prozess nach der ausführlichen Zusammenfassung und Analyse erneut durchgeführt werden. Zudem können explizites Wissen detaillierter verknüpft sowie das während der Auswertung gefestigte Wissen und die Aspekte im folgenden Interview abgefragt werden. Nach jeder Durchführung des Tiefeninterviews müssen die Ergebnisse sorgfältig zusammengefasst und analysiert werden, da nur so ein zielbringender Leitfaden entstehen kann.

Um das Wissen und die Theorien zum Abschluss des Gesprächs zu fundieren, wird nach der letzten Anwendung des Tiefeninterviews ein Fragebogen herangezogen. Auch hier werden die Ergebnisse zusammengefasst und ausgewertet. Die Ergebnisse dienen als explizite Grundlage für eine optimierte Ausgangsbasis zur Modellierung der Domain. Dieser Kreislauf beginnt von vorne, jedoch auf einem höheren Wissenslevel, indem nun in diesem Einsatzgebiet ein rezeptives Interview zur Optimierung durchgeführt wird.

### 3.2.3 Einordnung in den Prozess der Wissenserfassung im Rahmen des umgebenden Projekts

Nachdem der Ablauf des integrativen Interviews vorgestellt wurde, sollen an dieser Stelle die betrachteten Teilverfahren in den Gesamtablauf der Wissenserfassung eingeordnet werden. Als Grundlage wurde hierfür ein Modell aufgegriffen, das die zwei Kreisläufe der Wissenserfassung aufzeigt. Abbildung 7 stellt diese Kommunikationskreise durch durchgezogene Kreise dar.

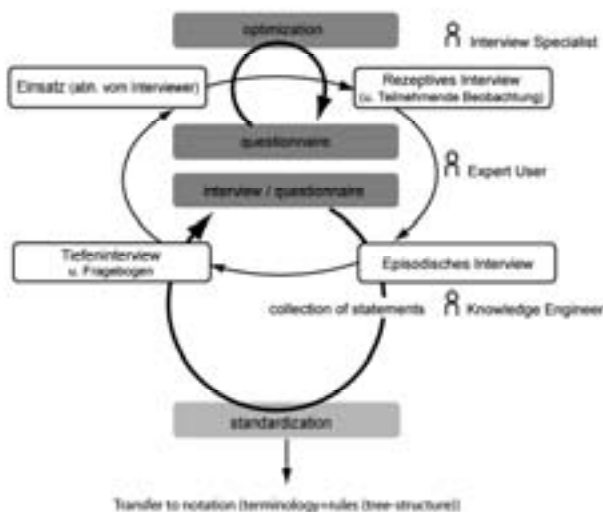


Abbildung 7. Einordnung und Struktur des integrativen Interviews im Gesamtprozess

Im ersten Zyklus findet die Interaktion zwischen einem Interviewer und dem Experten statt und hat zum Ziel, Optimierungen vorzunehmen. Bei diesen Optimierungen kann es sich um die Herauskristallisierung der relevanten Themen, aber auch um das allgemeine Ziel der Verbesserung in bestimmten Bereichen handeln. Der zweite Kreislauf findet zwischen dem Experten und einem Knowledge Engineer statt und dient dazu, Wissen langfristig und vielfältig einsetzbar zu archivieren. Da das Wissen hier greifbarer sein muss, wird überwiegend explizites Wissen erfasst und weiterentwickelt. Die Teile des integrativen Interviews sind mit ihrem Zyklus ebenfalls in Abbildung 7 eingeordnet.

Durch die gestrichelten Linien lässt sich der Gesamtablauf aus dem vorherigen Abschnitt (vgl. Abbildung 5) ablesen, wobei das Modell im Uhrzeigersinn weitergedreht wurde und nun der Einsatz den ersten Prozess darstellt. Auch das rezeptive Interview setzt an diesem Punkt an und versucht aus dem Umfeld Fähigkeiten aufzunehmen. Die Einordnung sagt nicht aus, dass diese beiden Prozesse bei jedem Zyklus zusammen angewandt werden müssen, sondern optional einsetzbar sind. Die beiden anderen Teilprozesse, das episodische Interview und das Tiefeninterview, werden durch den Knowledge Engineer angewandt um das Wissen zu erfassen. Das episodische Interview könnte auch in den ersten Kreislauf aufgenommen werden. Dies ist in diesem idealtypischen Ablauf nicht der Fall, da in Anlehnung an Abbildung 5 besonders das Tiefeninterview und das episodische Interview wiederholt aufgerufen werden. Für eine gründliche Wissenserfassung erfolgen die beiden gegebenen Kreisläufe aus Abbildung 7 also ergänzend und können in Kombination mit dem integrativen Interview exakt nach dem Prozess aus Abbildung 5 agieren.

### 3.3 Bewertung des integrativen Interviews

Die innerhalb des Projektsrelevanten Gütekriterien werden in diesem Abschnitt auf das neu entwickelte integrative Interview angewandt. Die Untersuchung beschränkt sich auch hier auf die Gegenstandsangemessenheit, die Offenheit für die subjektiven Sichten, das Hypothesenverhalten, die Betrachtung über einen längeren Zeitraum und die Grenzen der Methoden.

- **Gegenstandsangemessenheit:** Das integrative Interview bietet sich für die Erfassung von Wissen an, für das kaum Vorwissen bestehen muss. Im Laufe des Zyklus werden Wissensartefakte generiert und der Interviewer durchläuft den gesamten Wissensentstehungsprozess. Das Spektrum der aufgenommenen Fähigkeiten umfasst Wissen für Routinesituationen, Alltags- und Expertenwissen bis hin zu Einstellungen und Erfahrungen.
- **Offenheit für subjektiven Sichten:** Durch die starke Verwendung von (eher) erzählorientierten Interviewverfahren in den Anfangsphasen sind für den Berichtenden sehr viele Freiheiten vorhanden. Diese nehmen in den Phasen, in denen bewusst explizites Wissen erfasst wird, ab.
- **Hypotheseverhalten:** Auch hier muss der gesamte Kreislauf betrachtet werden. Zu Beginn des Interviews gibt es wenig Hypothesen. Im Laufe des Verfahrens entsteht verschiedenes Wissen durch die Vorteile der eingesetzten Interviewmethoden. Vor dem Einsatz des Tiefeninterviews wurde, aufbauend auf dem entstehenden Wissen, eine Theorie entwickelt, deren Fundierung und Verbesserung angestrebt wird.
- **Zeitraum:** Das integrative Interview findet aufgrund der starken Orientierung an der Wissensspirale über einen längeren Zeitraum statt. Der Gesamtzyklus sowie die Wiederholungen des episodischen Interviews und die des Tiefeninterviews innerhalb eines Durchlaufs sorgen für eine längere Auseinandersetzung und somit für eine bessere Verarbeitung und Erfassung des Wissens.

- **Grenzen:** Nicht für alle Wissensbereiche sind Erkenntnisse durch Beobachtungen möglich. In diesen Fällen muss auf diesen Teil des Verfahrens verzichtet werden, wodurch Grundlagen für den Aufbau von Vorwissen für die folgenden Interviews entfallen. Der Einsatz der erzählerorientierten Verfahren zu Beginn des Zyklus erhöht die Menge des Auswertungsmaterials und kann eine zu große Komplexität bewirken. Das Verfahren ist also sehr zeit- und somit auch kostenintensiv.

#### 4. Fazit

Das vorgestellte Verfahren zur Generierung der Ausgangsbasis für die Modellierung ist wie angedeutet sehr zeit- und damit kostenintensiv ermöglicht aber eine detaillierte Berücksichtigung von Kontext und Dialogkomponenten. Gerade die zunehmende Interaktivität fordert eine solche feingranulare und ganzheitliche Modellierung. Im Rahmen des Gesamtprojekts steht außerdem die Forderung nach Wiederverwendung der Wissensartefakte und Verwertung von Domainwissen in modularer Weise im Vordergrund. Dadurch wird das Verfahren auch wirtschaftlich interessant und sinnvoll. Hierbei wird dann innerhalb der verschiedenen Wissensartefakte nach unterschiedlichen Typen unterschieden. Kontext, Unternehmensspezifika, Branchenstandards, Erfahrungskomponenten sowie umgebende Regularien. In Kapitel 2 wurde ganz kurz der Rahmen des Gesamtprojekts skizziert bei dem sichtbar wird, dass die Ergebnisse schon teilweise in die Anfänge der Modellierung reichen. Eine genaue Betrachtung der weiteren Verarbeitung, Optimierung und Transformation sprengt den Rahmen des Papers. In dieser Arbeit wird aber schon ersichtlich, dass durch die Schematisierung, die Unterscheidung der unterschiedlichen Sprachstufen hier eine generische Arbeitsweise ermöglicht wird. Die Möglichkeiten der konkreten Modellierung in Bezug die Dialogkomponenten, Interaktivität und Reasoningansätze werden in einer weiteren Arbeit vertieft.

#### Literaturverzeichnis

- [DE07] David Nussbaum, Erich Ortner, Sven Scheele, Joachim Sternhuber: Discussion of the Interaction Concept Focusing on Application Systems. Web Intelligence/IAT (IEEE) Workshops 2007: S. 199-203
- [Am03] Amelingmeyer, Jenny: Wissensmanagement: Analyse und Gestaltung der Wissensbasis von Unternehmen. Wiesbaden : Dr. Th. Gabler Verlag, 2000
- [He02] Heckert, Uwe: Informations- und Kommunikationstechnologie beim Wissensmanagement. Wiesbaden : DUV, 2002
- [PR06] Probst, Gilbert ; Raub, Steffen ; Romhardt, Kai: Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Wiesbaden : Dr. Th. Gabler Verlag, 2006

- [RK96] Rehäuser, Jacob ; Krcmar, Helmut: Wissensmanagement im Unternehmen. 1996. - URL [http://www.winfobase.de/lehrstuhl/publikat.nsf/intern01/FC0F0EC41403EF3D412566500029C4A5/\\$FILE/96-14.pdf](http://www.winfobase.de/lehrstuhl/publikat.nsf/intern01/FC0F0EC41403EF3D412566500029C4A5/$FILE/96-14.pdf)
- [On00] Ortner, Erich: Wissensmanagement- Teil1: Rekonstruktion des Anwendungswissens, in: Informatik Spektrum, April 2000, S. 100- 108
- [NT95] Nonaka, Ikujiro ; Takeuchi, Hirotaka: The knowledge-creating company: How Japanese companies create the dynamics of innovation. New York : Oxford University Press, 1995
- [Pp98] Pawlowsky, Peter: Wissensmanagement: Erfahrungen und Perspektiven. Wiesbaden : Dr. Th. Gabler Verlag, 1998
- [Ls05] Lamnek, Siegfried: Qualitative Sozialforschung. Weinheim : Beltz, Psychologie Verlags Union, 2005
- [Ra07] Riege, Andreas: Convergent Interviewing. In: Buber, Renate (Hrsg.) ; Holzmüller, Hartmut M. (Hrsg.): Qualitative Marktforschung. Konzepte - Methoden - Analysen. Wiesbaden: Dr. Th. Gabler Verlag, 2007, S. 437-447
- [Kh07] Kocyba, Hermann: Die Bedeutung der Kategorie Wissen für den Wandel der Arbeit. In: Technikfolgenabschätzung - Theorie und Praxis 16. (2007), Juni, Nr. 2, S. 43-49
- [Kk05] Kassner, Karsten ; Wassermann, Petra: Nicht überall, wo Methode draufsteht, ist auch Methode drin: Zur Problematik der Fundierung von ExpertInneninterviews. In: Bogner, Alexander (Hrsg.) ; Littig, Beate (Hrsg.) ; Menz, Wolfgang (Hrsg.): Das Experteninterview. Theorie, Methode, Anwendung. Wiesbaden : VS-Verlag, 2005, S. 95-113

# Modellierung von Kennzahlensystemen mit BPMN

Nicole Zeise  
Fachgebiet Entwicklung von Anwendungssystemen  
Technische Universität Darmstadt  
Hochschulstr. 1  
D-64289 Darmstadt  
Zeise@winf.tu-darmstadt.de

**Abstract:** Mit dem Ziel einen möglichen Ansatz zur Verbindung von Kennzahlensystemen mit Prozesssystemen zu finden, diskutiert der vorliegende Beitrag Fragestellungen hinsichtlich der Modellierung von Kennzahlensystemen innerhalb einer Prozessstruktur unter Berücksichtigung der Eigenschaften eindimensionaler und mehrdimensionaler Kennzahlensysteme. Die ausgewählte Modellierungssprache für die Untersuchung ist BPMN. Innerhalb dieser Sprache wird untersucht, welche Möglichkeiten bestehen, Kennzahlensysteme direkt in der Prozessstruktur zu modellieren.

## 1 Einleitung

Prozessmanagement und Unternehmensmanagement mit Kennzahlen stehen in einer engen Verbindung. Für die Modellierung der Einzelsachverhalte existieren zwar Tools wie z.B. Oryx (auf Basis BPMN) oder ARIS (auf Basis EPK) zur Prozessmodellierung sowie die Balanced Scorecard zur Kennzahlenmodellierung, allerdings finden sich weder in der Literatur noch in der Praxis genügend Ratgeber zur konsequenten Verknüpfung der verschiedenen Ebenen von Prozesssystemen und Kennzahlensystemen. Durch die Unabhängigkeit der einzelnen Tools können die Ursache-Wirkungs- Zusammenhänge der Kennzahlen nur manuell plausibilisiert werden. Im Rahmen dieses Beitrags wird ein möglicher Ansatz zur Verbindung von Kennzahlensystemen mit Prozesssystemen, anhand der Modellierung solcher Kennzahlensysteme innerhalb einer Prozessstruktur erarbeitet.

Mit dem Ziel Kennzahlensysteme innerhalb von Prozesssystemen modellierbar zu machen, werden zuerst theoretische Grundlagen zu Kennzahlen und Kennzahlensystemen herausgearbeitet. Im nächsten Schritt werden Prozesssysteme aus einem Modellierungsstandpunkt heraus vorgestellt, um danach das Zusammenspiel zwischen Kennzahlen- und Prozesssystem zu analysieren. Die Eigenschaften der Kennzahlen sind dabei Ausgangspunkt zur Erarbeitung einer Methode, mit der eine Integration von Kennzahlensystemen- in Prozesssysteme gelingen könnte. Als Basissprache der Modellierung dient BPMN, mit der Erweiterung um ein spezifisches Kennzahlenelement.



## 2 Kennzahlen und Kennzahlensysteme als Werkzeuge der Koordination von Prozesssystemen

Wirkungsvolle Unternehmenssteuerung wäre ohne Kennzahlen nicht denkbar, da diese Zusammenhänge in hoch verdichteter Form transparent machen. Im Rahmen von Kennzahlensystemen ermöglichen sie eine tiefer gehende Ursachenanalyse und schärfen den Blick für die entscheidenden Zusammenhänge der unternehmerischen Tätigkeit. Im Folgenden werden insbesondere die Arten und Eigenschaften von Kennzahlen und Kennzahlensystemen dargestellt.

### 2.1 Grundlagen: Kennzahlen und Kennzahlensysteme

Kennzahlen sind entweder absolute Zahlen, Verhältniszahlen oder Indexzahlen, die in konzentrierter Form über zahlenmäßig erfassbare Sachverhalte berichten. [Küt06, S.51f] Tabelle 1 zeigt die Einteilung der Kennzahlen im Überblick. Die genannten Formeln entsprechen der grundlegenden Berechnung der Kennzahlen auf übergeordneter Ebene, wobei die Parameter  $x_i$  und  $y_i$  Merkmalswerte sind, die durch verschiedenartige Berechnungen von Einzelwerten entstehen können. Die entsprechenden Einzelberechnungen sind nicht Umfang dieser Arbeit.

Kennzahl	Ermittlung	Formel	Beispielkennzahlen
Absolute Zahlen	betriebswirtschaftliche Einzelwerte, Summenwerte, Differenzwerte und Mittelwerte	$f(x) = +(-) \sum x_i$ Mittelwert: $f(x) = \frac{1}{n} \sum x_i$	Umsatz, Jahresüberschuss, Bilanzsumme, Materialkosten
Gliederungszahlen (Verhältniszahl)	„Anteil einer statistischen Teilmasse an einer übergeordneten statistischen Masse“	$f(x) = \frac{x_i}{\sum x_i}$	Umsatzanteil, Materialkostenanteil, Eigenkapitalanteil
Beziehungszahlen (Verhältniszahl)	Vergleich sachlich verschiedenartiger Merkmalswerte, die einen inneren Zusammenhang aufweisen.	$f(x,y) = \frac{\sum x_i}{\sum y_i}$	Stückkosten, Return on Investment, Durchlaufzeit, PKW-Dichte pro Einwohner, Heiratshäufigkeit pro Einwohner
Messzahlen (Verhältniszahl)	Vergleich sachlich gleicher, aber örtlich oder zeitlich verschiedenartiger Merkmalswerte	$f(x(t)) = \frac{x_i(t_2)}{x_i(t_1)}$	Umsatzwachstum, Preisentwicklung, Marktwachstum
Indexzahlen	Erweitern das Konzept der Messzahlen, indem sie die zeitliche Entwicklung mehrerer Größen untersuchen	Preis- und Mengenindizes (nach Laspeyres, Paasche, Fischer)	Preisindizes, Mengenindizes, Wertindizes

Tabelle 1: Arten von Kennzahlen

Verknüpft man einzelne Kennzahlen empirisch über Ursache-Wirkungs-Beziehungen oder mathematisch entstehen Kennzahlensysteme. Diese Kennzahlensysteme werden nach Art der Verknüpfung in Rechen- und Ordnungssysteme unterschieden, wobei die Bezugsgrößen in Rechensystemen eindimensional sind, d.h. nur monetäre Kennzahlen berücksichtigt werden und die Bezugsgrößen in Ordnungssystemen mehrdimensional sind, d.h. monetäre und nichtmonetäre Kennzahlen berücksichtigt werden.

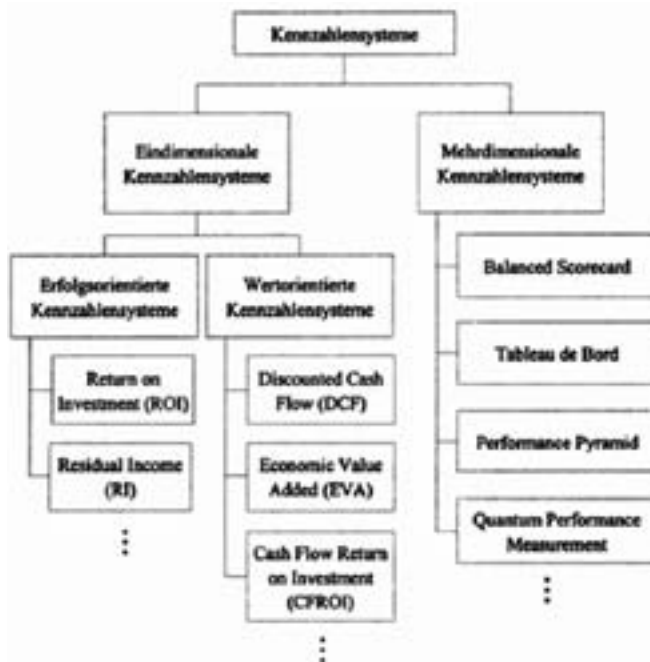


Abbildung 1: Kennzahlensysteme [Fri03, S.404]

Der Fokus wird im Weiteren auf den mehrdimensionalen Kennzahlensystemen liegen, da diese eine differenziertere Sicht auf die einzelnen Prozesse erlauben. Nichts desto trotz können eindimensionale Kennzahlensysteme nicht vernachlässigt werden, da sie systemübergreifend die Ergebnisse des Prozesssystems der Unternehmung aufzeigen. Sie bilden dementsprechend eine Makroperspektive, deren durchgängige Analyse durch ein multidimensionales Kennzahlensystem unterstützt werden kann.

## 2.2 Modellierung von Kennzahlensystemen

Bei der Verknüpfung von Kennzahlen zu Kennzahlensystemen sollten nach Gladen grundsätzlich folgende Anforderungen berücksichtigt werden:

- Objektivität und Widerspruchsfreiheit,

- Einfachheit und Klarheit,
- Informationsverdichtung,
- Multikausale Analyse,
- Indikatorfunktion und Systemoffenheit und
- Partizipation (systemübergreifende Konzipierung durch alle Betroffenen) [Gla08, S.92f]

Die Modellierung der Kennzahlen erfolgt zumeist Top-Down ausgerichtet an der Zielstruktur des Unternehmens. Ausgehend von den Unternehmenszielen ergeben sich Einzelziele auf verschiedenen Ebenen der Prozessstruktur. Deren Vielzahl hängt einerseits von der Granularität der Prozesse ab, andererseits von der Zielanzahl. Abbildung 2 zeigt dazu ein Beispiel aus dem Bereich der Textilindustrie. Das Unternehmensziel „Senkung der Wareneinsatzquote“, gemessen von der Kennzahl „Wareneinsatz zu Umsatz“ (nach Gesamtkostenverfahren), wird hier in einzelnen Unterprozessen detailliert. Dazu dienen wiederum Ziele und Kennzahlen, die eine Erreichung des übergeordneten Ziels fokussieren.

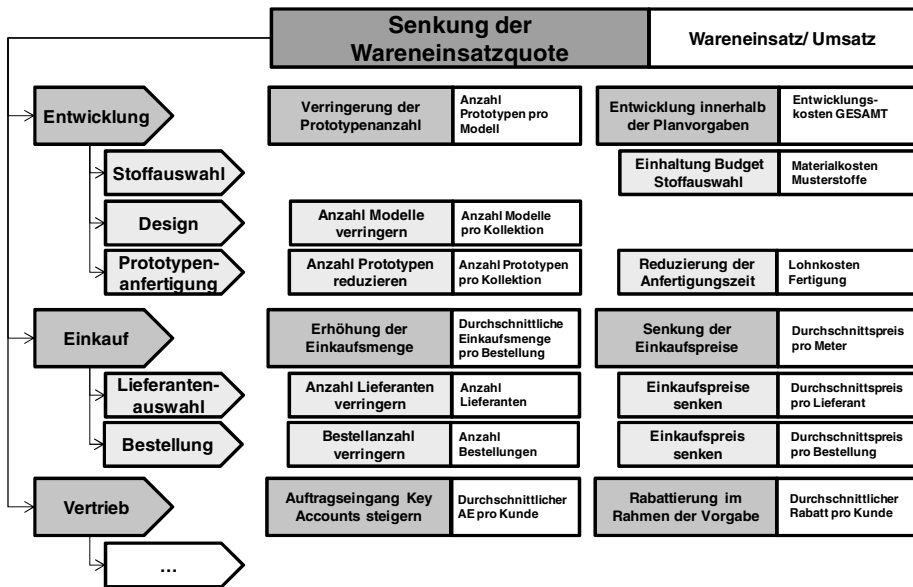


Abbildung 2: Beispiel für Top Down Modellierung von Zielen und Kennzahlen innerhalb der Prozessstruktur

Für die Detaillierung der Unternehmensziele anhand von Kennzahlen bis auf unterste Prozessebene, spielt die Art der Kennzahl hinsichtlich der Anforderung „Widerspruchsfreiheit“ eine bedeutende Rolle. So ist es einerseits sinnvoll, dass Kennzahlen untergeordneter Ebenen die gleiche Kennzahlenart aufweisen, wie Kennzahlen übergeordneter Ebenen. Andererseits ist es hinsichtlich der multikausalen Analyse notwendig Ursache – Wirkungsketten zu bilden, die es ermöglichen, Ursachen für die Entwicklung bestimmter Faktoren zu ermitteln. Dazu eignen sich besonders Beziehungszahlen, da Ihre jeweilige Berichtsgröße in einer angenommenen Art von einer zugehörigen Bezugsgröße verursacht wird z.B. Qualitätskosten in Abhängigkeit der Fehlerhäufigkeit oder der Gesamtstückzahl.

### 2.3 Kennzahlen in Prozesssystemen

Orientiert man sich an der Prozessteilung in Input, Transformation und Output und erweitert diese um Vor- und Folgesystem, erhält man ein durchgängiges Prozessmodell, welches auch die Schnittstellen zur Umwelt enthält. Nimmt man die Unterscheidung nach diesen Prozessstellen vor, ergeben sich sowohl Kennzahlen mit direktem Prozessbezug als auch Produktbezug, die beide wiederum Inhalte der Kundenzufriedenheit oder der Lieferantenbewertung sind und darauf basierend am Prozess selbst erfasst werden können. Innerhalb dieser Untergliederung kann das magische Dreieck als Orientierung dienen, alle Faktoren für die Zielerreichung in einem ausgewogenen Verhältnis zu berücksichtigen.

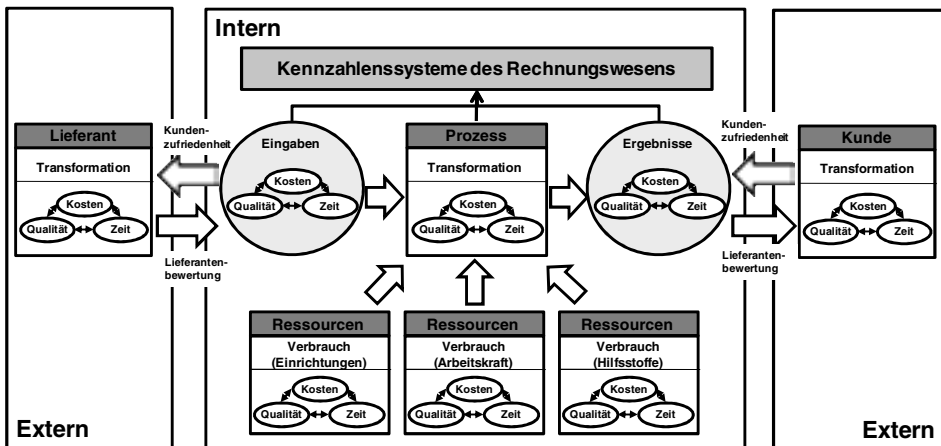


Abbildung 3: Pressestellen und Prozessmessung (eigene Darstellung)

Prozesse können also durch diverse Kennzahlen an mehreren Prozessstellen gemessen werden. Um dem Grundsatz der Einfachheit und Klarheit zu genügen, ist es allerdings unabdinglich, sich auf die wichtigsten Kennzahlen für die Zielerreichung zu beschränken, also die Steuerungsparameter der strategischen Steuerung zu fokussieren. Außerdem sollten Aufwand und Nutzen zur Erfassung bestimmter Zahlen immer in einem ausgewogenen Verhältnis stehen.

Ressourcen lassen sich nicht immer verursachungsgerecht den einzelnen Prozessen zuordnen. Beispielsweise sind Ressourcenkosten wie Instandhaltungskosten, der Verbrauch an Hilfsmaterial oder Bereitschaftskosten Kostenstelleneinzelkosten, die Kosten der Arbeitsvorbereitung oder des Betriebsbüros im Bezug auf die wertschöpfenden Prozesse sogar Kostenstellengemeinkosten, die keinem bestimmten Fertigungsauftrag oder Produkt zugeordnet werden können. [Rie 94, S.13ff] Die umfassende Beurteilung der Prozessleistung anhand von Kennzahlen wird allerdings nur modellierbar, wenn die Ressourcen betrachtet werden. Da sie nicht immer einen spezifischen Prozessbezug haben, stellen sie zusammen mit ihren Kennzahlen einen besonderen Fall der Modellierung dar.

Um den Bezug zum Rechnungswesen herzustellen ist es zudem sinnvoll Kennzahlensysteme mit den finanziellen Zielen des Unternehmens zu verknüpfen. Rechnungswesen orientierte Kennzahlensysteme werden zumeist als mathematische Kennzahlensysteme modelliert, da sie prozessübergreifend fungieren. Somit ist ein einfaches Herunterbrechen dieser Kennzahlen innerhalb der Prozessstruktur nicht möglich. Klar wird dies, wenn man das finanzielle Ziel Umsatzwachstum betrachtet. Zum Umsatzwachstum ist es notwendig, dass zunächst der Vertrieb den Kundenkontakt herstellt und Aufträge akquiriert. Deren Erfüllung wiederum muss innerhalb der Fertigung gewährleistet werden, um den Umsatz zu erzeugen. An dieser Stelle zeigt sich bereits, dass eine durchgängige Analyse gleicher Kennzahlen nur innerhalb funktionsgleicher Prozesse möglich ist. Daher ist die Unterscheidung der Kennzahlen nach den Funktionsbereichen des Unternehmens unabdinglich.

Zur Erfassung der Kennzahlen sind außerdem Faktoren wie Erfassungsverantwortung, Erfassungsintervall/ -zeitpunkt, Einheit, Datenquellen, Berechnungsformel, Zielwert und weitere von Bedeutung. [Lin02, S. 141f] Für die Modellierung sind jene zunächst nicht von Belang. Im Detail werden diese Faktoren erst in der „Arbeitsvorbereitung“ geplant.

### **3 Modellierung von Kennzahlensystemen innerhalb von Prozesssystemen**

Für die Modellierung von Kennzahlen zur Messung der Prozesse sind folgende Ausprägungen relevant:

- Art der Kennzahl (absolute Zahl, Gliederungszahl, Beziehungszahl, Messzahl, Indexzahl)
- Abhängigkeit von der Prozessstelle an der sie gemessen werden (Eingabe, Transformation, Mittelverbrauch und Ergebnisse)
- Inhaltlicher Kennzahlenbezug (Kosten, Zeit, Qualität)
- Produkt-, Prozess- oder Funktionsbezug (Einkauf, Entwicklung, Logistik...)
- Zielbezug

Weitere Detaileigenschaften können ergänzt werden.

### 3.1 BPMN 2.0 als Werkzeug zur Realisierung dieses Vorhabens

Management von Prozessen setzt die genaue Kenntnis der einzelnen Prozesse und insbesondere deren Zusammenhang voraus. BPMN bietet die Möglichkeit komplexe Systeme von Geschäftsprozessen anhand normierter Symbole und Syntax, transparent zu beschreiben. Außerdem liegt der grafischen Notation ein Metamodell zugrunde, wodurch eine standardisierte Modellausweitung ermöglicht wird.

BPMN 2.0 bietet folgende Diagrammart mit der entsprechenden Symbolik, um Geschäftsprozesssysteme zu modellieren [OMG 09, S.13ff]:

- Orchestrationsdiagramme (Modellierung privater und öffentlicher Prozesse),
- Kollaborationsdiagramme (Modellierung der Zusammenarbeit verschiedener „Pools“),
- Konversationsdiagramme (Modellierung der Informationsflüsse zwischen den Pools)
- und Choreographien (Kurzdarstellung für Prozesse zwischen den Pools).

Im Vergleich zu BPMN 1.2 wurde BPMN 2.0 um Conversations- sowie Choreographiediagramme erweitert. Während in den vorherigen Versionen Diagramme isoliert voneinander behandelt werden konnten, handelt es sich bei BPMN 2.0 um einen Multi-Perspektiven-Ansatz.

Mit diesen Möglichkeiten bietet der Standard einen Rahmen für die Geschäftsprozessmodellierung an sich, schließt aber explizit die Modellierung von Organisationsstrukturen und Ressourcen, Funktionseinheiten, Datenmodellen, Strategien und Geschäftsregeln aus. „In addition, operational simulation, monitoring and deployment of Business Processes are out of scope of this specification.“ [OMG 09, S.14]. Kennzahlen als Parameter der Prozessmessung werden damit innerhalb von BPMN nicht berücksichtigt.

Aufgrund der Nutzung eines standardisierten Metamodells ist es trotz der o.g. Einschränkungen möglich, neue Elemente zu integrieren. Open Source Produkte bieten dazu eine hinreichende Basis. Die Detailumsetzung als Open Source Projekt ist allerdings nicht Umfang dieser Arbeit. Um einen Ansatz zu finden, Kennzahlen- und Prozesssysteme konsequent zu verbinden, wird im Folgenden zunächst untersucht, welche Auswirkungen die Modellierung von Kennzahlen innerhalb von Orchestrationdiagrammen hätte, insbesondere im Hinblick auf die automatisierte Vererbung der Kennzeigeneigenschaften im Rahmen der Modellierung von Prozessen und Subprozessen.

### 3.2 Einführung einer Kennzalensymbolik für BPMN

Für die weiteren Betrachtungen wird als Symbol ein Halbkreis gewählt, dessen gerade Begrenzung an den Prozess anschließen soll. Ergänzt wird diese Symbolik durch folgende Kennzeichen innerhalb des Halbkreises:

Kennzeichen	Kennzahlenart				
	Absolute Zahlen	Gliederungszahlen	Beziehungszahlen	Meßzahlen	Indexzahlen
D	A	G	B	M	I

Tabelle 2: Auszug der Symbole für Kennzahlen im Rahmen der Modellierung

Zusätzlich kann diese Kennzeichnung um die Prozessstelle nach Vorsystem (V), Transformation (T), Ressourcen (Rx) und Folgesystem (F), den inhaltlichen Bezug nach Kosten (K), Zeit (Z) und Qualität (Q), als auch den Prozess- (P), Produktbezug (M) oder Funktionsbezug erweitert werden. Der Funktionsbezug muss dabei am Prozesssystem des jeweiligen Unternehmens ausgerichtet werden.

Bei dieser Betrachtungsweise tritt erneut die Ressourcenproblematik auf. BPMN bietet verschiedenartige Möglichkeiten zur Modellierung von Ressourcen. Sie können Output anderer Prozesse sein (z.B. Hilfsstoffe) und in Form von Sequenzflüssen dargestellt werden. Ressourcen können aber auch als Symbole innerhalb eines Prozesses z.B. manuelle Tätigkeiten (Arbeitskraft) modelliert werden. Eine weitere Möglichkeit besteht darin Ressourcen als Swimlanes z.B. Einrichtungen oder Maschinen, Abteilungen, Kostenstellen usw. zu zeigen. Die einheitliche Darstellung ist mit BPMN daher nur möglich, wenn zunächst Regeln für die Modellierung der Ressourcen festgelegt und eingehalten werden als auch die umfassende Ressourcenmodellierung angestrebt wird.

Kennzahlen könnten nicht nur im Prozess selbst, sondern zusätzlich für die Modellierung von Gateways als Mittel für das Treffen bestimmter Entscheidungen eine Rolle spielen. Dies ist allerdings nur in einer Mikroperspektive zu verstehen. Übergreifende Entscheidungen, wie eine eventuelle Prozessrestrukturierung werden zumeist nicht explizit modelliert, da diese auf prozessübergreifenden Managemententscheidungen basieren, die sich aus der Entwicklung bestimmter Kennzahlen ergeben können.

Wie in Abschnitt 2.2 gezeigt können für einen Prozess mehrere Kennzahlen existieren. Werden diese modelliert, soll das Symbol in Anlehnung an die Darstellung von Subprozessen um ein „+“-Zeichen erweitert werden.

### 3.3 Vererbung modellierter Kennzahleneigenschaften bei „einfach“ strukturierten Subprozessen

Die Strukturierung von Prozesssystemen spielt hinsichtlich der Vererbung bestimmter Kennzahlen auf Subprozesse eine erhebliche Rolle. Fraglich ist in dieser Hinsicht, ob eine übergeordnete Kennzahl (Vgl. Abschnitt 2.3 Beispiel der Umsatzproblematik) in den Folgeprozessen überhaupt weitergeführt werden kann, oder ob für die Unterprozesse weitere Kennzahlen definiert werden müssen, die die spezifische Leistung der Subprozesse im Hinblick auf das übergeordnete Ziel messen.

Teilweise besteht sicher die Möglichkeit Kennzahlen in Folgeprozessen weiterzuführen. Dies kann z.B. für Kosten, Durchlaufzeiten oder Fehlerraten gelten. Voraussetzung ist hier, dass die jeweiligen Größen den absoluten Kennzahlen, Gliederungskennzahlen oder auch Beziehungszahlen zuzuordnen sind. Zudem sollten Beziehungszahlen den gleichen Nenner aufweisen. Würde man in dem u.g. Beispiel im Entwicklungsprozess die Entwicklungskosten pro Prototyp messen, könnten sich für die Unterprozesse Stoffauswahl „Materialkosten pro Meter“ und für die Prototypenfertigung „Lohnkosten pro Prototyp“ ergeben. Das heißt, dass sich der Zähler der Kennzahl im Hauptprozess als Addition der Zähler beider Kennzahlen in den Subprozessen ergibt. Der Nenner wäre dann die Anzahl der Prototypen aus dem letzten Subprozess. Es ergeben sich also Ausnahmen, die eine einfache Vererbung der Eigenschaften in solchen Fällen erschweren.

Daher muss auf allen Ebenen des Prozesssystems die Möglichkeit bestehen, weitere Kennzahlen auch ohne eindeutige Abhängigkeit der Übergeordneten zu definieren. Diese Kennzahlen fungieren dann als Treibergrößen für die übergeordneten Kennzahlen. Im Weiteren wird davon ausgegangen, dass die Vererbung möglich ist und die Subprozesse anhand des gleichen Kriteriums gemessen werden können, wie die übergeordneten Prozesse.

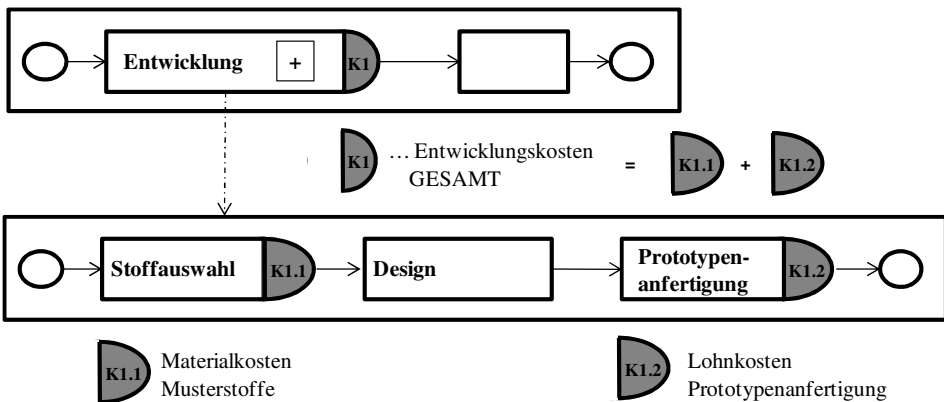


Abbildung 4: Prozess mit Subprozess und absoluter Kennzahl

Im einfachsten Fall der Modellierung bestehen Subprozesse aus einer Prozesskette von 2 Prozessen. In diesem Fall könnten die Eigenschaften der Kennzahlen vom Prozess direkt auf die Subprozesse vererbt werden. Wird die Kennzahlenart vererbt, ändert sich allerdings die Berechnung der Kennzahl im übergeordneten Prozess. Es existieren folgende Möglichkeiten im Hinblick auf die mathematische Abbildbarkeit in Abhängigkeit der Kennzahlenart:



Kennzahlenart	Subprozess 1	Subprozess 2	Kumulation
Absolute Zahlen	$f_1(x) = \pm \sum x_{i,1}$	$f_2(x) = \pm \sum x_{i,2}$	$f(x) = \pm \sum x_{i,1} \pm \sum x_{i,2}$
Absolute Zahlen (Mittelwerte)	$f_1(x) = \frac{1}{n_1} \sum x_{i,1}$	$f_2(x) = \frac{1}{n_2} \sum x_{i,2}$	Wenn: $n_1 = n_2$ $f(x) = \frac{\sum x_{i,1} + \sum x_{i,2}}{n}$
			Mittelwertbildung als gewichteter Durchschnitt
			Wenn: $n_1 \neq n_2$ $f(x) = \frac{g_1 \frac{1}{n_1} \sum x_{i,1} + g_2 \frac{1}{n_2} \sum x_{i,2}}{2}$
mit: $g_1 + g_2 = 1$			
Gliederungszahlen	$f_1(x) = \frac{x_{i,1}}{\sum x_{i,1}}$	$f_2(x) = \frac{x_{i,2}}{\sum x_{i,2}}$	Wenn: $\sum x_{i,1} = \sum x_{i,2}$ $f(x) = \frac{x_{i,1} + x_{i,2}}{\sum x_i} = 1$
			Wenn: $\sum x_{i,1} \neq \sum x_{i,2}$ $f(x) = \frac{x_{i,1} + x_{i,2}}{\sum x_{i,k}}$
Beziehungszahlen	$f_1(x,y) = \frac{\sum x_{i,1}}{\sum y_{i,1}}$	$f_2(x,y) = \frac{\sum x_{i,2}}{\sum y_{i,2}}$	Wenn: $\sum y_{i,1} = \sum y_{i,2}$ $f(x) = \frac{x_{i,1} + x_{i,2}}{\sum y_i}$
			Wenn: $\sum y_{i,1} \neq \sum y_{i,2}$ $f(x) = \frac{x_{i,1} + x_{i,2}}{\sum y_{i,k}}$
Messzahlen	$f_1(x(t)) = \frac{x_{i,1}(t_2)}{x_{i,1}(t_1)}$	$f_2(x(t)) = \frac{x_{i,2}(t_4)}{x_{i,2}(t_3)}$	$f(x(t)) = \frac{x_{i,1}(t_2) + x_{i,2}(t_4)}{x_{i,1}(t_1) + x_{i,2}(t_3)}$

Tabelle 3: Kumulation in einfach strukturierten Prozessen

Der Anwender muss also entscheiden, welches Kumulationsverfahren er wünscht und welches Verfahren zu den Folgeprozessen im Bezug auf die Zielerreichung passt.

## 4 Zusammenfassung und Ausblick

Insgesamt zeigen die Überlegungen, dass eine Modellierung von Kennzahlensystemen innerhalb von Prozesssystemen grundsätzlich möglich ist, allerdings ein breites Feld für weitere Untersuchungen in allen angesprochenen Themenfeldern bietet. Im Folgenden werden diese Themenfelder näher spezifiziert.

### **Modellierung von Kennzahlensystemen:**

Die Modellierung von Kennzahlensystemen ist stets abhängig von den unterschiedlichen Zielrichtungen der Unternehmen und muss daher individuell gestaltbar sein. Hier bietet sich ein breites Feld für weitere Untersuchungen, insbesondere im Hinblick auf die Modellierung von Kennzahlensystemen an sich, z.B. im Fall von Zielkonflikten, sowie die Methoden zur Ausgestaltung solcher Systeme. In diesem Zusammenhang sollte die Verknüpfung der Prozesskennzahlen mit den etablierten Kennzahlensystemen des Rechnungswesens fokussiert werden, um eine ganzheitliche Performanceanalyse zuzulassen.

Mit dem Ziel über alle Ebenen des Kennzahlensystems eine durchgängige Analyse für den Nutzer zu ermöglichen, ist im Weiteren zu untersuchen, welche Methodik dies zulässt, besonders da eine durchgängige Weitergabe der Kennzahlen lediglich in Spezialfällen möglich ist. Dies liegt vor, wenn absolute Zahlen, Mittelwerte oder Gliederungszahlen verwendet werden. Bei der Gruppe der Beziehungszahlen ist eine durchgehende Modellierung nur dann möglich, wenn die Subprozesse gleiche Prozesstreibergrößen besitzen.

Ein Ansatzpunkt weiterer Überlegungen kann zusätzlich zu der Kennzahlenmodellierung an sich, Zielwerte berücksichtigen. Bei solchen Betrachtungen könnte ein Ampelsystem konzipiert werden, dass auf übergeordneter Prozessebene die Zielerreichung zeigt. Hier haben beispielsweise Mevius, Oberweis und Stucky einen Ansatz auf Basis von Petri Netzwerken erarbeitet. [Mev09]

### **Modellierung von Prozesssystemen mit BPMN:**

BPMN als Sprache für die Prozessmodellierung bietet keinen Regelsatz, den Ressourcenverbrauch innerhalb der Prozessstruktur konsequent zu berücksichtigen. Eine solche einheitliche und umfassende Abbildung des Ressourcenverbrauchs ist allerdings auch nicht Ziel von BPMN, weshalb hier zunächst ein Regelframework zu erarbeiten ist, mit dem der Ressourcenverbrauch einheitlich abgebildet werden kann. Abweichend zu BPMN bietet beispielsweise das Resource-Event-Agent (REA) Framework einen individuellen Ansatz, der u.a. von Curch und Smith [Chu 07] untersucht wurde.

In weiteren Untersuchungen sollte die Auswirkung von Kennzahlen auf Gateways untersucht werden. Hier ergeben sich mehrere Ansatzpunkte, einerseits da Kennzahlen an sich Entscheidungsparameter für die Gateways liefern können, andererseits da der Durchlauf von Schleifen innerhalb von Prozessen die Kennzahlen beeinflusst. Dazu könnten in Anlehnung an die Ressourcenproblematik, einheitliche Modellierungsregeln diskutiert werden, die beispielsweise durch die spezifische Darstellung von Nacharbeitsprozessen, ein separates Aufzeigen von Kennzahlen ermöglichen.

### **Modellierung von Kennzahlensystemen innerhalb von Prozesssystemen:**

Um die Umsetzbarkeit des Vorschlags zu prüfen, ist zudem eine Betrachtung bereits existierender Werkzeuge der Prozessmodellierung im Hinblick auf die Modellierung von Kennzahlen unabdinglich. Zudem existieren weitere Betrachtungsfelder in der Entwicklung eines Metamodells für die Kennzahlen und deren detaillierte Verknüpfung mit BPMN als auch der Entwurf einer entsprechenden Softwarearchitektur. Der aktuelle Beitrag von Frank, Heise und Kattenstroth bieten in dieser Hinsicht interessante Anknüpfungspunkte [Fra09].

## **5 Literaturverzeichnis**

- [Chu07] Church, K.; Smith, R.: An Ontology-Based Dynamic Enterprise Model for Managerial Planning and Control, Mai 2007; Onlinepublikation: <http://www.aisvillage.com/rea25/program/smith.pdf>
- [Fra09] Frank, U./ Heise D./ Kattenstroth H.: "Use of a Domain Specific Modeling Language for Realizing Versatile Dashboards" in Matti Rossi; J. Gray; J. Sprinkle; Juha-Pekka Tolvanen (Hrsg.): Proceedings of the 9th OOPSLA workshop on domain-specific modeling (DSM), Helsinki Business School, Helsinki, 2009
- [Fri03] Friedl, Prof. Dr., B.: Controlling; Stuttgart: Lucius & Lucius Verlagsgesellschaft mbH; 2003
- [Gla08] Gladen, W.: Performance Measurement, 4. Aufl., Wiesbaden: Gabler GWV Fachverlage GmbH, 2008
- [Küt06] Küting, K.; Weber, C.-P.: Die Bilanzanalyse. Beurteilung von Abschlüssen nach HGB und IFRS, 8. Auflage, Schäffer-Poeschel, Stuttgart 2006
- [Lin02] Linß, Univ.-Prof. Dr.-Ing. habil. G., Qualitätsmanagement für Ingenieure, Fachbuchverlag Leipzig, 2002
- [Mev09] Mevius, Dr. M.; Oberweis, Prof. Dr., A.; Stucky, Prof. em. Dr. Dr. h.c., W.: Neue Ansätze bei der Modellierung eines kennzahlenbasierten Managements von Geschäftsprozessen, 2009
- [OMG09] Object Management Group, Inc.: Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0, Onlinepublikation: <http://www.omg.org/spec/BPMN/2.0>
- [Rei08] Reitz, A.: Lean TPM – In 12 Schritten zum schlanken Managementsystem, München: FinanzBuch Verlag GmbH, 2008
- [Rie94] Riebel, P.: Einzelkosten und Deckungsbeitragsrechnung – Grundfragen einer markt- und entscheidungsorientierten Unternehmensrechnung; 7. Aufl.; Wiesbaden: Gabler, 1994

# Interaktive Konfigurierung dynamischer Anwendungssysteme aus Komponenten

Tobias Grollius

Entwicklung von Anwendungssystemen  
Technische Universität Darmstadt  
Hochschulstr. 1  
64289 Darmstadt  
tobias.grollius@gmx.de

**Abstract:** Dynamische Anwendungssysteme haben das Potenzial, sich schnell an veränderte Bedingungen anzupassen. Sie werden prozess- und komponentenorientiert konstruiert. Dieser Beitrag stellt eine Methode zur Modellierung der komponentenbasierten Aufbaustruktur und interaktiven Konfigurierung dynamischer Anwendungen vor, die sowohl Komponenten-Varianten berücksichtigt als auch Raum für die Kreativität des konfigurierenden Menschen lässt.

## 1 Dynamische Anwendungssysteme

Aus Sicht des Enterprise Engineerings werden ganze Unternehmen als Anwendungssysteme in einem umfassenden und ganzheitlichen Begriffsverständnis aufgefasst [AGO09]. Ein solches Anwendungssystem reicht von den Technologieträgern und der Hardware über Betriebs- und Basissysteme sowie den (Software-)Anwendungen, die auch Services genannt werden, bis hin zu den Arbeitsprozessen, dem benötigten Wissen zu ihrer Ausführung und schließlich den Menschen in einem Unternehmen.

Dynamisch sind Anwendungssysteme, wenn sie das Potenzial besitzen, sich kurzfristig an sich ändernde Bedingungen anpassen zu können [OEE08]. Mit der Prozessorientierung stellt man den zielgerichteten und an den Kundenanforderungen orientierten Ablauf der Arbeitsprozesse in den Mittelpunkt der Organisationsgestaltung, um so schneller auf sich verändernde Markt- und andere Rahmenbedingungen reagieren zu können und wettbewerbsfähig zu bleiben [Sc05]. Aufgrund des hohen Automatisierungs- bzw. Unterstützungsgrads eines Großteils der Arbeitsprozesse mit Informationstechnologie (Anwendungen) wird die Anpassungsfähigkeit der Prozesse und folglich auch die Dynamik des Anwendungssystems jedoch maßgeblich von der Anpassungsfähigkeit der sie ausführenden oder unterstützenden Anwendungen bestimmt [PP05].

Komponentenbasierte Anwendungen bieten die notwendige Anpassungsfähigkeit [Or05a]. Sie werden bedarfsgerecht und flexibel aus wieder verwendbaren Komponenten zusammengesetzt, die eine abgeschlossene Funktionalität über standardisierte

Schnittstellen anbieten, standardisiert spezifiziert sind und lose gekoppelt werden. Die Komponenten werden nicht an Anforderungen angepasst (abgesehen von Parametrisierung), sondern die geeigneten Komponenten-Varianten ausgewählt, die in ausreichender Anzahl vorliegen müssen. So können Anwendungen in kurzer Zeit aus den geeigneten Komponenten(-varianten) *konfiguriert* und rekonfiguriert werden.

## **2 Varianten-basierte Konfigurierung**

In einem prozess- und komponentenorientierten Vorgehensmodell für die Entwicklung dynamischer Anwendungssysteme, wie z.B. dem Multi-Path-Process-Model (MP<sup>2</sup>M) [Or05b], folgt die Konfigurierung auf die fachliche Rekonstruktion eines Anwendungssystems und die Arbeitsprozessmodellierung. Aufgabe ist es, das Prozessschema – das rekonstruierte Arbeitsprozessmodell – mit Komponenten zu bestücken, die die Aktivitäten oder Teilprozesse eines Prozesses zu dem Prozessschema automatisch ausführen oder Menschen bei der Ausführung unterstützen.

Um einen Arbeitsprozess möglichst ideal, ohne unbeabsichtigte Prozessanpassungen ausstatten zu können und der geforderten Dynamik des resultierenden Anwendungssystems gerecht zu werden, sind zahlreiche Komponenten und Varianten dieser Komponenten erforderlich. Komponenten-Varianten sind zur gleichen Zeit gültig, sie besitzen jedoch bezüglich definierter Merkmale unterschiedliche, systematisch variierte Merkmalsausprägungen, bspw. als Ergebnis funktionaler Diversifikation. Eine Version entsteht hingegen aus einer Modifikation mit der Zeit und löst die vorangehende Version ab [We94].

Die Konfigurierung und der Entwicklungsprozess komponentenbasierter Anwendungen sollen im Folgenden durch die Varianten-Komponenten-Diagrammsprache und ein korrespondierendes Konfigurationsverfahren methodisch erweitert werden, indem die Aufbaustruktur einer „Familie“ von Anwendungssystemen (Anwendungssystem-Varianten) aus komplexen (zusammengesetzten) und einfachen Komponenten sowie zur Auswahl stehenden Komponenten-Varianten vorab modelliert und so der Auswahlprozess strukturiert, systematisiert und folglich beschleunigt wird.

### **2.1 Varianten-Komponenten-Diagrammsprache**

Ein Varianten-Komponenten-Diagramm stellt die Aufbaustruktur eines komponentenbasierten Anwendungssystems dar. Es basiert ursprünglich auf dem von Wedekind und Müller vorgestellten und durch Hümmer erweiterten Mereologischen Graphen als graphische Repräsentation einer Variantenstrukturstückliste [WM81, Hü04], die zeigt, wie in einem mehrstufigen Fertigungsprozess zunächst Bauteile zu Baugruppen kombiniert werden und diese über ggf. weitere Fertigungsstufen schlussendlich in das Enderzeugnis eingehen. Variantenstrukturstücklisten und mereologische Graphen haben sich in den klassischen Ingenieurbereichen zur Konstruktion und Fertigung variantenreicher, physischer Güter bewährt.

Übertragen auf die komponentenorientierte Anwendungsentwicklung erlaubt die Varianten-Komponenten-Diagrammsprache die Modellierung der Teil-Ganzes-Beziehungen einer Anwendung (Ganzes) und ihrer Komponenten (Teile). Abbildung 1 zeigt beispielhaft ein einfaches Varianten-Komponenten-Diagramm. Als Modellierungselemente werden zunächst vier grundlegende Knotentypen unterschieden:

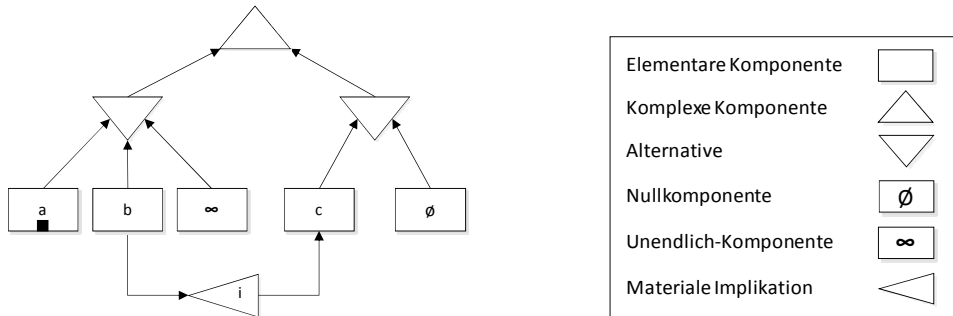


Abbildung 1: Beispiel eines einfachen Varianten-Komponenten-Diagramms (links) und graphische Symbole der Varianten-Komponenten-Diagrammsprache (rechts)

Einfache Komponenten, die nicht aus weiteren Komponenten zusammengesetzt sind und somit im Kontext der zu konfigurierenden Anwendung als elementar gelten, werden durch Knoten des Typs „*Elementare Komponente*“ symbolisiert. Hierbei ist zu beachten, dass „Elementare Komponente“-Knoten nicht unmittelbar eine Software-Komponente referenzieren, sondern die vollumfängliche, einem standardisierten Spezifikationsrahmen gehorchende [Ov06] Spezifikation einer Komponente, die bei der Modellierung ebenfalls anzugeben ist. Grundsätzlich werden Komponenten in der Modellierung durch ihre Spezifikation vertreten. Sie enthält wiederum – im Fall der elementaren Komponente – einen Verweis auf die tatsächliche, ausführbare Software-Komponenten, die einem Komponentenkatalog zu entnehmen sind.

Die *komplexe Komponente* setzt sich aus mindestens zwei, direkt in die Konjunktion eingehende Unterteile zusammen und entspricht der Baugruppe in einer klassischen Variantenstrukturstückliste. Die Teil-Ganzes-Beziehung zwischen Ober- und Unterteil wird durch eine gerichtete Kante vom Unterteil zum Oberteil dargestellt. Unterteile können elementare Komponenten, Konjunktionen oder Alternativen sein. Komplexe Komponenten verfügen über eine eigene Komponentenspezifikation. An der Spitze (in der Wurzel) eines Varianten-Komponenten-Diagramms befindet sich stets ein Konjunktivknoten, der an dieser Stelle die gesamte Anwendung repräsentiert.

Der dritte Knotentyp, der Alternativknoten („Alternative“), leistet einen wichtigen Beitrag zur Modellierung der Dynamik einer Anwendung. Die in ihn eingehenden Unterteile (elementare oder komplexe Komponenten) sind Varianten bzgl. eines oder mehrerer in der Spezifikation des Alternativknotens festzulegender Variationsmerkmale. Im Zuge der Konfigurierung darf nur genau eine Variante gewählt werden. Ihr gemeinsamer – invarianter – Spezifikationsanteil wird dem Alternativknoten zugeschrieben. Der Alternativknoten stellt somit einen geplanten Punkt der potenziellen

Veränderung in der Aufbaustruktur einer Anwendung dar. Die Kante zwischen Varianten und Alternativknoten sind vom Beziehungstyp Generalisierung.

Mit der *Nullkomponente* als möglichem Unterteil eines Alternativknotens können Kann-Alternativen modelliert werden. Ihre Auswahl ist gleichbedeutend mit dem Verzicht auf die zur Auswahl stehenden Varianten. Mit ihr können folglich Kann-Alternativen modelliert werden. Weiterhin ist es sinnvoll, eine Standard-Variante unter den Unterteilen eines Alternativknotens zu bestimmen. Sie geht in die Anwendung ein, wenn im Konfigurierungsprozess auf die explizite Entscheidung für eine Variante verzichtet wird. Ein schwarzes Quadrat markiert die Standard-Variante.

Zur Modellierung dynamischer Anwendungssysteme wird ein weiterer Knotentyp eingeführt werden: der *Unendlich-Knoten*. Im Gegensatz zu den bereits konstruierten elementaren Komponenten ist die an seiner Stelle in der Aufbaustruktur der Anwendung zu verwendende Komponente „frei“ durch den Konfigurator bestimmbar – es können potenziell unendlich viele Komponenten für ihn eingesetzt werden. Zum Zeitpunkt der Modellierung ist die noch unbestimmte Komponente also bereits in die Anwendung eingeplant, die Ausprägung wird aber erst zum (Re-)Konfigurierungszeitpunkt (siehe Abschnitt 2.2) *kreativ* entsprechend der dann gültigen Anforderung an die Anwendung bestimmt.

Der Unendlich-Knoten kann als Unterteil sowohl in komplexe Komponenten eingehen als auch zur Auswahl unterhalb eines Alternativknotens stehen (maximal ein Unendlich-Knoten). In letzterem Fall stehen damit nicht mehr nur die Komponenten zur Auswahl, die explizit in das Modell aufgenommen wurden, bspw. a oder b (vgl. fig. X), sondern ebenso das bei der Modellierung unbestimmte Dritte, repräsentiert durch den Unendlich-Knoten. Mit dem Unendlich-Knoten stellt die Varianten-Komponenten-Diagrammsprache eine Anwendung der konstruktiven Logik dar, die im Gegensatz zur klassischen Logik den Satz vom „tertium non datur“ aufhebt und lediglich den Satz vom ausgeschlossenen Widerspruch beansprucht [KL96].

Um Abhängigkeiten zwischen der Variantenauswahl bezüglich zweier unabhängiger Alternativknoten modellieren zu können, wird die *materiale Implikation* verwendet [We89]. Sie stellt die Beziehung zwischen einer oder mehrerer gleichzeitig gültiger Vorbedingungen (Antezedens) und einer Folgerung (Konsequens) dar. Im Beispiel in Abb. 1 muss bei Auswahl von Komponente b auch die Komponente c gewählt werden (materiale Implikation i). Rückwärtig wirkend darf dann, wenn Komponente c nicht ausgewählt werden kann, auch die Auswahl von Komponente b nicht möglich sein. Die materiale Implikation ist nur zwischen direkten oder indirekten Unterteilen unabhängiger Alternativknoten erlaubt. Unabhängig sind zwei Alternativknoten, wenn keiner der beiden Alternativknoten direktes oder indirektes Unterteil des anderen ist. Auf die Diskussion weiterer notwendiger Modellierungseinschränkungen soll an dieser Stelle verzichtet werden.

## 2.2 Konfigurierung einer Anwendung

Zur Konfigurierung eines komponentenbasierten Anwendungssystems müssen zum Prozessschema die in das System eingehenden Komponenten bestimmt und anschließend zusammengesetzt werden. Im ersten Schritt werden dazu die Alternativknoten eines Varianten-Komponenten-Diagramms entschieden.

Ausgehend von der Wurzel des Varianten-Komponenten-Diagramms wird dazu der Graph gemäß der Tiefen- oder Breitensuche traversiert. Wird hierbei ein Alternativknoten besucht, so fordert das Konfigurierungswerkzeug (vgl. Abschnitt 3) den Konfigurator (Anwender) auf, eine Entscheidung zugunsten einer der Komponentenvarianten zu fällen oder durch Verzicht auf die Entscheidung die Standardvariante zu wählen. Die Entscheidungsfindung wird unterstützt, indem die gemeinsamen Spezifikationsanteile der zur Auswahl stehenden Varianten und die Variationsmerkmale bereitgestellt werden. Im folgenden Beispiel (vgl. Abb. 2) ist als erstes der Alternativknoten 1 zu entscheiden, anschließend der Alternativknoten 2.

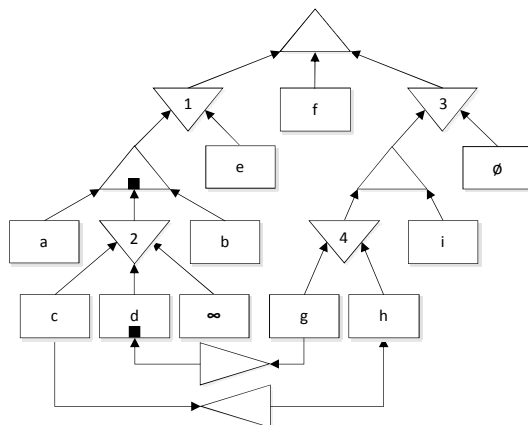


Abbildung 2: Beispiel eines Varianten-Komponenten-Diagramms

Wählt der Konfigurator einen Unendlich-Knoten (siehe Alternative 2 im Beispiel), so kann er zunächst frei in einem kreativen Prozess bestimmen, welche Komponente an dieser Stelle des Anwendungssystems zum Einsatz kommen soll. Dazu kann er beispielsweise die gewünschte Komponente aus einem Komponenten-katalog entnehmen oder neu, entsprechend der aktuellen Anforderungen konstruieren. Der „Platz“ der Komponente im Anwendungssystem ist aber schon festgelegt.

Die neu bestimmte Komponente wird dem Konfigurierungswerkzeug mittels ihrer Spezifikation bekanntgegeben. Das Konfigurierungswerkzeug muss dann eine *Konsistenzprüfung* bzgl. der bestimmten Komponente, die anstelle des Unendlich-Knotens tritt, vornehmen. Dazu wird die Spezifikation der Komponente mit dem festgelegten Spezifikationsanteil des Alternativknotens abgeglichen. Genügt die Spezifikation nicht den Anforderungen, so weist das Konfigurierungswerkzeug die Komponente begründet zurück und fordert den Konfigurator erneut auf, den Alternativknoten zu entscheiden. Kann das Werkzeug die Konsistenz nicht sicherstellen,



z.B. wegen unvollständiger Informationen, so fordert es diese Informationen entweder ein oder überträgt die Entscheidung an den Konfigurierer.

Die Entscheidung für eine Variante hat zur Folge, dass alle weiteren Varianten, die nicht mindestens einem weiteren Alternativknoten untergeordnet sind, samt ihrer untergeordneten Knoten aus dem Diagramm eliminiert werden. In der Folge müssen auch diejenigen Knoten entfernt werden, die als Antezedens in einer Implikationsbeziehung zu den zu eliminierenden Knoten stehen. Wird dadurch implizit ein weiterer Alternativknoten eindeutig entschieden, tritt der verbleibende Knoten anstelle dieses Alternativknotens. Die Wahl der Komponente c oder des Unendlich-Knotens würde so im Beispiel die Elimination der Komponente g bewirken und damit den Alternativknoten 4 entscheiden, an dessen Stelle die verbleibende Komponentenvariante h rückt.

Ist die gewählte Variante unterhalb eines Alternativknotens direkt oder indirekt – vermittelt ihrer nicht entscheidbaren Unterteile – Antezedens einer Implikationsbeziehung, dann wird auch automatisch zugunsten des Konsequens entschieden. Wenn es sich um eine Implikationskette handelt, also der Konsequens seinerseits Antezedens einer weiteren Implikation ist, dann muss die vollständige Implikationskette aufgelöst werden, bevor mit der Traversierung des Diagramms fortgefahren werden kann. Im Beispiel entscheidet die Wahl der Komponente c auch den Alternativknoten 4 und in der Folge den Alternativknoten 3. Würde hingegen der Unendlich-Knoten gewählt und eine neue Komponente n für ihn eingesetzt, so wäre zwar Alternativknoten 4 entschieden, nicht aber Alternativknoten 3 (vgl. Abb. 3).

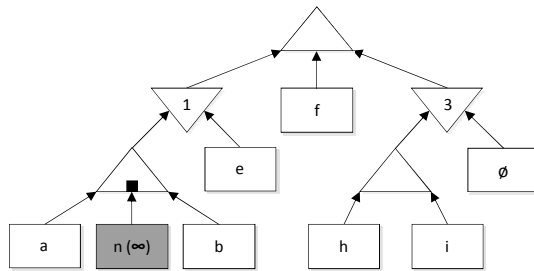


Abbildung 3: Teilweise aufgelöstes Varianten-Komponenten-Diagramm

Sind alle Alternativknoten eines Varianten-Komponenten-Diagramms aufgelöst, konkrete Komponenten für die Unendlich-Knoten eingesetzt und stehen somit alle Komponenten des Anwendungssystems fest, müssen diese zusammengesetzt werden. Sie können beispielsweise lose gekoppelt durch eine Process Engine im Betrieb gesteuert werden [LO09].

### 3 Werkzeugunterstützung der Konfigurierung

Zur Unterstützung der Entwicklung eines dynamischen komponentenbasierten Anwendungssystems unter Berücksichtigung von Varianten ist ein interaktives Werkzeug als Implementierung der vorgestellten Modellierungs- und Konfigurierungsmethode unverzichtbar. Insbesondere die materiale Implikation als Modellierungselement und die mit ihr in der Regel modellierten zahlreichen, ggf. auch transitiv wirkenden Abhängigkeitsbeziehungen erfordern eine *Validierung* des Varianten-Komponenten-Diagramms bestenfalls bereits während der Modellierung, um Widersprüche, Zyklen, unzulässige Beziehungen zwischen Knoten oder ähnliche Modellierungsfehler im Diagramm zu verhindern oder aufzudecken.

Neben der Validierung des Diagramms integriert das Werkzeug verschiedene Komponentenkataloge oder -marktplätze, aus denen Komponenten ausgewählt und in das Varianten-Komponenten-Diagramm, repräsentiert durch ihre Spezifikation, aufgenommen werden können. Die Komponentensuche und -auswahl wie auch das funktionale Matching zwischen Komponenten und Prozessaktivitäten und Teilprozessen werden durch normsprachliche Integration unterstützt. Dazu ist das Modellierungs- und Konfigurationswerkzeug an das zentrale Unternehmensrepository angebunden.

Der Konfigurationsprozess wird durch das Konfigurationswerkzeug gesteuert. Gemeinsam mit dem Konfigurier wird interaktiv [Nu07], wie in Abschnitt 2.2 beschrieben, das Varianten-Komponenten-Diagramm aufgelöst, indem alle Alternativknoten entschieden werden. Ist ein Alternativknoten zu entscheiden, fordert das System den Konfigurier dazu auf und unterstützt ihn mit der Bereitstellung der Komponentenspezifikationen unter Hervorhebung der abweichenden Merkmale sowie der Darstellung der Folgen einer Entscheidung für eine Variante, wenn sie in implikativer Beziehung zu anderen Komponenten steht. Wird ein Unendlich-Knoten gewählt, so prüft das Werkzeug nach Bestimmung der an seiner Stelle zu verwendenden Komponente die Konsistenz der daraus resultierenden Anwendung.

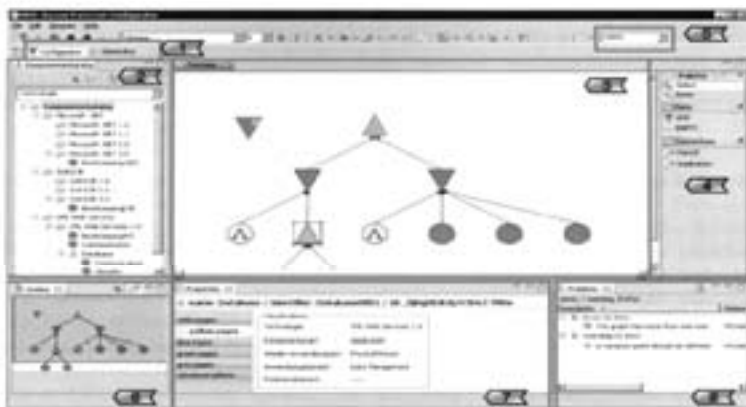


Abbildung 4: VPLC – Prototyp eines Konfigurierungswerkzeugs für komponentenbasierte Anwendungen

Wenn eine Rekonfiguration der Anwendung notwendig wird, weil Prozessänderungen dies beispielsweise erfordern, dann zeigt das Konfigurierungswerkzeug die betroffenen Komponenten im Varianten-Komponenten-Diagramm und ihre Verwendung an anderer Stelle (Verwendungsnachweis). Alternative Komponente-Varianten, Anpassungsmöglichkeiten und ihre Auswirkungen sind sofort ersichtlich.

Mit dem VPLC (vgl. Abb. 4) wurde am Fachgebiet Entwicklung von Anwendungssystemen der TU Darmstadt ein Prototyp des vorgestellten Modellierungs- und Konfigurierungswerkzeugs implementiert, der in ein umfassendes Anwendungssystem-Managementsystem [BGR05] einzubetten ist und gegenwärtig weiterentwickelt wird.

## 4 Zusammenfassung und Ausblick

Die Varianten-Komponenten-Diagrammsprache erlaubt die Modellierung der Aufbaustruktur komponentenbasierter Anwendungen unter expliziter Berücksichtigung von Komponenten-Varianten. Darüber hinaus können mit dem Unendlich-Knoten zum Modellierungszeitpunkt unbestimmte Komponenten in eine Anwendung eingeplant werden, die erst durch den Menschen, den Konfigurierer, kreativ zum Konfigurierungszeitpunkt tatsächlich eingebracht werden. Modellierung und (Re-)Konfigurierung werden interaktiv werkzeugunterstützt, indem bspw. das Modell validiert und während der Konfigurierung die Konsistenz der Anwendung nach dem hinzukommen neu bestimmter Komponenten – das kreative Element – geprüft wird.

Diese methodische Erweiterung zur prozess- und komponentenorientierten Entwicklung von Software-Anwendungen soll in der weiteren Forschungsarbeit auch auf die Arbeitsprozesse und das Wissen eines Unternehmens übertragen werden und so zur Weiterentwicklung des Enterprise Engineering beitragen. Entscheidend ist die Integration der Methode in ein umfassendes Anwendungssystem-Managementsystem.

## Literaturverzeichnis

- [AGO09] Amin, T.; Grollius, T.; Ortner, E.: Language-critical Development of Process-centric Application Systems. In (Cordeiro, J. et al. Eds.): ENASE 2008 Revised Best Papers, Lecture Notes in Communications in Computer and Information Science. Springer, Heidelberg, 2009, to be published.
- [BGR05] Berbner et al.: Management of Service-oriented Architecture based Application Systems. Enterprise Modelling and Systems Architecture, 2 (1) 2007, S. 14-25.
- [Hü04] Hümmer, W.: Vertragsverhandlungen um konfigurierbare Produkte im elektronischen Handel. Dissertation. Universität Erlangen-Nürnberg, Technische Fakultät, 2004.
- [KL96] Kamlah, W.; Lorenzen, P.: Logische Propädeutik: Vorschule des vernünftigen Redens, 3. Aufl. Metzler, Weinheim, 1996.
- [LO09] Link, M.; Ortner, E.: Organisational-oriented service architecture. Using the workflow management approach to control service system execution. In: Proc. of the 2009 International Conference on Engineering Management and Service Sciences (EMS 2009), IEEE Computer Society, Los Alamitos, 2009; to be published.

- [Nu07] Nussbaum, D. et al.: Discussion of the Interaction Concept Focusing on Application Systems. In: Proc. of the IEEE International Conference on Web Intelligence and Intelligent Agent Technology Workshops 2007, 5.-12.11.2007, Silicon Valley, California, USA. IEEE Computer Society, Los Alamitos; S. 199-203.
- [OEE08] Ortner, E.; Eller, B.; Elzenheimer, M.: Das elastische Unternehmen. Wie die heutige Informationstechnologie von dynamischen Unternehmen eingesetzt werden könnte. Arbeitsberichte des Fachgebiets Entwicklung von Anwendungssystemen, Nr. 2008/01, Technische Universität Darmstadt, 2008.
- [Or05a] Ortner, E.: Component-based Application Architecture for Enterprise Information Systems. In (Härder, T., Lehner, W. Eds.): Data Management in a Connected World. Springer, Heidelberg, 2005; S. 181-200.
- [Or05b] Ortner, E.: Sprachbasierte Informatik. Edition am Gutenbergplatz, Leipzig, 2005.
- [Ov06] Overhage, S.: Vereinheitlichte Spezifikation von Komponenten. Dissertation. Universität Augsburg, Wirtschaftswissenschaftliche Fakultät, 2006.
- [PP05] Pal, N.; Pantaleo, D.C. (Eds.): The Agile Enterprise. Springer, New York, 2005.
- [Sc05] Schulte-Zurhausen, M.: Organisation. Vahlen, München, 2005.
- [We89] Wedekind, H.: Konstruktionserklären und Konstruktionsverstehen. Elemente des Aufbaus eines Konstruktionsführungssystems. Zeitschrift für wirtschaftliche Fertigung und Automatisierung, 84 (1) 1989; S. 623-629.
- [We94] Wedekind, H.: Are the terms “version” and “variant” orthogonal to one another?, ACM SIGMOD Record, 23 (4) 1994; S. 3-7.
- [WM81] Wedekind, H.; Müller, T.: Stücklistenorganisation bei einer großen Variantenzahl. Angewandte Informatik, 23 (9) 1981; S. 377-383.





**Erster Workshop zur Zukunft der Entwicklung  
softwareintensiver, eingebetteter Systeme**

**(ENVISION 2020)**



## **ENVISION 2020: Erster Workshop zur Zukunft der Entwicklung softwareintensiver, eingebetteter Systeme**

Kim Lauenroth, Ernst Sikora

Universität Duisburg Essen, Software Systems Engineering  
Schützenbahn 70, 45127 Essen  
{kim.lauenroth; ernst.sikora}@sse.uni-due.de

### **1 Zielsetzung des Workshops**

Softwareintensive, eingebettete Systeme unterstützen den Menschen schon heute in vielen Bereichen des Lebens – sichtbar und unsichtbar. Beispielsweise verbessern sie im Automobil die Sicherheit, regulieren das Klima in Gebäuden oder steuern medizinische Geräte bis hin zu ganzen Industrieanlagen. Experten prognostizieren für die Zukunft eine rasante Zunahme softwareintensiver, eingebetteter Systeme.

Die Ausweitung des Funktionsumfangs und die zunehmende Vernetzung eingebetteter Systeme führen gleichzeitig zu einer rasanten Zunahme der Komplexität, die auch im Entwicklungsprozess Berücksichtigung finden muss. Existierende Vorgehensweisen und Methoden stoßen bereits unter den heutigen Rahmenbedingungen (z.B. Zeit- und Kostendruck) an ihre Grenzen. Existierende Ansätze und Methoden müssen aufgrund der wachsenden Herausforderungen in Frage gestellt und in Teilen neu konzipiert werden.

Der Workshop ENVISION 2020 verfolgt das Ziel, die Entwicklung und Diskussion zukünftiger Ansätze, Vorgehensweisen und Methoden zur Entwicklung softwareintensiver, eingebetteter Systeme zu fördern. Ein besonderes Augenmerk gilt dabei modellbasierten Entwicklungsansätzen.

### **2 Workshop-Organisation**

ENVISION2020 wurde organisiert durch die Innovationsallianz SPES2020, gefördert durch das BMBF, Förderkennzeichen 01IS08045. SPES2020 hat sich zum Ziel gesetzt, neue und innovative Ansätze und Methoden für die Entwicklung softwareintensiver eingebetteter Systeme zu erforschen und zu erproben.





Ausgehend von der Analyse verschiedener Anwendungsdomänen (bspw. Automotive, Avionik und Energie) soll ein zusammenhängender, domänenübergreifender Entwicklungsansatz entstehen. Dieser Ansatz soll anhand der Bedürfnisse der jeweiligen Anwendungsdomänen angepasst und in Zusammenarbeit mit den Industriepartnern erprobt werden. Siehe <http://www.spes2020.de> für weitere Informationen zu SPES2020.

Organisationskomitee:

- Manfred Broy, *TU München*
- David Cruz, *TU München*
- Martin Deubler, *TU München*
- Kim Lauenroth, *Uni Duisburg-Essen*
- Klaus Pohl, *Uni Duisburg-Essen*
- Ernst Sikora, *Uni Duisburg-Essen*

Programmkomitee:

- Reinhold Achatz, *Siemens*
- Klaus Beetz, *Siemens*
- Ottmar Bender, *EADS*
- Werner Damm, *OFFIS*
- Harald Hönninger, *Bosch*
- Stefan Jähnichen, *FIRST*
- Peter Liggesmeyer, *IESE*
- Ulrich Löwen, *Siemens*
- Dieter Rombach, *IESE*
- Wilhelm Schäfer, *Uni Paderborn*

Darüber hinaus waren Dominik Domis (IESE), Thomas Kuhn (IESE) und Mark-Oliver Reiser (TU Berlin) als externe Gutachter tätig.

### **3 Teilnehmer**

Die 30 Teilnehmer des Workshops stammten zu etwa gleichen Teilen aus Forschung und Entwicklung. Unter den Teilnehmern aus der Entwicklung befanden sich Vertreter verschiedener Anwendungsgebiete, darunter Automobilbau, Raumfahrt und Medizintechnik.

Die Teilnehmer waren insbesondere daran interessiert, aktuelle Trends in dem Forschungsgebiet kennen zu lernen, Einblicke in aktuelle Methoden zu erhalten sowie Anregungen für neue Entwicklungsansätze zu erhalten. Als Interessensschwerpunkte gaben die Teilnehmer die Gebiete Requirements Engineering, Architekturentwurf und Testen von eingebetteten Systemen an.

## 4 Zusammenfassung der Workshopvorträge und -diskussionen

Bei dem Workshop wurden sechs wissenschaftliche Beiträge vorgestellt und diskutiert. Des Weiteren stellte Kim Lauenroth als Mitglied der ENVISION2020-Organisation den Teilnehmern die vom Bundesministerium für Bildung und Forschung geförderte Innovationsallianz Software Plattform Embedded Systems (SPES 2020) vor.

Im Folgenden geben wir einen Überblick über die wissenschaftlichen Beiträge des Workshops und fassen die Diskussionen zu den einzelnen Beiträgen kurz zusammen.

### 4.1 Beitrag 1: Model-based Development Tools for Embedded Systems in the Industry – Results from an Empirical Investigation

In dem ersten Vortrag zum Beitrag „Model-based Development Tools for Embedded Systems in the Industry – Results from an Empirical Investigation“ stellte Thomas Kofler die Ergebnisse einer Industriebefragung über den Einsatz von Entwicklungswerkzeugen vor. Bei der Befragung wurde eine große Heterogenität an Modellierungswerkzeugen und eine meist proprietäre Werkzeugintegration festgestellt. Zudem wurde eine geringe Verbreitung der Generierung von Code aus Modellen und der formalen Verifikation von Modellen beobachtet.

In der Diskussion erfragten die Teilnehmer mögliche Einflussfaktoren für die gemachten Beobachtungen und regten eine vertiefende Untersuchung dieser Faktoren an. Zudem wurde vorgeschlagen, neben der Code-Generierung auch weitere Verwendungsarten der modellbasierten Entwicklung einzubeziehen, die auf Requirements Engineering und Testen ausgerichtet sind. Die beobachtete, unzureichende Werkzeugintegration wurde einem mangelnden Interesse der Hersteller und fehlenden Standards zugeschrieben.

### 4.2 Beitrag 2: [(Spacecraft Bus Controller) + (Automotive ECU)] / 2 = UltimateController

Im zweiten Vortrag “[Spacecraft Bus Controller) + (Automotive ECU)] / 2 = UltimateController” stellte Frank Dannemann eine Forschungsarbeit zur Integration eines systemnahen Software-Frameworks aus der Raumfahrt mit einer Service-orientierten Middleware aus der Automobilforschung vor. Das Vorhaben verknüpft Code-Generierung für sicherheitskritische Echtzeitsysteme und Service-Technologie zu einer Entwicklungsumgebung, die es Entwicklern ermöglicht, sich auf die Entwicklung der Applikationslogik zu konzentrieren. Die Arbeiten wurden anhand der Fernsteuerung eines Versuchsfahrzeugs durch ein mit Bewegungssensoren ausgestattetes Mobiltelefon illustriert und sollen auf Fahrerassistenzsystem ausgeweitet werden. Die Gegenüberstellung der Raumfahrt- und der Automobildomäne wurden in dem Beitrag von den Teilnehmern mit großem Interesse aufgenommen.

In der Diskussion wurden unterschiedliche Ziele und Rahmenbedingungen in den beiden Domänen herausgestellt, die eine Übertragung von Lösungen zwischen den Domänen erschweren, wie zum Beispiel der hohe Kostendruck in der Automobilbranche, der den Einsatz von redundanter Hardware zur Erhöhung der Ausfallsicherheit nur in sehr begrenztem Umfang zulässt. Des Weiteren erörterten die Teilnehmer den Einfluss der Systemkomplexität auf die Entwicklung und kamen zu dem Schluss, dass in beiden Domänen erhebliche Entwicklungsressourcen aufgewendet werden müssen, um die sehr hohe Systemkomplexität zu beherrschen.

#### **4.3 Beitrag 3: Ansätze zum Nachweis der Gleichwertigkeit von Software-Komponenten**

Der dritte Vortrag „Ansätze zum Nachweis der Gleichwertigkeit von Software-Komponenten“, gehalten durch Hans-Werner Wiesbrock, befasste sich mit dem Vergleich kontinuierlicher Signale, dem erwarteten und dem tatsächlichen Ausgabesignal, beim Testen von eingebetteten Systemen in der Medizintechnik. Ziel der Arbeiten ist es, durch einen Back-to-Back-Test von zwei Systemen die Gleichwertigkeit der Software dieser Systeme nachzuweisen. Das Verfahren soll es gestatten, eine Softwarekomponente in einem bereits zugelassenen System zu ersetzen, ohne dass eine vollständige Neu-Zertifizierung des Systems erforderlich wird.

In der Diskussion erörterten die Teilnehmer, in welchem Ausmaß sich der Austausch der Software auf die für ein System durchgeführten Sicherheitsanalysen auswirkt. Ein Teilnehmer wies auf vergleichbar gelagerte Fälle in der Luftfahrt hin, in denen eine Alt-Software gegen eine neue Software ersetzt wurde.

#### **4.4 Beitrag 4: Evaluation eines modellbasierten Requirements-Engineering-Ansatzes für den Einsatz in der Motorsteuerungs-Domäne**

In dem vierten Vortrag „Evaluation eines modellbasierten Requirements-Engineering-Ansatzes für den Einsatz in der Motorsteuerungs-Domäne“ stellte Ernst Sikora die Ergebnisse der Evaluation einer modellbasierten Requirements-Engineering-Methode in der Automobilbranche vor. Die Evaluation sollte Aufschluss über den Nutzen und die Anwendbarkeit der ziel- und szenariobasierten Methode COSMOD-RE bei der Entwicklung von Anforderungen für eine Motorsteuerungssoftware geben. Die Evaluationsergebnisse sollen in die Weiterentwicklung der Methode einfließen.

In der Diskussion erörterten die Teilnehmer mögliche Erweiterungen der in dem Vortrag skizzierten Methode. Unter anderem wurden eine Spezialisierung der Methode für unterschiedliche Projekttypen wie Neuentwicklung von Weiterentwicklung sowie eine dedizierte Betrachtung von Anforderungen, die aus der Sicherheitsanalyse in den RE-Prozess einfließen, diskutiert.

#### **4.5 Beitrag 5: A System for Seamless Abstraction Layers for Model-based Development of Embedded Software**

Der fünfte Vortrag "A System for Seamless Abstraction Layers for Model-based Development of Embedded Software", gehalten durch Eike Thaden, befasste sich mit einem Ansatz zur Strukturierung von Entwicklungsartefakten durch Abstraktionsebenen und Sichten. Das Ziel der vorgestellten Arbeit ist die Unterstützung eines durchgängigen, modellbasierten Entwicklungsprozesses über alle Abstraktionsstufen eines Systems hinweg.

In der Diskussion hinterfragten die Teilnehmer das Ziel einer vollständigen Durchgängigkeit und erörterten, inwieweit eine über alle Abstraktionsstufen durchgängige der Systementwicklung erreichbar ist. Als wesentliche Hindernisse für eine durchgängige Entwicklung wurden der bislang methodisch nicht gelöste Übergang von den Anforderungen zur Architektur und die Notwendigkeit von nicht strukturkonformen Verfeinerungen im Entwicklungsprozess angeführt.

#### **4.6 Beitrag 6: Eine erweiterte Systemmodellierung zur Entwicklung von softwareintensiven Anwendungen in der Automobilindustrie**

Im letzten abschließenden Vortrag „Eine erweiterte Systemmodellierung zur Entwicklung von softwareintensiven Anwendungen in der Automobilindustrie“ stellte Jan Meyer eine Echtzeiterweiterung für UML Zustandsmaschinen vor. Die vorgestellte Arbeit zielt darauf ab, die formale Spezifikation und Verifikation von Zustandsmodellen mit Zeiteigenschaften zu unterstützen.

In der Diskussion wurde die Abgrenzung der vorgestellten Arbeit zu existierenden Ansätzen und Werkzeugen erörtert. Zudem wurde die Skalierbarkeit des Ansatzes für umfangreiche Zustandsmodelle hinterfragt.

### **5 Ausblick und Danksagung**

Die Komplexität eingebetteter Systeme wird sich in Zukunft durch die Zunahme der Funktionsvielfalt und insbesondere durch den steigenden Vernetzungsgrad (auch über Systemgrenzen hinweg) weiter erhöhen. Bereits heute verzögert die hohe Komplexität zahlreiche Entwicklungen und verhindert innovative Neuentwicklungen teilweise sogar.

Entwicklungsansätze und -vorgehensweisen, die hohe Systemkomplexität beherrschbar machen, bilden für Unternehmen in allen Bereichen der Domäne eingebetteter Systeme einen entscheidenden Erfolgsfaktor und Wettbewerbsvorteil. Somit ist die Erforschung von Ansätzen und Vorgehensweisen zur Entwicklung softwareintensiver, eingebetteter Systeme ein bedeutendes Betätigungsfeld. Forschungseinrichtungen und Universitäten können in diesem Betätigungsfeld nicht nur einen Beitrag für den wissenschaftlichen Fortschritt leisten, sondern können zugleich durch Technologietransfer und den

Austausch mit der Industrie einen wertvollen Beitrag zur Sicherung und Stärkung der Wettbewerbsfähigkeit der Industrie erbringen.

Die Beiträge und Diskussionen bei ENVISION 2020 zeigen, dass die Forschung vielversprechende Konzepte anbietet und die Praxis ein großes Interesse an der Erprobung und Einführung dieser Konzepte hat. Doch obwohl das Gebiet in den letzten Jahren bedeutende Fortschritte erzielt hat, gilt es auf dem Weg hin zu einer durchgängigen, modellbasierten Entwicklungsmethodik für softwareintensive eingebettete Systeme noch zahlreiche offene Fragen zu klären und grundlegende Herausforderungen zu überwinden.

Die Resonanz der Teilnehmer zum ersten ENVISION 2020 Workshop war durchweg positiv. Alle Teilnehmer sprachen sich für eine Fortführung des Workshops in 2011 aus.

Wir sprechen an dieser Stelle ein herzliches Dankeschön an alle aus, die zum Gelingen dieses Workshops beigetragen haben. Ein besonderer Dank geht an die Organisatoren der SE2010, die ein hervorragendes Umfeld für ENVISION 2020 geboten haben.

# Model-based Development Tools for Embedded Systems in the Industry

## – Results from an Empirical Investigation –

Markus Herrmannsdoerfer, Thomas Kofler, Stefano Merenda, Daniel Ratiu, Judith Thyssen  
Institut für Informatik, Technische Universität München, 85748 Garching, Germany  
{herrmama, koflert, merenda, ratiu, thyssen}@in.tum.de

**Abstract:** Model-based development (MBD) is adopted in practice in different degrees. A prerequisite for building a new generation of tools for model-based development of embedded systems is to gain an overview over their current use and the needs of the industry. In this paper, we present the results of an empirical study—that we conducted in the context of the SPES2020 research project—about the use of model-based development tools in industrial projects. We investigated the following general research questions by collecting empirical data from several industrial project partners: (1) What is the status quo of the tools currently used in industry? and (2) What are the requirements for the next generation of industrial tools? The result of our study shows that there is a considerable heterogeneity in both the development tools and in the manner in which model-based development is perceived in different industrial branches (e. g., automation, avionics, automotive). Furthermore, we identified that the most important need for industrialization of model-based development in the future is a higher integration of different tools and methods.

## 1 Introduction

Model-based development promises to raise the abstraction level at which software development currently is performed and thereby to increase the quality and automation and to lower development costs. In the purest way, model-based development implies the pervasive use of adequate models in every development step and as main development means. In practice however, model-based development is adopted in different degrees. There is a wide spectrum under which model-based development is understood and applied in current industrial projects.

A prerequisite for a new generation of tools for model-based development of embedded systems is to have an overview over the current situation and the needs of the industry. In the context of the research project “Software Platform Embedded Systems” (SPES2020), we tackled this problem by performing a questionnaire-based empirical study. The main goal of this questionnaire was to get an overview over the situation and the needs of model-based development in industry today. In this questionnaire, we concentrate on questions regarding technologies and tools used to develop embedded systems. Although these questions focus on tooling issues, the results also reflect the methods used today. We conducted the questionnaire to investigate the following general research questions:

1. **What is the status quo of the tools currently used in industry?** From the tools currently used in industry, we aim at drawing conclusions about the methods used for the model-based development of embedded systems.
2. **What are the requirements for a next generation of industrial tools?** We aim at deriving requirements for a next generation of industrial tools.

According to our results, the most important issues today are that tools and technologies are considerably heterogeneous, that model-based development is not yet used to its full extent, and that existing tools are not yet fully appropriate for efficient model-based development. The most important need is the integration of existing tools for seamless model-based development. These results provide a valuable input for current and future academic research projects to ensure that the most important needs of the industry are taken into account.

The current paper is based on a deliverable written in the context of the SPES2020 research project [HM09]. It is part of a wider research area at the Chair for Software & Systems Engineering from Technische Universität München about tool architectures for seamless model-based development of embedded systems [BFH<sup>+</sup>10]. Our longer term goal is to take the status quo and requirements into account for the tool architecture which should be developed as part of the SPES2020 project.

**Outline.** In Section 2, we briefly present the design of our study. In Section 3, we present the results obtained after evaluating and interpreting the answers. We focus here mainly on qualitative results and only exemplarily present several quantitative data. In Section 4, we discuss threats to the results' validity, before we conclude in Section 5.

## 2 The Empirical Study

In this section, we provide an overview of the design of our study. To answer the research questions, we developed a questionnaire which was distributed to the industrial partners of the SPES2020 project.

**Subjects of the study.** We performed our study by distributing the questionnaire to all industrial partners of the project SPES2020. They originate from a wide variety of domains in the field of embedded systems ranging from automotive over avionics or industrial automation to energy. Furthermore, some of the partners play the role of suppliers, while others play the role of system integrators.

**Design of the questionnaire.** The questions from our questionnaire can be divided into the following categories:

- *Context.* The questionnaire should be filled out from the perspective of a certain project in which the respondent works. The questions in this category ask for some

information about the project and the respondent's role in the project.

- *Target platform.* This category contains questions about the target platform onto which the developed embedded system is deployed. We have identified the four aspects which characterize the target platform (i. e., programming language, operating system, bus system, and middleware).
- *Overview of today's tools.* This category contains questions about the status quo of tools currently used in industry. The questions are subdivided into the following two classes:
  - *Nature of tools.* This category concentrates on questions about the tools currently used in industry. It is further subdivided into the categories *vertical tools* and *horizontal tools*. *Vertical tools* focus on a special development phase – we were interested in tools used in requirements management, modeling, and quality assurance. *Horizontal tools* are necessary for all development phases – we were interested in cross-cutting tools like those that belong to the infrastructure, tools for configuration management, change management or process support.
  - *Availability of tools.* This category concentrates on the questions whether the needed tools are available on the market and what the industry does in order to achieve the wanted tool functionality.
- *Deficits of today's tools.* This category contains questions about problems with the current tools and about requirements for future tools. We concentrate on the missing features and their impact, as well as the current and desired integration of the used tools.

**Types of questions.** In the questionnaire, we asked three different types of questions: (1) *One possible answer.* In these questions, we provided a number of answers from which the respondents should choose one. For missing answers, we added the additional category *no answer*. (2) *Multiple possible answers.* In these questions, we provided a number of answers from which the respondents could choose multiple ones. Furthermore, we included a category *other* through which the respondents could name additional answers. (3) *One possible answer for different aspects.* In these questions, the respondents should choose one answer for different predefined aspects.

**Execution.** To answer the research questions, we developed the questionnaire which consists of 25 questions in total. We built a first version of the questionnaire which was presented to a small number of project participants. Based on the feedback we got, we improved the questionnaire. The second version of the questionnaire was distributed via mailing list to all industrial partners. We altogether received 24 completely filled out questionnaires.



### 3 Results of the Empirical Study

In this section, we discuss the results of the questionnaire in terms of the research questions posed at the beginning of this paper. In Section 3.1, we discuss research question 1 – i. e., to what degree model-based development is adopted in industry today. In Section 3.2, we discuss research question 2 – i. e., which tooling deficiencies hamper the pervasive use of model-based development and what are the requirements for a next generation of industrial tools. Due to the limitation of space, we mainly present qualitative results. The detailed quantitative results of the questionnaire can be found in [HM09].

#### 3.1 What is the status quo of the tools currently used in industry?

Following our study, when looking at the current status of model-based development tools in industry, we make the following qualitative observations:

**Heterogeneity of target platform.** The target platform onto which the embedded system is deployed has a considerable heterogeneity. The target platform varies a lot from project to project in terms of both programming language, bus system, operating system, and middleware. It is important to note that this heterogeneity is not only between different companies, but also within a single company. As a consequence, the development tools are also affected by heterogeneity, as each of them is usually specialized for a certain target platform.

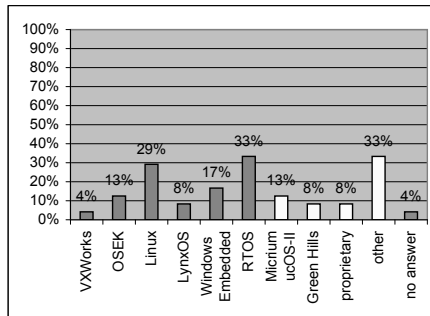


Figure 1: Which operating systems are running on the developed system? (Multiple possible answers.)

To characterize the target platform, we investigated the four aspects programming language, operating system, bus system, and middleware. Exemplary, Figure 1 shows the distribution of the answers to the question which operating systems are running on the developed system. The most widely used operating systems are real-time operating systems (RTOS) with 33%, Linux with 29%, and Windows Embedded with 17%. 33% of the respondents employ *other* operating systems like Embedded SW, MACOS (Thales), Microware OS-9 RTOS, Simatic, Simatic TDC, Sysgo (Pike OS), Timer Interrupt Driven

Proprietary Software Sequencer, TinyOS, Windows XP, and a simple scheduler. The extremely huge number of about 20 different operating systems clearly illustrates how heterogeneous are the platforms addressed by the development of embedded systems.

**Heterogeneity of tools.** There is also a big heterogeneity in the tools that are used for the model-based development of embedded systems. This holds for both vertical tools, i. e., tools which are specific to a certain development phase, and horizontal tools, i. e., tools which are necessary for all development phases, like version control systems or change management tools. Again, this heterogeneity is not only between different companies, but also within a single company. The heterogeneity of tools hampers their integration.

The results show that horizontal tools vary a lot in terms of the tool platform as well as for the activities of version and change management. Vertical tools vary a lot for the activities of requirements management, modeling, and quality assurance.

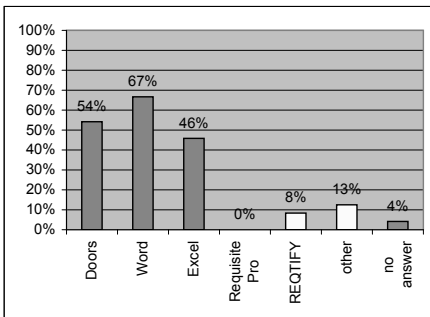


Figure 2: What tools are used for requirements elicitation and management? (Multiple possible answers.)

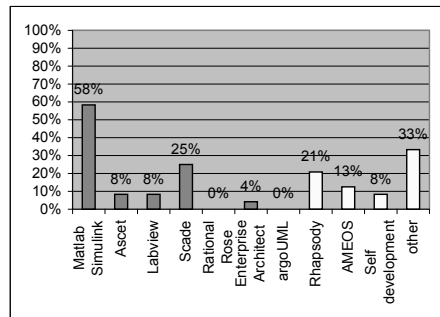


Figure 3: What modeling tools are used? (Multiple possible answers.)

Figure 2 shows the distribution of the answers over typical tools for requirements elicitation and management. Even though they are not tools tailored for requirements management, Microsoft Word and Microsoft Excel are still the most widespread tools with a share of 67% and 46%, respectively. However, Doors is used as a specific requirements management tool in more than 50% of the answers.

Figure 3 shows the share of typical modeling tools in the answers. With a usage rate of 58%, Matlab Simulink is the most widespread modeling tool—followed by SCADe with 25%. The UML and SysML tool Rhapsody is only on the third position, with a share of 21%. Besides Rhapsody, there are other UML tools used in industry like AMEOS and Enterprise Architect. 33% of the respondents named *other* modeling tools like MID Innovator, NX, PCS7 Engineering, radCase, SiGraph, Solid Edge, TAU, Teamwork, UML, X32 BlueRiver, and “not specified”. The high number of employed modeling tools shows how heterogeneous the tool landscape for modeling is.

**Weakly defined models.** Tools which are based on models without a precise meaning are still widely used in industry. For requirements management, the industry still relies a

lot on Microsoft Word and Excel which are general purpose and thereby not completely adequate for modeling requirements (cf. Figure 2). The situation is much better for system modeling with a high usage of modeling tools such as Matlab Simulink and Scade (cf. Figure 3). Models with a precise meaning can be better analyzed and the tools can offer a higher automation. Consequently, there is still much opportunity for improving the appropriateness of tools in order to work with semantically rich models and thereby to benefit from the promises of model-based development.

For process support, a lot of tooling is based on MS Project and Excel – with a share of 79% and 63%, respectively, as illustrated in Figure 4. However, both tools allow the developers only to define the process in an informal way and thus do not allow automation.

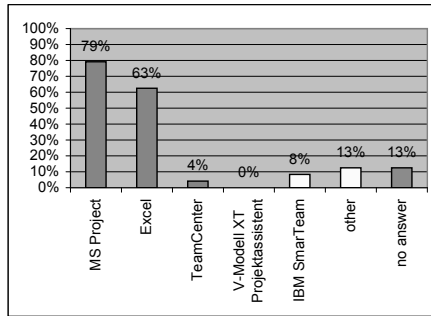


Figure 4: What tools do you use for process support? (Multiple possible answers.)

**Sporadic generation of code.** The tool chains used in practice do not yet provide the full benefits of model-based development. Even though the industry uses a lot of advanced modeling tools like Matlab Simulink and Scade, a lot of code is still handwritten instead of being generated from high-level models. As illustrated in Figure 5, only 16% of the respondents generate from 80% to 100% of the code. 34% generate from 40% to 70% of the code, and 42% generate only 0% to 30% of the code. An interesting point is, however, that there are already some companies that claim 100% code generation in certain projects.

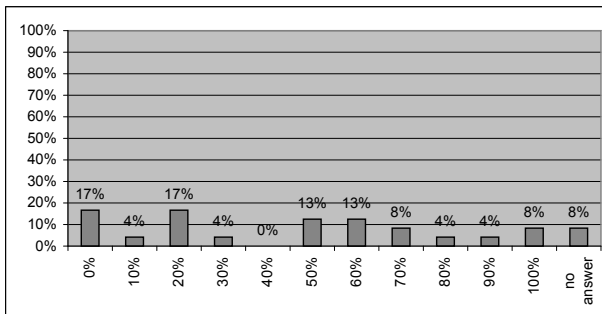


Figure 5: How large is the fraction of code generated from models? (One possible answer.)

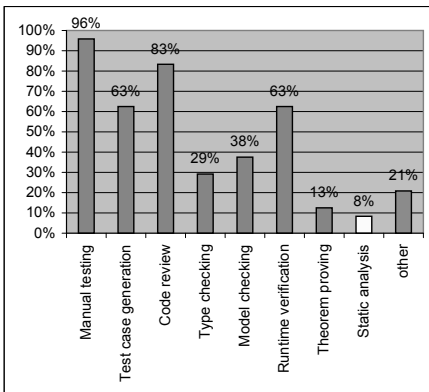


Figure 6: Which methods do you use for quality assurance? (Multiple possible answers.)

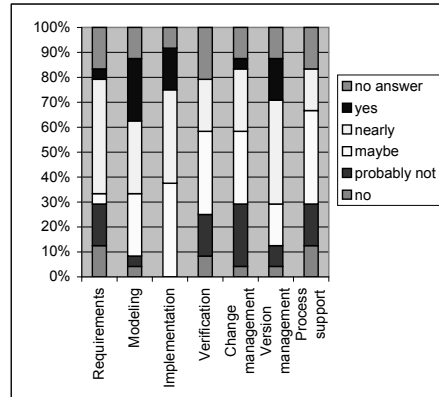


Figure 7: Do you think that you use the best tools available on the market? (One possible answer for each aspect.)

**Predominance of manual quality assurance.** For quality assurance, the industry still uses a lot of manual techniques like manual testing and code review. As illustrated in Figure 6, nearly every project performs manual tests and code reviews for quality assurance (96% and 83%, respectively). Automatic techniques for model-based quality assurance like model checking and theorem proving are not yet widely adopted in industry. This may be due to the fact that appropriate tools are not available on the market, or the respondents cannot use them due to other restrictions, like e. g., the missing interoperability with the current tool chain.

**Inability to use best-in-class tools.** A lot of respondents to the questionnaire are not able to use the best tools available on the market. More specifically, the results show that only 15% of the respondents think that they use the best tool (average over all process phases, cf. Figure 7).

A reason for this may be the high barriers to switch to another tool because of the high migration costs. For instance, migration effort is necessary for migrating existing data as well as for training developers to use the new tool. Sometimes, it may be even impossible to switch to another tool, as the new (and presumably better) tool may not integrate well within the existing tool chain.

**High rate of proprietary tools.** The industry makes use of a lot of proprietary tools (instead of commercial-off-the-shelf (COTS) tools). In particular, the results show us that proprietary tools are developed for all process phases. Although a huge number of development COTS tools are available, the market does not seem to provide solutions satisfying the requirements of individual projects. This may be also due to the missing customizability of existing tools to the individual requirements. In particular, most effort is spent on the integration of existing tools which is very important for seamless model-based development.

### 3.2 What are the requirements for a next generation of industrial tools?

Following the results of our questionnaire, when discussing the major needs of future development tools, we can make the following observations:

**Importance of tool integration.** A deep integration of the development tools is seen as the major issue. Most effort for proprietary tool development is spent on coupling of existing tools. This is consistent with the result that—with the share of 71%—tool integration is the most missing feature of today’s tools followed by process support with 42% (cf. Figure 8).

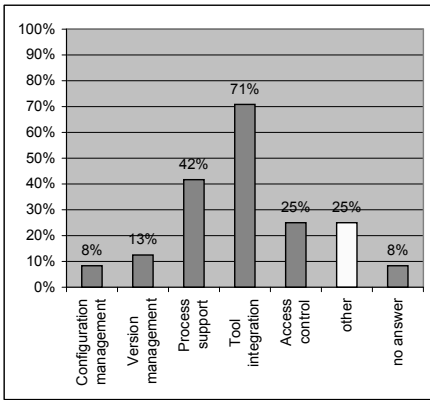


Figure 8: What features do you miss in the tools you use today? (Multiple possible answers.)

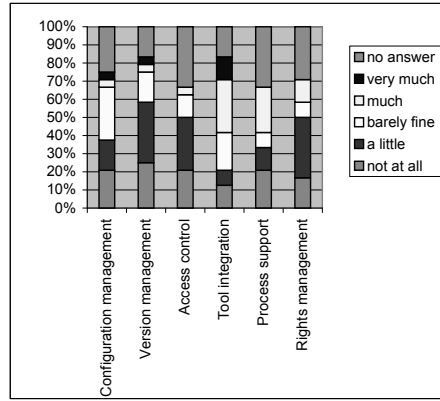


Figure 9: How far is your work restricted by the missing features? (One possible answer for each aspect.)

Moreover, the missing integration of the tools heavily restricts the engineers in their daily work. Figure 9 shows the distribution of the respondents’ rating with respect to the different activities in which tool support is required. The numbers show that the major restrictions are caused by the missing integration of tools: More than 40% of the respondents voted that the missing tool integration restricts their work much or even very much.

**Importance of process support.** Support of the development process is seen as an important feature which is not fully satisfied by current tools. Currently, the respondents mainly use Microsoft Project and Excel for process support (cf. Figure 4), but at the same time do not think that they use the best tools on the market (cf. Figure 7). However, the results show that they do not invest too much effort into the development of proprietary process support. This is probably due to the fact that good process support also requires tool integration which is also missing today. After tool integration, the respondents see process support as the most important missing feature (cf. Figure 8). The missing process support restricts a lot of respondents in their daily work (cf. Figure 9).

**Back end integration is more important than front end integration.** Integrating the back end is seen to be much more important than integrating the front end. In particular,

back end integration techniques like a central repository, tool couplers and consistency checks are considered as highly important, whereas integration of the user interface is not as important (cf. Figure 9). This seems to be the right prioritization, since without an integrated back end an integration of the front end cannot provide significant advantages.

## 4 Threats to Validity

We are aware that our results can be influenced by various threats to validity. They can be divided into threats to internal (design of the questionnaire) and external (representativeness of the results) validity.

**Internal validity.** The results might be influenced by the design we chose and the way we posed the questions for the questionnaire.

*Implicit assumptions in questions.* We make implicit assumptions about what is the optimal scenario of model-based development (e. g., full code generation and full tool integration are the optimal solutions, . . .). However, these assumptions are in line with the paradigm and the promises of model-based development.

*Priming through predefined answers.* The selection of the possible answers may influence the result of the questionnaire. To mitigate this threat, we provided a field to specify *other* answers.

*Different understanding of terms.* The respondents may have a different understanding of the terms we used in the questions (e. g., “tool platform”, “tool integration”, . . .). To mitigate this threat, we tried to use commonly understood terms as far as possible.

**External validity.** The results might be influenced by the fact that we questioned only a subset of the possible developers of embedded systems and might thereby not be representative for model-based development of embedded systems in general.

*Number of respondents.* We received only 24 filled out questionnaires which poses a great threat to the results’ validity. However, the obtained answers come from all project sizes, different levels of project success, different domains where embedded systems are developed, and all process phases.

*Distribution of respondents.* The respondents were not equally distributed over all participating industry partners – taking their size into account. We received more filled out questionnaires from smaller companies, but fewer from bigger ones. However, we can still derive that the tools and technologies used for the development are quite heterogeneous today.

*Locality of respondents.* All the questionnaires were filled out by respondents located in Germany – threatening the transferability of our results to model-based development in other countries. However, we also had respondents working in companies that act globally as well as in globally distributed projects.

## 5 Conclusion

In this paper we presented the results of an empirical study to better assess the current situation of model-based development of embedded systems in the industry. According to our two research questions, we are able to come to the following conclusions:

Firstly, we found a considerable heterogeneity in the development tools as well as the target platform onto which the software is deployed. This heterogeneity makes it difficult to seamlessly integrate the tools and methods for model-based development. The results also show that model-based development is currently far from being used to its full extent in industry. Partially, only weakly defined models are used which makes it impossible to obtain a high automation and to employ advanced techniques for model-based quality assurance. Additionally, high migration costs from one tool to another hamper the usage of the best tools on the market.

Secondly, we wanted to identify the most important needs for model-based development in future. Our results show that the integration of tools and methods is considered by engineers as the most promising way to more efficient model-based development. The integration of the tools is not only important on the level of the models, but also on the level of the process. However, seamless integration on the higher levels (e. g., the process) requires integration on the lower levels (e. g., the models).

The success of model-based development of embedded systems in future depends on the deep integration of tools and methods. This includes the adaptability of tools and the flexibility of methods to support the overall process of developing embedded systems in an appropriate way.

Many tools used by our participants have quite some functionality in common, e. g., persistency, configuration management, and so on. A possible solution to integrate these tools is to build an integrated tool platform that factors out this common functionality [BFH<sup>+</sup>10]. Our long term goal is to provide the necessary concepts for building such an integrated tool platform. This integrated tool platform should be extensible with new modeling languages and should provide a mechanism to integrate these languages. To reduce the effort, modeling languages themselves should also be built using model-based development by defining explicit metamodels. Moreover, an integration of the modeling languages enables the tool platform to provide process support throughout model-based development.

## References

- [BFH<sup>+</sup>10] Manfred Broy, Martin Feilkas, Markus Herrmannsdoerfer, Stefano Merenda, and Daniel Ratiu. Seamless Model-based Development: from Isolated Tools to Integrated Model Engineering Environments. *Proceedings of the IEEE - Special Issue on Aerospace & Automotive*, to appear in 2010.
- [HM09] Markus Herrmannsdoerfer and Stefano Merenda. Result of the Tool Questionnaire. Technical Report TUM-I0929, Technische Universität München, 2009.

$$\frac{(\textit{Spacecraft Bus Controller}) + (\textit{Automotive ECU})}{2} = \textit{Ultimate Controller}$$

Sergio Montenegro, Frank Dannemann, Lutz Dittrich, Benjamin Vogel  
DLR Institute for Space Systems  
Dpmt. Central Avionics  
Robert-Hooke-Str. 7  
D-28359 Bremen

Sergio.Montenegro@DLR.de, Frank.Dannemann@DLR.de, Lutz Dittrich@DLR.de  
Benjamin.Vogel@DLR.de

Ulf Noyer, Jan Gacnik, Marco Hannibal, Andreas Richter, Frank Köster  
DLR Institute of Transportation Systems  
Dpmt. Automotive  
Lilienthalplatz 7  
D-38108 Braunschweig

Ulf.Noyer@DLR.de, Jan.Gacnik@DLR.de, Marco.Hannibal@DLR.de,  
Andreas.Richter@DLR.de, Frank.Koester@DLR.de

**Abstract:** In many cases, similar challenging problems arise in different domains and are often solved with an only limited focus. Synergy effects between different research fields can lead to an enormous technical improvement. This is illustrated by combining experiences and competencies of the institute of Space Systems and the institute of Transportation Systems of the German Aerospace Center (DLR). Their cooperation/collaboration aims at more reliable and highly available platforms for embedded (software/hardware) systems. Both institutes have developed software frameworks to be used in embedded systems of their field of work. In this paper we present an approach of integrating an embedded real time OS to be used in satellites into the DOMINION implementation-platform, which is used for the specification of embedded in-vehicle software. Finally a safety critical application with hard real time requirements will be chosen as a use-case to demonstrate the combination of aerospace and automotive technologies.

## 1. Introduction

As pointed out above, similar challenges in the context of embedded systems related research are currently to be focused in the space as well as in the automotive domain, e.g. dependability, reliability, availability, and safety of the built systems has to be improved. The fast growing complexity of the systems leads to the famous "design-gap" which has to be handled with a more maintainable and structurable design. Re-usability of requirement analysis, designs, and implementations (e.g. as an intellectual property (IP) core) lead to faster time to market. If the system can heal itself by being adaptable to changing circumstances (self-x-properties), it can be used in unknown or at least unpredictable environments, i.e. the space or a car engine.



Besides sharing similar problems, similar trends can be determined. The growing trend of software driven innovations in the system design emerges as using a large share in the development costs (up to 70%). While the automotive industry is already fighting the complexity to be handled and is trying to reduce it, space industry only recently realizes the growing complexity. With the planned technology transfer, the complexity collapse could be avoided.

Both above mentioned domains have different approaches and views of common problems and have reached different maturity in different fields. The following table shows some of these differences and shows which branch may profit from the achievements reached by the other one (see the transfer arrow).

<b>Automotive</b>	<b>Transfer</b>	<b>Space</b>
Real industrial production (thousands of units per day)	→	Artisan level (hand-craft): each unit hand made, and unique (thousands of days per unit)
Europe has a leading role	→	Europe is not in the cutting edge
Good reuse practice	→	Hardly reuse
Scarcely fault tolerance	←	Widely spread fault tolerance: No single point of failure is allowed
Safety/dependability relies mainly on mechanic fall back mechanisms and on the driver	←	All safety/dependability measures are based on autonomous reactions of the avionics (no mechanic/operator fall back mechanisms)
Traditionally based on mechanics	←	Electronic and software based from its origin
One important optimisation factor is the cost.	→	Cost was not important. Now changing, but lack of experience.
Power, mass and volume were not important. Now changing but lack of experience.	←	Design optimized for lowest resources: power, mass, volume
Interoperability between different suppliers and between main contractor (e.g. AUTOSAR)	→	No interoperability between suppliers.

Complexity management	← →	Complexity Management
Active steps to reduce the complexity (e.g. reduce number of ECUs: AUTOSAR)	→	No active steps to reduce the complexity
Standardized communication, e.g. CAN	→	No global accepted standard. Each device has its own protocol. Now one standard is emerging: SpaceWire, but it is too complex to be accepted widely.
Complexity is already out of control		Complexity is becoming out of control
Experience on safety critical hard real-time systems	→	Time violations are not considered to be mission critical
Experience on handling sensors failures and errors due to environment conditions	→	Systematic sensors errors/failures are not a central theme of fault tolerance
Transient computation and storage errors are not a central theme for fault tolerance	←	Experience on handling transient computations and storage (CPU + MEMORY) errors, due to radiation effects.
Experience using diversity to avoid systematic errors (especially for sensoric)	→	Diversity is not a central theme for dependability
Large experience on quality management (but different than space)	← →	Large experience on quality management (but different than automotive)
Large experience on system tests (but different than space)	← →	Large experience on system tests (but different than automotive)

Both domains can learn from each other about quality management, systematic tests and complexity management because both have totally different solutions for the same kind of problems. In the other fields one has more experience than the other one, and in one field (getting rid of complexity) both are having the same problem. In any case both can profit from sharing experiences and competencies.

## **2. Partners**

### **2.1 The DLR Institute of Space Technology**

The DLR Institute of Space Technology ([www.dlr.de/irs](http://www.dlr.de/irs)) has the aim to investigate and evaluate complex astronautic systems in the context of space research given consideration of technological, economic as well as socio-political and security aspects. Furthermore, the dynamic team of employees at the institute develops concepts for innovative space missions on national and international levels. The focal points of the working groups are: system analysis for space transport systems and space segments, system technology (rover wheels, satellite data management, and attitude control, cryogenic rocket propulsion systems) and operation of test and simulation facilities.

The Department Central Avionics develops software and hardware for the core avionics of a spacecraft (avionics, excluding devices). The central element in our architecture is a network to interconnect devices, computers, storage units, and radio communication units. Usually devices are attached directly to the central computer, in our architecture the computer has only link(s) to the network and all devices are attached to the spacecraft network (SAN).

### **2.2 The Institute of Transportation Systems**

With basic research activities and numerous projects with industrial as well as scientific partners the institute of Transportation Systems ([www.dlr.de/ts](http://www.dlr.de/ts)) aims for improving safety and efficiency of road and rail transportation. Therefore, scientists from different disciplines like engineering, psychology, and computer sciences strive for innovative solutions for automotive and railway systems as well as for future traffic management.

Beside activities in the institute's departments of Railway Systems and of Traffic Management, the human-centered design of advanced driver assistance and automation systems is the major topic of the institute's automotive department. This and other research activities are supported by elaborated research facilities: With the DLR ViewCar® driving in real traffic is observed with regard to understand driver behavior and the dynamics of surrounding traffic. Based on such results assistance and automation systems are developed to support the driver in an adequate manner and to prevent errors as well as accidents. First prototypes are usually implemented in a virtual reality laboratory and other simulation facilities for fast and efficient testing and evaluation. If necessary, a special dynamic driving simulator can be used for further evaluation steps. Experimental vehicles (FASCar-I/II) are available to demonstrate proposed concepts of assistance and automation in reality.

For different purposes technologies from the aerospace sector are adapted. That includes e.g. image processing, location services, and control- as well as planning algorithms. For an effective integration of functions flexible and modular system architectures are being developed.

### 3 RODOS - Building Blocks Execution Framework

In the first step, the Real Time Onboard Dependable Operating System (RODOS) [3] is provided by the DLR Institute of Space Technology to the joint venture between the space and automotive industry. In the near future, network technologies, and hardware implementations of middleware's may be included to the joint venture.

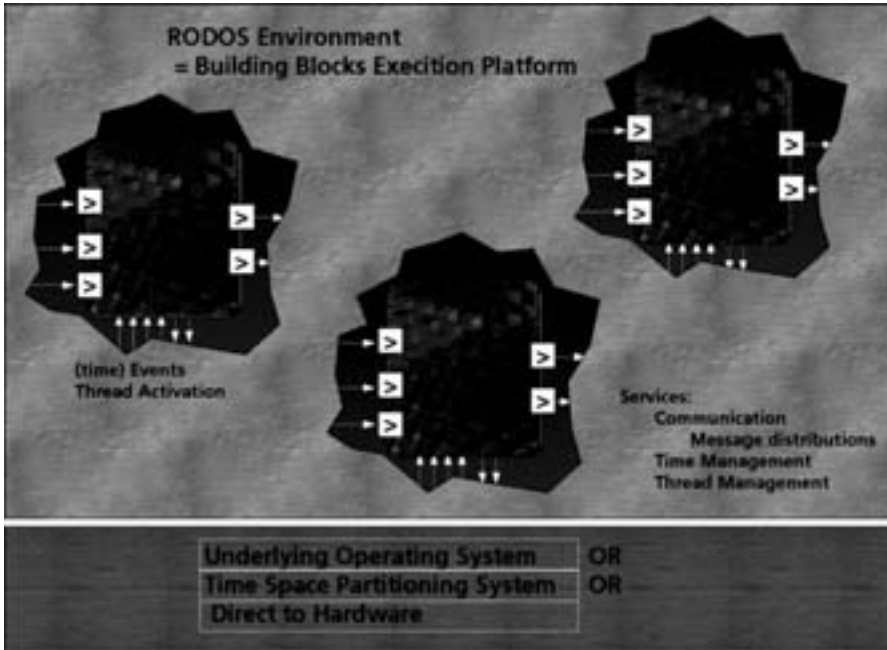


Figure 1: The RODOS Environment executes S/W building blocks.

RODOS is a building block execution platform (see figure 1) designed for space applications and for applications demanding high dependability. Simplicity is our main strategy for achieving dependability, as complexity is the cause of most development faults. The system was developed in C++, using an object-oriented framework simple enough to be understood and applied in several application domains. Although targeting minimal complexity, no fundamental functionality is missing, as its microkernel provides support for resource management, thread synchronization and communication, input/output, and interrupts management. The system is fully preemptive and uses priority-based scheduling and round robin for threads sharing the same priority level.

The execution platform provides a (software) interconnection network between applications / building blocks. A building block requires some services (incoming messages) in order to be able to provide other services (outgoing messages). The execution platform RODOS distributes such services (messages) from producer to consumers.

RODOS may be executed on top of other operating systems or TSP (Time Space Partitioning) systems, or directly on the hardware in case no other operating system is running on the target hardware. In all cases, the interfaces to the building blocks (or applications) remain the same, and a network of applications may be executed on different platforms and operating systems without modifications.

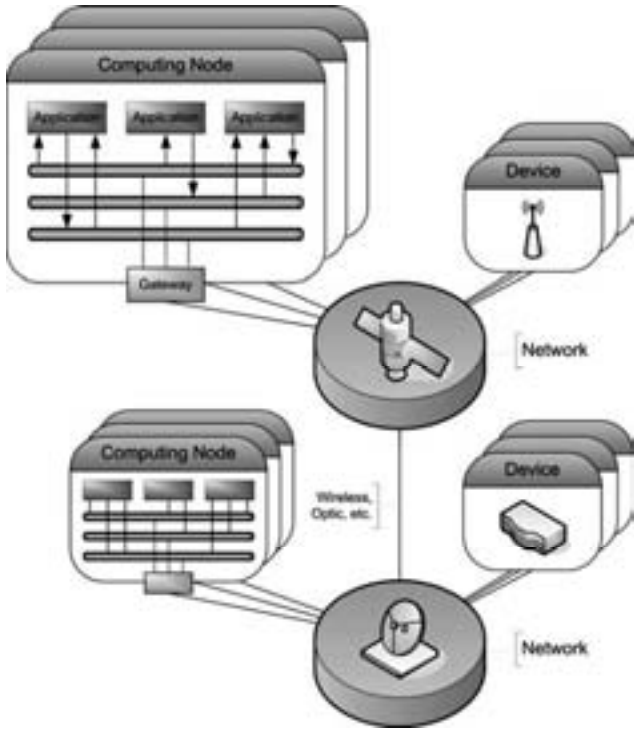


Figure 2: Example of the network centric network

RODOS provides a middleware which carries out transparent communications between applications and computing nodes. The messages exchange is asynchronous, using the publisher-subscriber protocol. Using this approach, no fixed communication paths are established and the system can be reconfigured easily at run-time. For instance, several replicas of the same software can run in different nodes and publish the result using the same topic, without having to know each other. A voter may subscribe to that topic and vote on the correct result. The middleware core distributes messages only locally, but using the integrated gateways to the “NetworkCentric” network [2] [3], messages can reach any node and application in the network (see figure 2). The communication in the whole system includes software applications, computing nodes and IO devices.

Our interconnection links in the network define a very simple message distribution protocol. Each message in the network receives a topic tag which identifies its content and purpose. Examples of topics could be: position, temperature, and attitude. Services will be published as topics regardless of whether they are produced by software tasks or by hardware devices. Any application or device may subscribe to the topics which it needs to help to produce its services.

This method is used to interconnect service providers and consumers including hardware and software. All our devices and software components provide this interface. In order to be able to attach COTS (Commercial Off The Shelf) devices (with their own protocols) to the network, the network provides the required interfaces and protocol converters. The COTS devices receive and send their own messages, and then the protocol converters translate them into our internal “universal language”. The network will perform all required transformations in order to make the message transport transparent.

#### **4. Implementation-Platform DOMINION**

DLR Institute of Transportation Systems has defined an architecture for development of in-vehicle services called DOMINION [1]. The step especially seems reasonable with the current standardization of Intelligent Transport Systems (ITS) technologies, including wireless communications (C2X – car to car, and car to infrastructure communication) and related services (e.g. Internet enhanced navigation, hazard warnings etc.).

Part of DOMINION uses modified versions of business standards, like VSDL (in-Vehicle Service Description Language) and VPEL (in-Vehicle Process Execution Language) . This concept integrates model-driven aspects like code-generation for real-time targets as well as the deployment of very flexible SOA services. The focus of the development of in-vehicle services is in the context of assistance and automation systems. A lightweight, platform independent (currently Windows, Mac OS X, iPhone OS, and RT-patched Linux are supported) DOMINION runtime environment has been developed, providing a flexible test environment of time-critical services. By using service descriptions, the description of communication and data structures etc. are independent of the programming language. A built-in code generator generates source code for heterogeneous platforms and programming languages. This allows deploying DOMINION based software on various environments and makes DOMINION a valuable approach for testing in different research facilities like simulators or even entire vehicles. Furthermore, to integrate the DOMINION runtime environment and SOA technology, DOMINION services might be deployed into both worlds by configuring the bindings of ports and operations of a service. Moreover, DOMINION offers general development services, especially when dealing with recording and storing data from testing assistance and automation systems.

## 5. Merging Automotive and Space Technology

The first step in our joint efforts is a software rendezvous: Generating RODOS applications out of DOMINION's semi-formal service descriptions.

Model driven development including high-level formal specification of services is more and more used within the automotive domain, which abstracts from low-level target implementations. Using DOMINION, services are specified in a simple, reproducible way, independent from implementation details.

In the aerospace domain, the complementary part to this approach is located. RODOS is an execution platform which handles real hardware of embedded systems taking into account all possible limitations and restrictions of real resources like memory, CPU performance, power consumption etc. RODOS targets dependability even if running on non-dependable components, while the fault tolerance mechanisms are transparent to the applications. These fault tolerance mechanisms can be added to the DOMINION applications without having to modify them. The available redundant hardware resources will be managed by RODOS to obtain the maximum of possible redundancy at any moment without intervention of the DOMINION applications.

A demonstrator which will be controlled by the RODOS-DOMINION pair is developed by the two DLR institutes (see Chapter 5.1 for details). During the development of the demonstrator, the shared problems can be handled by the know-how of both institutes. In the joint-venture, a fault tolerant embedded computer running RODOS is combined with the automotive applications specified in DOMINION. The code generator of DOMINION will be adapted to generate C++ code for RODOS. The DOMINION specification only contains a complete description of the service interface and its timing aspects, whether functional aspects nor low-level dependencies and restrictions are included. Fault tolerance aspects will be added by RODOS in a transparent way. This allows the application programmer to concentrate all his efforts on functional development. Both systems, RODOS and DOMINION, support the distribution of services, but in a different way. RODOS uses a publishers/subscriber message passing system while DOMINION uses a shared memory paradigm. As the distribution mechanism is transparent to the applications, i.e. an application doesn't need to know how data is transported, the DOMINION-RODOS system is easily ported to the message passing system.

### 5.1. Demonstrator (Car Case Study)

A safety critical application with hard real time requirements, which is a standard practice in automotive engineering, should be chosen to demonstrate the combination of aerospace and automotive technologies as described in the previous chapters. An emergency brake assistance system represents a suitable application. Such an assistance system is proposed to avoid rear-end collisions in case of critical situations, or at least attenuate possible impacts.

Reliability of hard- and software components as well as plausibility of sensor information illustrate typical problems with such an assistance system. To deal with that problem the system is equipped with two independent front radar sensors that are connected to a set of three redundant Electronic Control Units (ECUs).

Handling transient computations and storage errors is one of the main problems in aerospace and is mainly caused by cosmic radiation. A possible way of dealing with this problem is the usage of triple modular redundancy (TMR). In a TMR-based system, three ECUs perform a calculation, while the result is checked by a voting instance. Only if two of the three ECUs deliver the same result, the result is treated as correct. In automotive engineering, different problems arise in the reliability of sensor information and defective hardware components such as ECUs. As an example, different ECUs process sensor inputs with different algorithms providing a wide variety of different sensor information combined with different processing algorithms.

DOMINION's automatic code-generation for both DOMINION and RODOS run-time environments allows parallel development and testing of the assistance system in DLR's driving simulators using the DOMINION run-time environment as well as in test vehicles using the RODOS run-time environment.

On the one hand, this allows an economic and fast functional processing already in very early stages of development (software-in-the-loop implementation and evaluation). On the other hand, hardware components can also be tested early and without the need of expensive integration into test vehicles. In addition, it is possible to test functionalities or system failures, especially in simulated critical situations without any "real-life" risks and with the benefit of exact reproducibility of the simulated situations.

Since DOMINION provides a run-time environment supporting all research facilities in the transportation system branch, it is easy to migrate from one facility to another during different development steps or changing requirements. Necessary steps for migration are supported by DOMINION. During the development process of advanced driver assistance systems, the actual site of operation is completely transparent from the developers or users point of view. For this reason, it is not necessary to gain expert knowledge regarding the different research facilities such as driving simulators or test vehicles. Thereby, the developer can completely concentrate on the implementation of their functions.

The components of the emergency brake assistance system that will be developed in this car case study, shall be, as shown, developed, parametrized, and evaluated in the simulators to be finally tested under real-life conditions in the test vehicle.

### **Detailed Description**

The figure 3 shows the usage of the combined DOMINION and RODOS environment in practical usage. Inputs for the DOMINION-code-generation are the architecture definition as well as the behavior description.



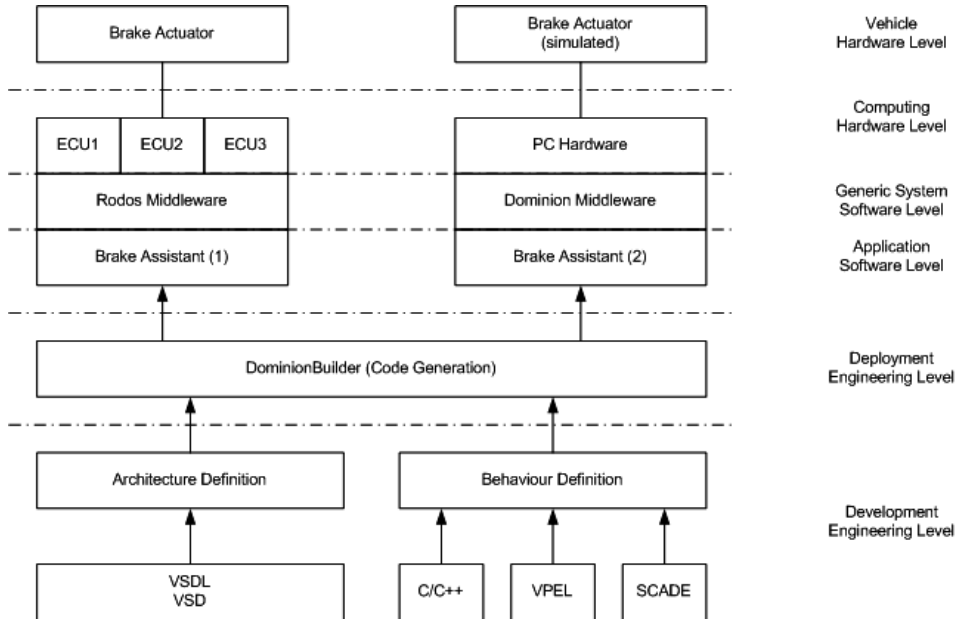


Figure 3: Combination of RODOS and DOMINION

The architecture definition describes modules and their input and output parameters, specified in VSDL. The code-generation can be switched to produce code for different target platforms. The application logic itself can then be described in programming languages like C/C++, VPEL, or SCADE. In case of critical functional development model-driven tools like SCADE support the development process by the application of certified code generators [1].

In the shown case, RODOS and DOMINION are based on C/C++, so code generation transforms all input specifications into valid C/C++ code. Practically, for the conversion of the code to another target platform only the specially generated base classes have to be exchanged.

The right part of the figure shows the classic DOMINION work-flow. The application is based on the DOMINION foundations and is deployed on a standard PC. To test the application, a simulation environment providing vehicle dynamics, street traffic, and a visualization is used. In this simulation environment, an easy and comparatively cheap evaluation of the newly added functions is possible.

After successful testing, the same application is generated for the RODOS platform (see left part of the figure). Code generation provides the necessary base integration classes, whereby the behavior description (i.e. the actual logic) of the considered application itself remains unchanged. No manual programming is necessary to adapt the application for the RODOS platform. To control the brake, three completely separated instances of the application are built to be run on three EDUs. The redundant setup then controls the brake. To minimize the influence of erroneous sensor signals, every application compares sensor inputs itself and performs plausibility checks. With that scenario, the most likely state of the environment is considered (sensor fusion). As pointed out, triple modular redundancy is used before the hardware brake is triggered. In this way the application can be tested in the FASCar-I as a real testing vehicle. The FASCar-I is equipped with several different sensors (e.g., radar, laser scanner, (D)GPS). Furthermore, steering and acceleration can be completely controlled by application logic on the on-board computers [2].

Because of the previous testing in the simulator the function is already in a mature stage and should be even work in critical situations as designed. However, since reality is more complex than any simulation, still further more testing and development is always necessary. Furthermore, development cycles can be performed, in which new advances are tested in the simulator again, before deploying them in the testing vehicle.

For the presented scenario the application logic for the brake controller application is diverse implemented three times by different developers. It is important to mention that only the application logic is implemented three times. As the application logic is only concerned with basic physical formulas for the braking conditions, it is short and reasonably well to understand. Therefore, errors in the application logic should be detectable by code reviews or similar precautions. Furthermore, the application logic is also checked with static software analysis for error recognition, which is the standard procedure for software in the space industry.

## 6. Summary

In this paper the merging of two software frameworks with a different background is discussed. The one is named RODOS and is mainly concerned with safety issues like redundancy and real-time communication with space industry background. The other platform, DOMINION, is developed within the context of automotive systems and allows to formally specifying services for later code generation and deployment on different target platforms. With that approach, we are basically able to combine the strength of both products. That allows us to use service descriptions and code generation for safety critical applications, dramatically accelerating software development in such scenarios. Until now, basic functionalities of RODOS and DOMINION have been successfully combined to show the potential of this approach. For further development, a complex scenario for a redundant, safety critical brake is planned, which proofs the value of the approach under realistic conditions.

Space sector and automotive employees have recognized that they can learn many things from each other and can even combine their work to reach more than just adding two parts. Based on the results of the planned scenario, further developments would be easily possible, i.e. a redundant steer-by-wire wheel controller. The authors believe that the described combination of the products offers great chances for future software development activities in the space and automotive sector.

## References

- [1] Schröder, Mark und Hannibal, Marco und Gacnik, Jan und Köster, Frank und Harms, Christian und Knostmann, Tobias (2010) *Ein Labor zur modellbasierten Gestaltung interaktiver Assistenz und Automation im Automotive-Umfeld*. AAET 2010 , 10.-11. Feb. 2010 , Braunschweig
- [2] Noyer, Ulf und Schomerus, Jan und Mosebach, Henning und Gacnik, Jan und Löper, Christian und Lemmer, Karsten (2008) *Generating High Precision Maps for Advanced Guidance Support*. In: IEEE Intelligent Vehicles Symposium 2008 , Seite 67. IEEE Intelligent Vehicles Symposium 2008 , 2008-06-04 - 2008-06-06 , Eindhoven (Niederlande).
- [3] Dr. Sergio Montenegro, Frank Dannemann *RODOS: Real Time Kernel Design for Dependability* , In: DATA Systems In Aerospace (DASIA) May 26 - 29, 2009, Istanbul, Turkey
- [4] Dr. Sergio Montenegro, John Richardson: *RODOS for Network Centric Core Avionics* In: Conference on Advances in Satellite and Space Communications July 20-25, 2009 - Colmar, France
- [5] Dr. Sergio Montenegro, Gunter Schoof, Ebrahim Haririan, In: *Network Centric Core Avionics*, DASIA 2009 DATA Systems In Aerospace, 26 to 29 May 2009, Istanbul

# Ansätze zum Nachweis der Gleichwertigkeit von Software-Komponenten

Hans-Werner Wiesbrock  
IT Power Consultants, 13355 Berlin, Germany  
hans-werner.wiesbrock@itpower.de

**Abstract:** Wie kann man heute ein bereits seit Jahren im Markt befindlichen System nach einer Änderung effizient rezertifizieren?

In diesem Artikel wird ein Lösungsansatz skizziert. Ausgehend von einer Risikoanalyse werden geeignete Vorgehensweisen und Kriterien abgeleitet, mit denen sich operational die Gleichwertigkeit von Software-Komponenten fassen lässt. Ein effizientes Vorgehen sähe dann wie folgt aus: Abgestimmt mit der Zertifizierungsbehörde werden für die zu ersetzenden Komponenten anhand eines Risikomodells die geeigneten Kriterien ihrer Gleichwertigkeit definiert. Der Hersteller weist anschließend nach, dass die geforderte Äquivalenz gilt.

## 1 Herausforderung

Mit dem Vordringen Software-intensiver, eingebetteter Systeme in sicherheitsrelevante Bereiche steigen die Anforderungen an Sicherheit und Zuverlässigkeit. Damit wachsen auch die Aufwände, um die qualitätsabsichernden Zulassungsverfahren zu passieren. Lange Produktzyklenzeiten und die Entwicklung neuer Technologien andererseits drängen auf partielle Ersetzungen oder Aufwertungen bestehender Systeme. Der Austausch einer Softwarekomponente kann erforderlich sein, wenn sich neue Programmierparadigmen und -architekturen bewähren und durchsetzen, die modellbasierte Vorgehensweise gegenüber dem herkömmlichen Codieren Einzug findet, verbesserte Kommunikationsstandards alte Standards ersetzen oder alte Compiler nicht mehr gewartet werden. Wie in vielen Bereichen können notwendige und erprobte Richtlinien und mit ihnen häufig verbundene Zertifizierungsprozesse hier auch zum Hemmschuh für sinnvolle Neuentwicklungen und nutzbringende Erweiterungen werden.

*Medizinprodukte sind auf Grund der hohen Anforderungen an Sicherheit und Zuverlässigkeit durch besonders lange Produktlebenszyklen und extrem aufwändige Zulassungsverfahren gekennzeichnet. Eine besondere Rolle hierbei nehmen softwarebasierte Systeme ein. Auf Grund der seit Jahren stetig zunehmenden normativen Anforderungen stellen sich zwei grundlegende Fragen:*

- *Wie kann man heute ein bereits seit Jahren im Markt befindliches System nach einer Änderung effizient rezertifizieren?*

- *Wie muss ein System heute entwickelt werden, um in der Zukunft bei Änderungen effizient rezertifiziert werden zu können?*

[SPE09]

Diese Fragestellungen haben enge Parallelen in der allgemeinen Systementwicklung. Wie kann man ein System erstellen, welches flexibel auf Änderungen reagieren kann? Die objektorientierte Analyse (OOA) antwortete hier mit Kapselung. Es sind geeignete Komponenten zu identifizieren, in der möglichst weitgehend die eigentliche Funktionalität verborgen ist [Bal96]. Entwurfsmuster unterstützen dabei den Entwickler [GHJV95, BMR<sup>+</sup>98]. Dann lassen sich einzelne, gekapselte Komponenten kontrollierbar austauschen. Allerdings sind im medizintechnischen Bereich neben einer objektorientierten Analyse zusätzlich die Sicherheitsmodelle zu berücksichtigen. Meist werden hierzu Fehlerbaumanalyse oder FMEA eingesetzt. In der deduktiven Fehlerbaumanalyse (FTA) geht man von einem Fehlverhalten aus und untersucht, wie es dazu kommen kann. Dazu komplementär ist die induktive Fehlerursache und -wirkungsanalyse, (FMEA), bei der aus möglichen Ausfällen im System deren Wirkung untersucht wird. Diese Sicherheitsanalysen sind auf die einzelnen Komponenten herunterzubrechen und wirken ihrerseits auf die gewählten Architekturen zurück [KLM03, GTS04].

Wie lässt sich sicherstellen, dass beim Ersetzen nicht unerwartete Nebeneffekte auftreten? Auch hier gibt es Standardmethoden: Back-To-Back-Tests oder Regressionstests, d.h., die neue Version wird gegen die alte Version evaluiert. Im Kontext sicherheitsrelevanter Alt- und Neusysteme ist hierbei jedoch offen, wie diese Tests aufzusetzen, welche Testdaten zu wählen sind und wie mögliche Abweichungen ihrer Ausgaben beurteilt werden sollen. Für eine systematische Antwort dieser Fragen bilden Sicherheitsmodelle die Grundlage, wie im Folgenden gezeigt werden soll.

Neben dynamischen Methoden gibt es auch statische Analysen, die für einen Nachweis der Gleichwertigkeit herangezogen werden können. So ist hier sicherlich das Model Checking zu nennen. Diese Arbeit beschränkt sich auf dynamische Methoden. Es werden verschiedene Kriterien hierfür vorgestellt. In einer späteren Untersuchung sollen weitere, insbesondere auch statische Ansätze verfolgt werden.

## **1.1 Voraussetzung: Komponentenbasierte Risikoanalyse**

Nach (DIN EN 62304) ist die Systementwicklung durch ein geeignetes Risikomanagement (DIN EN ISO 14971) abzusichern. Dazu gehört ein Verfahren zur Erkennung und Minimierung von Einflüssen und Risiken. Etablierte Verfahren hierzu sind Fehlerbaum-, Auswirkungs- und Hazard-Analysen. In ihren ursprünglichen Varianten wurden sie zur Untersuchung von Risiken ganzer Systeme entwickelt, mit Hardware im Fokus.

Für die obige Fragestellung eines zu rechtfertigenden Komponentenaustausches sind sie jedoch unzureichend, da wir sie auf einzelne Komponenten und ihre Risiken anwenden müssen. Betrachten wir dazu ein einfaches System, bestehend aus einer Sensor-, Software- und Aktuatorkomponente. Software altert nicht, Fehlverhalten ist reproduzierbar und ge-

schieht nicht zufällig über der Zeit. Andererseits treten bei der Verarbeitung systematische Rundungs- und Diskretisierungsabweichungen auf, die jedoch einem anderen statistischen Modell folgen. Die Sensoren ihrerseits haben quantifizierbare, beschränkte Wertediskriminierungen und die Aktuatoren physikalische Trägheiten, so dass es hier zu systematischen Messungenauigkeiten und Ansteuerungen stets kommt. Aus dieser Betrachtung folgt, dass eine induktive Wirkungsanalyse von Abweichungen für dieses komponentenbasierte System angemessener ist.

Kaiser et. al [KLM03] entwickelten einen komponentenbasierten Zugang für die Fehlerbaumanalyse. Giese et.al. [GTS04] schlagen einen eng mit der UML verzahnten Weg vor, Hazard-Analyse und Komponentenmodell zu vereinen. Ihre Grundidee ist, dass Fehler in den SW-Komponenten klassifiziert und darüber typisiert werden können. Die Fehlerauswirkungen pflanzen sich dann über getypte Ports in anderen Komponenten fort. Zu weiteren Ansätzen siehe [GKP05].

In dieser Arbeit soll nur ein sehr vereinfachtes, abstraktes Beispiel zur Illustration genutzt werden.

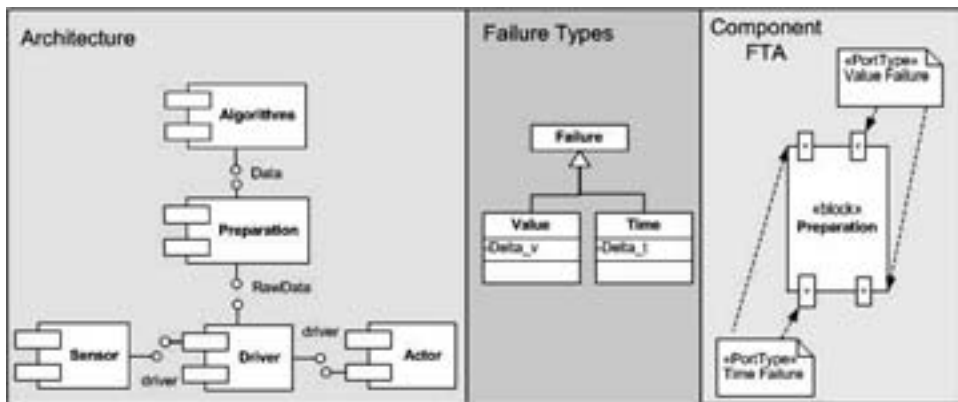


Abbildung 1: Architektur, Fehlertypen und Komponenten FTA

In Abbildung 1 wird ein generischer Aufbau eines eingebetteten Systems als Schichtenmodell mit Treiber/Vorverarbeitung/Algorithmen-Komponenten beschrieben, welches Sensordaten aufnimmt und einen Aktuator ansteuert. Dazu werden im Wesentlichen zwei Fehlertypen betrachtet: zum Einen fehlerhafte Werte, zum Anderen verfrühte oder verspätete Werte. In der dritten Grafik wird dann exemplarisch eine Ursache-Wirkungsanalyse für die Vorverarbeitungskomponente gezeigt, bei der fehlerhafte Werteeingaben und Zeitverzögerungen herein gehen und weitergegeben werden können.

Eine erschöpfende Modellierung ergibt eine detaillierte Sicht auf die möglichen Fehler und ihre Auswirkungen. Dieses Verfahren kann verfeinert werden, in dem man nach den Fehlerabweichungen  $\Delta_v$  und  $\Delta_t$  die Fehlertypen weiter differenziert, in unkritische, vertretbare und fehlerhafte Abweichungen. Dies wird nahegelegt, da Sensormessungen, Microcontroller-Berechnungen etc. technisch bedingt nur eine begrenzte Diskriminierung leisten, d.h. Rundungs- und Diskretisierungsabweichungen systematisch mit einzubezie-

hen sind. So kann erfasst werden, dass z.B. eine zeitliche Abweichung von 0.2 s unkritisch sein kann. Addiert sich diese jedoch mit weiteren späteren Verzögerung, verursacht durch eine Berechnung in einer anderen Komponente, so kann die Gesamtabweichung kritisch werden. Anhand einer solchen Verfeinerung des Sicherheitsmodells lassen sich dann Kennzahlen über die Werte- und Zeit-Toleranzen ableiten, die von den einzelnen Komponenten einzuhalten sind.

Als Ergebnis einer solchen Sicherheitsanalyse erhalten wir zweierlei. Zum Einen leitet sich aus der Einordnung des Systems in eine Sicherheitsklasse ab, welche Maßnahmen grundsätzlich zu ihrer Qualitätssicherung zu treffen sind.

**Beispiel:** Das System gehöre zur Medizin Produkt Klasse III. Dann sind Tests mit folgender Testtiefe durchzuführen:

- 100 % Anforderungsüberdeckung
- 100 % Zweig Überdeckung
- 95 % MCDC
- ...

Daraus erhalten wir, welche Menge von Testdaten mindestens zu beachten ist.

Zum Anderen erhalten wir Vergleichskriterien über erlaubte Abweichungen zwischen den auszutauschenden Komponenten:

**Beispiel:** Das Ausgabesignal der Komponente  $K_{neu}$ ,  $\sigma_1$ , darf höchstens um den Wert  $\Delta_y > 0$  zum Ausgabewert der Komponente  $K_{alt}$  abweichen mit einer maximalen zeitlichen Verzögerung  $\Delta_t > 0$ .

## 2 Back-To-Back Tests

Die Leitfrage dieses Artikels ist, unter welchen Umständen ein Austausch zweier Softwarekomponenten sicher durchgeführt werden kann. Dazu wird jetzt untersucht, wie die in der allgemeinen Softwareentwicklung etablierten Techniken von Back-To-Back- oder Regressionstests genutzt werden können. Die allgemeine Idee dieser Tests ist, eine Menge von gleichen Testeingaben auf die zwei Repräsentationen (Back-To-Back-) oder Versionen (Regressions-Tests) zu geben und die jeweiligen Ausgaben gegeneinander zu vergleichen. Damit stellen sich unmittelbar zwei Fragen: Welche Testdaten sollen betrachtet werden (Test Suite) und welche Vergleichskriterien sind heranzuziehen? Ausgehend von einer Risikoanalyse können, wie in 1.1 skizziert, diese Fragen systematisch beantwortet werden. So legt sie beispielsweise fest, ob der der Nachweis korrekter Umsetzung aller Anforderungen durch Tests genügt, ob gegebenenfalls eine bestimmte Testtiefe erreicht werden muss oder andere Aspekte zu verifizieren sind. Ferner zeigt sie, ob komplette Werteverläufe vorzugeben oder bestimmte Eigenschaften von Signalen und ihren Abhängigkeiten gefordert sind. Entsprechend sind dann Wertevergleiche mit Wert- und Zeittoleranzen für

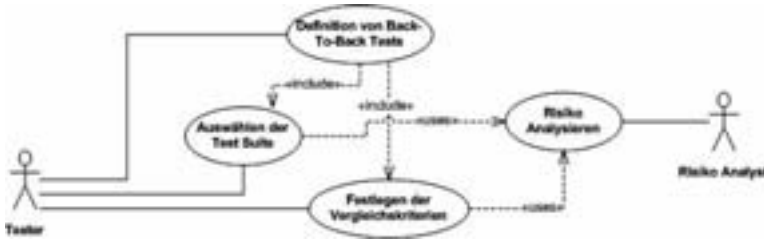


Abbildung 2: Use Cases für einen Back-To-Back Test

einen Vergleich heranzuziehen, siehe 2.2.1, oder Merkmalsbeschreibungen mit geeigneten temporallogischen Formeln, siehe 2.2.2.

## 2.1 Test Suiten

In diesem Kapitel werden verschiedene Test Suiten charakterisiert, die für einen Nachweis auf Gleichwertigkeit herangezogen werden können. Welche Test Suiten zu nehmen sind, aus welchen Kriterien heraus sie zu entwickeln sind, leitet sich im Allgemeinen aus der Risikoeinstufung des Produktes bzw. der Komponente ab. Nach der Medizinprodukte-richtlinie beispielsweise werden sie gemäß ihres Gefährdungspotenzials eingestuft. Daraus leitet sich dann auch ab, in welcher Testtiefe, und damit welcher Test Suite sie abzusichern sind.

### 2.1.1 Anforderungsbasiert

Nach DIN EN 62340 ist die korrekte Umsetzung von Systemanforderungen durch Tests verfolgbar nachzuweisen. Meist werden die Anforderungen auch auf die einzelnen Komponenten heruntergebrochen. Ausgehend von den Komponentenanforderungen lassen sich dann alle nachweisenden Tests über eine Tracinganalyse identifizieren. Diese bilden dann die anforderungsbasierte Test Suite. Zwei Komponenten sind dann *gleichwertig im Sinne der Anforderungen*, wenn beim Durchlaufen der anforderungsbasierten Test Suite ihre Ausgaben gemäß definiertem Vergleichskriterium übereinstimmen. Gleichwertigkeit in diesem Sinne ist sicherlich eine minimale Forderung.

### 2.1.2 Strukturbasiert

Weit eingesetzte Kriterien für die Testtiefe von Test Suiten sind Überdeckungsmaße. So werden insbesondere in sehr kritischen Bereichen  $x\%$  Zweigüberdeckung oder MCDC (Modified Condition/Decision Coverage) verlangt [Lig02]. Dazu muss eine Test Suite erstellt werden, die diese Überdeckung erreicht. Da eine Reimplementierung jedoch andere Strukturen aufweisen kann, ist für den Nachweis der Äquivalenz eine Test Suite zu erstellen, die diese Überdeckungstiefe bezüglich beider Komponenten erreicht. Zwei Kom-



ponenten sind dann *gleichwertig im Sinne der Strukturüberdeckung*, wenn beim Durchlaufen der strukturbasierten Test Suite ihre Ausgaben gemäß definiertem Vergleichskriterium übereinstimmen. Abhängig von der Sicherheitseinschätzung und den Programmiersprachen sind hier die geeigneten Strukturüberdeckungsmaße für den Äquivalenznachweis heranzuziehen, siehe die Diskussionen in [Lig02].

### 2.1.3 Robustheit

Meist werden die einzelnen Funktionen einer Komponente für sich genommen gut getestet und verglichen. Um nun auch deren Zusammenspiel abzusichern und damit die Robustheit auch gegen unwahrscheinliche Eingabekombinationen zu testen, kann man zufallsgesteuerte Tests durchführen. Da bei reinem, kombinatorischem Zufall jedoch zumeist nur abwegige, uninteressante Situationen gewürfelt werden, ist eine zufallsgesteuerte Testdatengenerierung über Markov-Automaten zu empfehlen, in denen typische Szenarien für die einzelnen Funktionen sowie auch unplausible Werte modelliert wurden. Zur Bestimmung der Testtiefe können kombinatorische Maße genutzt werden.

Die Ausgaben bei stochastischer Ansteuerung können in der Regel nicht mehr systematisch bestimmt werden. Auch können verschiedene Implementierungen hier zu erheblichen Abweichungen führen. Um sie dennoch vergleichen zu können, wird hier ein anderer Weg empfohlen. Ausgehend von den Anforderungen an die Komponente und der Risikoanalyse werden möglichst umfassend allgemeine Eigenschaften gefordert, die einzuhalten sind, siehe Kapitel 2.2.2. Wenn zwei Systeme sich bei stochastischer Ansteuerung und gegebenem Katalog geforderter Eigenschaften und Abhängigkeiten gleich verhalten, so heißen sie *gleichwertig im Sinne der Robustheit*. Dieses Verfahren sollte ergänzend zu den Anforderungs- und strukturbasierten Ansätzen eingesetzt werden, wenn die Sicherheitsanforderungen sehr hoch sind und zudem vielfältige Funktionalitäten interferieren können.

## 2.2 Vergleichskriterien

Wie in Kapitel 1.1 skizziert, kann aus einer verfeinerten, komponentenbasierten Risikoanalyse ein qualitatives Maß für die erlaubten Abweichungen abgeleitet werden. In diesem Kapitel sollen diese Kriterien etwas genauer untersucht werden. Zunächst wird dabei ein einfaches Kriterium vorgestellt, welches vielfach zur Absicherung der Code Generierung aus Modellen eingesetzt wird.

Zunächst einige Vorbemerkungen. Die Ausgänge eines eingebetteten Systems bilden Werte über der Zeit. Allgemein sollen hier Ausgänge als Zahlenwerte über einen Zeitbereich verstanden werden, im weiteren Signale  $\sigma$  genannt. Es sind verschiedene Zeitmodelle ( $\mathbb{Z}$  oder  $\mathbb{R}$  z.B.) denkbar. Wir vereinfachen diesen Ansatz und gehen von endlichen Zeitintervallen aus, sogar nur von  $[0,1]$ . Dies stellt keine wirkliche Einschränkung dar, siehe [MEv, CSW05]. Im Folgenden werden Kriterien für den Vergleich von Signalen entwickelt, die einen informellen Begriff der Gleichheit operationalisieren. Dabei werden zwei verschiedene Ansätze verfolgt. Im ersten werden die Signaldaten ohne weitere Semantik

mathematisch verglichen. Im zweiten Ansatz hingegen werden die Signalabhängigkeiten und erwarteten Eigenschaften formuliert, die dann die Basis für einen Vergleich geben.

### 2.2.1 Wertevergleiche

Als reell-wertige Funktionen über die Zeit lassen sich zwei Signale kanonisch über mathematische Normen vergleichen:

**Definition 2.1 (Absolute Differenz)**

Seien  $\sigma_1$  und  $\sigma_2$  Signale mit Werten in  $\mathbb{R}$ . Dann definieren wir:

$$\|\sigma_1 - \sigma_2\|_\infty := \sup_{t \in [0,1]} |\sigma_1(t) - \sigma_2(t)|$$

wobei  $|\cdot|$  der absolute Betrag auf  $\mathbb{R}$  ist.

Dieses Kriterium lässt sich einfach auf gewichtete Differenzen erweitern, d.h. anstelle des Betrages wird hier die Differenz noch durch geeignete Gewichtsfunktionen modifiziert.

Wenn also für zwei Signale  $\sigma_1, \sigma_2$  und vorgegebenem  $\epsilon > 0$  gilt:  $\|\sigma_1 - \sigma_2\|_\infty < \epsilon$ , so werden diese Signale als ähnlich eingestuft.

Dieses Kriterium jedoch ist ungeeignet, auch zeitliche Abweichungen zu berücksichtigen. Eine gängige Erweiterung besteht in der Betrachtung von sogenannten Schlauchumgebungen.

**Beispiel:** Ein Signal springt zunächst von 0 auf 1 und kurz darauf von 1 auf -1 und zurück auf 0. Das ist signifikant verschieden zu: von 0 auf -1, kurz darauf von -1 auf 1 und wieder auf 0, auch wenn diese Sprünge sehr dicht beieinander liegen. Legt man jedoch einen Schlauch um beide Signale, können sie leicht beide darin Platz finden. Ein Kriterium, welches die Ähnlichkeit zweier Signale durch solch eine Schlauchumgebung definiert, ist somit nicht geeignet, diese signifikanten Unterschiede zu detektieren, siehe 3.

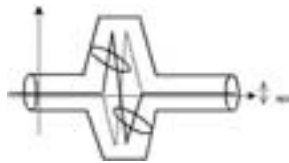


Abbildung 3: Kriterium: Schlauch  $\bigwedge t : \sigma_1(t)$  liegt im  $\epsilon$ -Schlauch von  $\sigma_2(t)$

Es wird deshalb ein anderes Kriterium hier vorgeschlagen. Zwei Ausgänge  $\sigma_1, \sigma_2$  heißen  $(\Delta_y, \Delta_t)$  gleich, wenn es (orientierungserhaltene) Zeitreparametrisierungen  $\gamma_1, \gamma_2 \in \text{Diff}_0([0, 1])$  gibt mit

$$\begin{aligned} \|\sigma_1 \circ \gamma_1(t) - \sigma_2(t)\| &< \Delta_y, \quad \forall t \in [0, 1] \\ \|\gamma_1(t) - t\| &< \Delta_t \quad \forall t \in [0, 1] \end{aligned}$$

und ebenso für  $1 \leftrightarrow 2$ .

Mathematisch präzisiert:

### Definition 2.2

Seien  $\sigma_1$  und  $\sigma_2$  Signale mit Werten in  $\mathbb{R}$ . Dann definieren wir für  $\Delta > 0$  :

$$d_{\Delta}^{\infty}(\sigma_1, \sigma_2) := \inf_{U_{\Delta}(id)} \|\sigma_1 \circ \gamma - \sigma_2\|_{\infty}$$

wobei  $\|\cdot\|_{\infty}$  die Differenz aus Definition 2.1 ist und  $U_{\Delta}(id) = \{\gamma \in \text{Diff}_0([0, 1]) \mid \|\gamma - id\|_{\infty} < \Delta\}$ .

Die Existenz und Eindeutigkeit einer solchen Größe lässt sich über mengentheoretische Eigenschaften von  $\text{Diff}_0([0, 1])$  und der vollständigen Ordnung der reellen Zahlen zeigen (Zornsches Lemma).

Zwei Signale  $\sigma_1, \sigma_2$  heißen  $(\Delta_y, \Delta_t)$ -ähnlich, g.d.w.  $d_{\Delta_t}^{\infty}(\sigma_1, \sigma_2) < \Delta_y$  gilt.

Für die diskretisierten Signale (Zeitmodell  $[0, 1] \cap \mathbb{Z} \cdot \delta$ ,  $\delta$  die Schrittweite) existieren Algorithmen und Implementierungen (Differenzmatrix-Verfahren, Dynamic Adaption), die einen automatischen Vergleich von Ausgängen nach diesen Kriterien unterstützen [MEv, CSW05]. Der Tester gibt für die relevanten Ausgangssignale die Toleranzen in den Werte- und Zeitabweichungen an. Das Werkzeug sucht dann eine geeignete Reparametrisierung und vergleicht anschließend die Signale nach Definition 2.2.

### 2.2.2 Merkmalsvergleiche

Ein anderer Zugang zu Vergleichskriterien geht über eine qualitative Beschreibung der Ausgänge. Dabei wird auch eine weitere Fragestellung aus dem Fallbeispiel aufgegriffen.

Wie kann man Anforderungen modellieren ohne die Lösung vorweg zu nehmen? [SPE09]

Diese Frage ist eng verbunden mit einer Beschreibung der zu erwartenden Ausgaben eines Systems. Zur Konstruktion von Eingangssignalen eines Systems gibt es zahlreiche Ansätze. Darin werden die Verläufe durch abschnittsweise Rampen, Splines etc. mit endlich vielen Stützstellen beschrieben. Viele erprobte Testsysteme setzen hier auf (CTE, MTest: [GK93, Con04]). Jedoch reichen diese Ansätze nicht allzu weit, will man die Verläufe der Ausgangssignale ebenso erfassen. Diese folgen nicht unbedingt elementaren Funktionsverläufen. Erwartete Nulldurchgänge, Signalanstiege, Extrema und andere markante Stellen sind relevant.

In [GW07, Loo08] werden Konzepte vorgestellt und Algorithmen skizziert, wie ein deskriptiver Ansatz realisiert werden kann. Im Gegensatz zum konstruktiven Ansatz, der häufig von einer vorgegebenen Zeiteinteilung ausgeht, folgt man hier dem erwarteten Werteverlauf. In einem ersten Schritt wird der erwartete qualitative Verlauf eines Signales beschrieben, also, z.B. eine Liste der Art: Erst Nulldurchgang, dann steiler Anstieg (Steigung

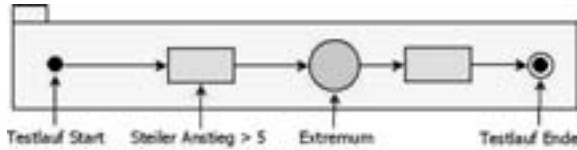


Abbildung 4: Verlaufdiagramm zur Beschreibung eines einfachen Signalverlaufes

>1), Maximum,.... Dazu wird eine grafische Notation vorgeschlagen, in der formalisiert ein solcher qualitativer Verlauf erfasst werden kann, siehe Abbildung 4.

Das Eintreten so spezifizierter Merkmale wie z.B. eine Flagänderung bildet ein Ereignis. Werden nun verschiedene Signale mit ihren Merkmalsbeschreibungen betrachtet, lassen sich über die in den Verlaufdiagrammen spezifizierten Ereignisse temporallogische Abhängigkeiten fordern. Insbesondere können in diesen Formeln Zeittoleranzen kodiert werden wie: Synchron bis auf  $\Delta_t$ -Abweichung. Auf diese Weise können geforderte Signalabhängigkeiten spezifiziert und in Testdurchläufen algorithmisch überprüft werden. Zwei Signalausgaben sind dann äquivalent, wenn sie definierten Verlaufdiagrammen und temporallogischer Abhängigkeiten genügen.

Dieser Ansatz kann dazu verwendet werden, Signalmuster zu definieren und Ausgänge auf allgemein verbindliche Abhängigkeiten zu überprüfen. Dies ist besonders für stochastische Ansteuerungen wichtig, bei denen die erwarteten Ausgangswerte nicht mit vertretbarem Aufwand vorweg bestimmt werden können. In einem Forschungsprojekt aus dem Automotive-Bereich wurde eine Variante dieses Vorgehens prototypisch umgesetzt und bereits sehr erfolgreich eingesetzt.

### 3 Vorgehen

Es wird eine Softwarekomponente reimplementiert, sei es, weil neue Hardware eingesetzt werden soll, der ursprüngliche Code nicht mehr wartbar war, kein gewarteter Compiler mehr existiert oder neue Programmierparadigma berücksichtigt werden sollten. Der Entwicklungsprozess, auch für die neue Komponente, ist bereits konform zu den gültigen Normen.

Im Folgenden gehen wir von einer erschöpfenden Risikoanalyse aus, die insbesondere auf Komponentenebene verfeinert wurde, siehe Abschnitt 1. Aus dieser Analyse lassen sich nun in Absprache mit der Zertifizierungsbehörde Kriterien festlegen, unter welchen Bedingungen die Komponenten als gleichwertig im Sinne eines kontrollierten Austausches gelten können. Dazu können statische Analysen ebenso gehören, wie dynamische Testtechniken, z.B. Back-To-Back-Tests. In Abbildung 5 wird exemplarisch und vereinfacht gezeigt, wie ein Vorgehen im Falle von Back-To-Back-Tests aussehen könnte.

Ausgehend von der komponentenbasierten Risikoanalyse werden mit der Zertifizierungsbehörde Kriterien abgestimmt, Test Suite und Vergleichskriterien, die dann vom Hersteller durchgeführt werden. Falls keine signifikanten Abweichungen auftreten, können die bei-

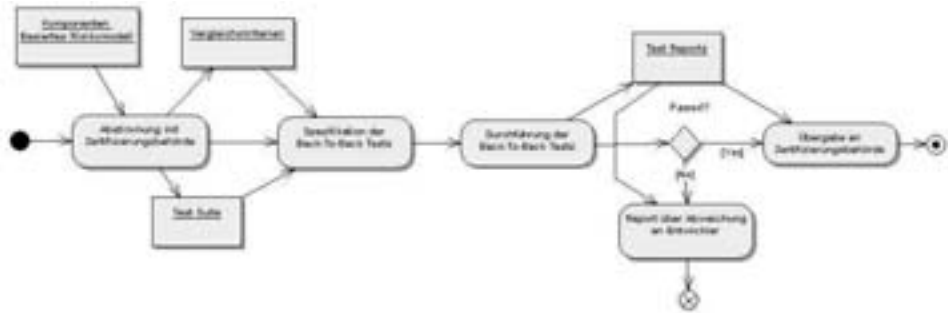


Abbildung 5: Vorgehen zur Rezertifizierung

den Komponenten als gleichwertig angesehen werden und ein Austausch ist abgesichert.

In dieser Arbeit konnte nur eine grobe Skizze gegeben werden. Viele Ideen, so insbesondere das Testen auf Robustheit über Markov-Automaten und ihre Auswertung über Merkmalsbeschreibungen, wie auch die Verwendung statischer Analyse-Kriterien werden zurzeit im Rahmen des Forschungsprojektes SPES 2020 weiter bearbeitet.

## Literatur

- [Bal96] H. Balzert. *Lehrbuch der Software-Technik, Bd. 1: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, 1996.
- [BMR<sup>+</sup>98] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad und M. Stal. *Pattern-orientierte Software-Architektur*. Addison-Wesley, Bonn, 1998.
- [Con04] M. Conrad. *Modell-basierter Test eingebetteter Software im Automobil*. Dissertation, TU-Berlin, 2004.
- [CSW05] M. Conrad, S. Sadeghipour und H.-W. Wiesbrock. Automatic Evaluation of ECU Software Tests. In *SAE World Congress*, Detroit (US), 2005.
- [GHJV95] E. Gamma, R. Helm, R. Johnson und J. Vlissides. *Design Patterns: Abstraction and Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [GK93] M. Grochtmann und K.Grimm. Classification Trees For Partition testing. In *Software testing, Verification & Reliability*, Jgg. 3 , Number 2, Seite 63–82. Wiley, 1993.
- [GKP05] L. Grunske, B. Kaiser und Y. Papadopoulos. Model-Driven Safety Evaluation with State-Event-Based Component Failure Annotations. In *CBSE*, Seiten 33–48, 2005.
- [GTS04] H. Giese, M. Tichy und D. Schilling. Compositional Hazard Analysis of UML Component and Deployment Models. In *SAFECOMP*, Seiten 166–179, 2004.
- [GW07] C. Gips und H.-W. Wiesbrock. Proposal for an Automatic Evaluation of ECU Output Signals. In *Dagstuhl-Workshop MBEES 2007*, Braunschweig, 2007.

- [KLM03] B. Kaiser, P. Liggesmeyer und O. Mäckel. A New Component Concept for Fault Trees. In *SCS*, Seiten 37–46, 2003.
- [Lig02] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Springer, 2002.
- [Loo08] M. Loose. *Entwicklung eines graphischen Editors und zugehöriger Algorithmen zur werkzeugunterstützten Überprüfung von Signalabhängigkeiten*. Diplomarbeit, HU Berlin, 2008.
- [MEv] MEval. [www.itpower.de/12-0-Tool-Produkte.html](http://www.itpower.de/12-0-Tool-Produkte.html).
- [SPE09] SPES 2020, Teilprojekt Medizintechnik, Berlin Heart. Beschreibung der Fallstudie im Anwendungsgebiet Medizintechnik, November 2009.



# Evaluation eines modellbasierten Requirements-Engineering-Ansatzes für den Einsatz in der Motorsteuerungs-Domäne<sup>1</sup>

Ernst Sikora, Klaus Pohl

Universität Duisburg Essen  
Schützenbahn 70, 45127 Essen  
{ernst.sikora; klaus.pohl}@sse.uni-due.de

**Abstract:** Modellbasierte Entwicklungsansätze für softwareintensive, eingebettete Systeme sollen dazu beitragen, Fehler im Entwicklungsprozess zu vermeiden, die Entwicklungsdauer zu verkürzen und die hohe Systemkomplexität besser beherrschbar zu machen. Besonders vielversprechend erscheint der Einsatz von Modellen bereits in frühen Entwicklungsaktivitäten, in denen die Anforderungen und die Architektur definiert werden. COSMOD-RE ist ein ziel- und szenariobasierter Requirements-Engineering-Ansatz, der eine verzahnte Entwicklung von Anforderungs- und Architekturmodellen über mehrere Abstraktionsstufen eines softwareintensiven, eingebetteten Systems hinweg unterstützt. In diesem Beitrag werden die Erkenntnisse und Erfahrungen beschrieben, die bei der Evaluation von COSMOD-RE in einem spezifischen Entwicklungskontext, der Entwicklung von Anforderungen für die Anwendungssoftware einer Motorsteuerung, gesammelt wurden. Hauptmotivation für diese Evaluation war es, Erkenntnisse über den Nutzen und die Anwendbarkeit von COSMOD-RE in dieser Domäne zu gewinnen.

## 1 Motivation und Problemstellung

Bei der Entwicklung komplexer, softwareintensiver, eingebetteter Systeme müssen Anforderungen auf verschiedenen Abstraktionsstufen definiert und analysiert werden, beispielsweise Anforderungen an das Gesamtsystem sowie Anforderungen an die Systemsoftware. Modellbasierte Requirements-Engineering-Ansätze sollen dazu beitragen, Fehler in Anforderungen an das System bzw. die Software zu vermeiden, sowie in kürzerer Zeit zu einem umfassenden Verständnis der Anforderungen zu gelangen. Der COSMOD-RE-Ansatz (siehe [PS09] und [Po10]) unterstützt die modellbasierte Entwicklung von Anforderungen an softwareintensive, eingebettete Systeme. COSMOD-RE verknüpft dazu verschiedene, im RE etablierte Modellierungstechniken wie Ziel-, Szenario- und Funktionsmodellierung.

---

<sup>1</sup> Dieser Beitrag wurde gefördert durch die BMBF Innovationsallianz SPES 2020, Förderkennzeichen 01IS08045.



Um Erfahrungen für eine Weiterentwicklung des Ansatzes zu sammeln, wird COSMOD-RE in verschiedenen Teilgebieten der Domäne softwareintensiver, eingebetteter Systeme evaluiert. Dieser Beitrag beschreibt die Evaluation von COSMOD-RE im Kontext der Entwicklung von Anwendungssoftware für Motorsteuerungen im Automobilbereich (siehe [BDJ03]). Die Evaluation von COSMOD-RE in der Motorsteuerungs-Domäne stellt einen kritischen Anwendungsfall für den Ansatz dar, da Ziele und Szenarien im RE typischer Weise zur Beschreibung der Systemnutzung durch Personen und andere Systeme eingesetzt. Die Motorsteuerung hat jedoch (zum Beispiel im Vergleich zu einem Fahrerassistenzsystem) wenig direkte Nutzerinteraktion und wird weitgehend durch regelungstechnische Abläufe bestimmt. Der Beitrag ist wie folgt aufgebaut: Abschnitt 2 gibt einen Überblick über den COSMOD-RE-Ansatz. Abschnitt 3 berichtet über die durchgeführte Evaluation des Ansatzes in der Motorsteuerungs-Domäne. Abschnitt 4 zieht Schlussfolgerungen aus den Evaluationsergebnissen.

## 2 Überblick über den COSMOD-RE-Ansatz

COSMOD-RE [PS09] [Po10] ist ein ziel- und szenariobasierter RE-Ansatz für softwareintensive, eingebettete Systeme, der im Rahmen mehrerer Forschungs-Industrie-Kooperationen entwickelt und erprobt wurde. COSMOD-RE weist aufgrund der folgenden Eigenschaften eine prinzipielle Eignung für die Motorsteuerungs-Domäne auf:

- ❑ In der Motorsteuerungs-Domäne müssen Anforderungen auf verschiedenen Abstraktionsebenen betrachtet werden, wie zum Beispiel auf der Fahrzeugebene, der Systemebene und der Software/Hardware-Ebene. COSMOD-RE unterstützt die Spezifikation von Anforderungen auf verschiedenen Abstraktionsebenen mittels einer Abstraktionsstufenhierarchie sowie mittels Prozessstrukturen für die Entwicklung von Anforderungen über mehrere Abstraktionsebenen hinweg.
- ❑ In der Motorsteuerungs-Domäne ist eine funktionsorientierte Spezifikation von System- und Softwareanforderungen weit verbreitet. Von einigen Autoren wird darüber hinaus der Einsatz von Zielen und Szenarien angeregt [PK04]. COSMOD-RE unterstützt die modellbasierte Spezifikation von Anforderungen mittels Ziel-, Szenario- und Funktionsmodellen. Ziel- und Szenariomodelle unterstützen dabei die systematische Identifikation von System- und Softwarefunktionen.

In den folgenden Unterabschnitten wird ein kurzer Überblick über die wesentlichen Bestandteile des COSMOD-RE-Ansatzes gegeben. Eine umfassende Beschreibung von COSMOD-RE ist in [Po10] zu finden.

### 2.1 COSMOD-RE Abstraktionsstufenhierarchie

Eine wesentliche Eigenschaft komplexer Systeme ist deren Strukturierung in mehrere Abstraktionsebenen. COSMOD-RE unterstützt die Entwicklung von Anforderungen auf verschiedenen Abstraktionsebenen unter anderem durch eine vorgegebene, generische Abstraktionsstufenhierarchie. Diese Abstraktionsstufenhierarchie soll Requirements-Ingenieure dabei unterstützen, den richtigen Abstraktions- und Detaillierungsgrad für die Anforderungsartefakte auf einer bestimmten Abstraktionsebene zu wählen. Die

Abstraktionsstufenhierarchie führt zudem (auf höheren Abstraktionsstufen) zur Komplexitätsreduktion der erstellten Modelle, da sie die Abstraktion von Lösungsdetails fördert. Die Abstraktionsstufenhierarchie von COSMOD-RE ist in Abb. 1 dargestellt.

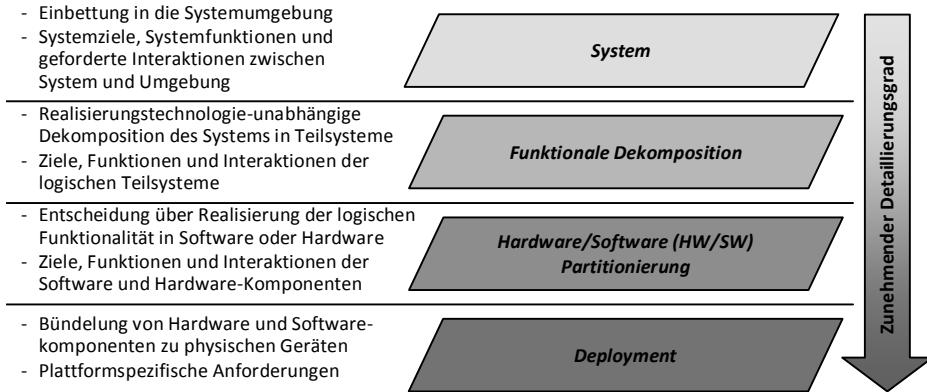


Abb. 1 Abstraktionsstufenhierarchie der COSMOD-RE-Methode

## 2.2 COSMOD-RE Artefakttypen

COSMOD-RE beinhaltet lösungsneutrale Anforderungsmodelle (Zielmodell und Szenariomodell), lösungsorientierte Anforderungsmodelle (Datenmodell, Funktionsmodell und Verhaltensmodell) sowie strukturelle Architekturmodelle. Tab. 1 zeigt einen Überblick über die wesentlichen von COSMOD-RE unterstützten Artefakttypen, deren Eigenschaften sowie mögliche Modellierungssprachen.

Artefakttyp	Kurzbeschreibung
<b>Zielmodell</b>	Definiert die Ziele, die das System erfüllen soll, einschließlich der hierarchischen Dekomposition der Ziele. Die Ziele werden unabhängig von einer möglichen oder intendierten Systemlösung definiert.
<b>Szenario-modell</b>	Definiert eine Menge von Szenarien und setzt diese zur Systemumgebung in Beziehung. Ein Szenario beschreibt eine Folge von Interaktionen, die zur Erfüllung eines bestimmten Ziels führen soll.
<b>Funktions-modell</b>	Definiert die Funktionen (Dienste), die das geplante System seiner Umgebung bereitstellen soll, die Eingabe- und Ausgabedatenflüsse der einzelnen Funktionen sowie wesentliche Datenspeicher des Systems.
<b>Datenmodell</b>	Definiert welche Informationen über die Umgebung im System abgebildet werden sollen und wie diese strukturiert sind.
<b>Verhaltens-modell</b>	Definiert die Betriebsmodi des Systems, Ereignisse die zum Wechsel des Betriebsmodus führen und, welche Funktionen in welchem Modus ausgeführt werden.
<b>Architektur-modell</b>	Definiert wesentliche Strukturen des geplanten Systems wie Schnittstellen zu anderen Systemen, die Dekomposition des Systems sowie Kommunikationsverbindungen zwischen den Systemkomponenten.

Tab. 1 Grobe Charakterisierung der in COSMOD-RE verwendeten Artefakttypen

Jedes in COSMOD-RE eingesetzte Anforderungs- oder Architekturmodell hat zwei wesentliche Bestandteile. Erstens beinhaltet ein Modell Diagramme, die die wesentlichen Modellelemente und deren Beziehungen grafisch dokumentieren. Zweitens beinhaltet jedes Modell textuelle Detailbeschreibungen der einzelnen Modellelemente. Damit soll sichergestellt werden, dass die Anforderungs- und Architekturartefakte möglichst verständlich und eindeutig interpretierbar sind. Die textuellen Detailbeschreibungen von Modellelementen werden mittels Schablonen strukturiert, die für den jeweiligen Artefakttyp die zu dokumentierenden Informationen festlegen.

Die in Tab. 1 aufgeführten Artefakttypen können auf jeder Abstraktionsebene eines Systems eingesetzt werden. Dies ist dadurch begründet, dass auf jeder Abstraktionsebene prinzipiell ähnliche Entwicklungsentscheidungen getroffen und die Übergänge zwischen verschiedenen Abstraktionsebenen möglichst frei von Brüchen sein sollten. Von der höchsten zur niedrigsten Abstraktionsebene hin steigt jedoch der Detaillierungsgrad der in den Modellen dokumentierten Anforderungen und Architekturinformationen.

### 2.3 COSMOD-RE Co-Design-Prozess

COSMOD-RE unterstützt durch so genannte „Co-Design“-Prozesse die verzahnte Entwicklung von Anforderungen und Architektur über mehrere Abstraktionsebenen hinweg. Abb. 2 zeigt die Struktur eines Co-Design-Prozesses in Form von Subprozessen und Informationsflüssen zwischen den Subprozessen. Ein Co-Design-Prozess verfolgt das Ziel, zunächst (lösungsneutrale) Anforderungen zu identifizieren und einen ersten, groben Architekturvorschlag zu erarbeiten. Erst nachdem die Problemstellung in Form von lösungsneutralen Anforderungsmodellen (Zielen und Szenarien) herausgearbeitet, und ein initialer Architekturvorschlag identifiziert wurden, erfolgt die Spezifikation detaillierter, lösungsorientierter Anforderungen.

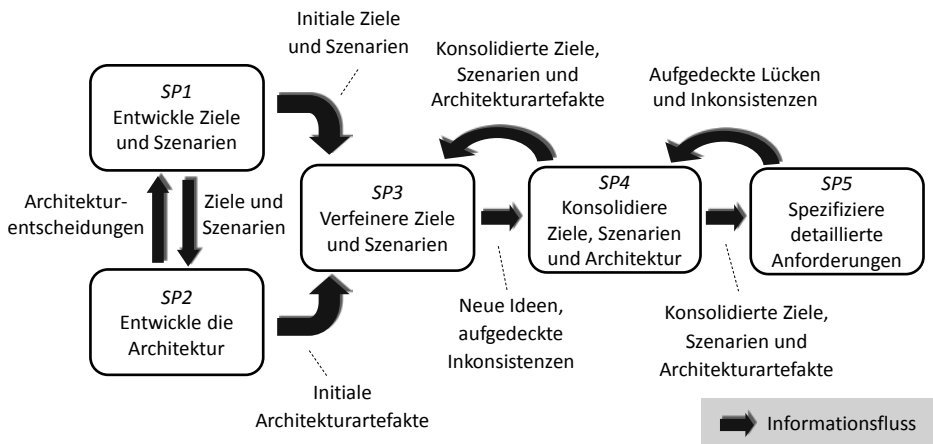


Abb. 2 Struktur eines COSMOD-RE Co-Design-Prozesses (siehe [PS09])

Die in Abb. 2 gezeigten Subprozesse können wie folgt charakterisiert werden:

- Subprozess  $SP_1$  ist für die Entwicklung initialer Ziele und Szenarien für das geplante System verantwortlich.
- Subprozess  $SP_2$  ist für die Entwicklung einer initialen, groben Systemarchitektur verantwortlich.
- Subprozess  $SP_3$  ist für die Definition verfeinerter Ziele und Szenarien und deren Zuordnung zu Architekturelemente verantwortlich.
- Subprozess  $SP_4$  ist für die Auflösung von Unstimmigkeiten zwischen Zielen, Szenarien und Architektur zuständig.
- Subprozess  $SP_5$  ist für die Spezifikation detaillierter, lösungsorientierter Anforderungen basierend auf den definierten und abgestimmten Zielen, Szenarien und Architekturartefakten zuständig.

### 3 Evaluation von COSMOD-RE in der Motorsteuerungs-Domäne

Die nachfolgend beschriebene Evaluation von COSMOD-RE in der Motorsteuerungs-Domäne umfasst die folgenden, wesentlichen Evaluationsschritte:

- Definition von Evaluationskriterien durch Domänenexperten
- Dokumentbasierte Evaluation
- Geführte Anwendung von COSMOD-RE
- Evaluation des wahrgenommenen Nutzens und der Verständlichkeit des Ansatzes mittels eines Fragebogens

Im Folgenden werden zunächst der Entwicklungskontext und anschließend die einzelnen Evaluationsschritte und deren Ergebnisse beschrieben.

#### 3.1 Entwicklungskontext in der Motorsteuerungs-Domäne

Die grundlegende Funktion einer Motorsteuerung ist die Umsetzung von Drehmomentanforderungen, die beispielsweise aus dem Beschleunigungswunsch des Fahrers resultieren. Dazu steuert die Motorsteuerung in geeigneter Weise den Verbrennungsprozess im Motor. Wesentliche Steuergrößen sind dabei Luftzufuhr, Kraftstoffzufuhr und Zündzeitpunkt (siehe Abb. 3). Eine umfassende Darstellung der Funktionen und der Systemstruktur einer Motorsteuerung ist z.B. in [BDJ03] zu finden.

Der Ausgangspunkt für die Entwicklung einer Motorsteuerung ist das Lastenheft des Fahrzeugherstellers. Basierend auf der Analyse der Anforderungen des Fahrzeugherstellers wird das Systemkonzept (meist eine existierende Plattform) festgelegt, auf deren Basis die Motorsteuerung realisiert wird. Bei der Funktionsentwicklung werden die Funktionen der Motorsteuerung ausgehend von den Anforderungen des Fahrzeugherstellers und des gewählten Systemkonzepts im Detail spezifiziert und in einem Pflichtenheft dokumentiert. Bei der Softwareentwicklung werden die spezifizierten Funktionen in Programmcode überführt. Die Software wird in

übergeordnete Steuerungsfunktionen (Anwendungssoftware) und Basisfunktionen unterteilt, welche der Ansteuerung von Hardwarekomponenten dienen.

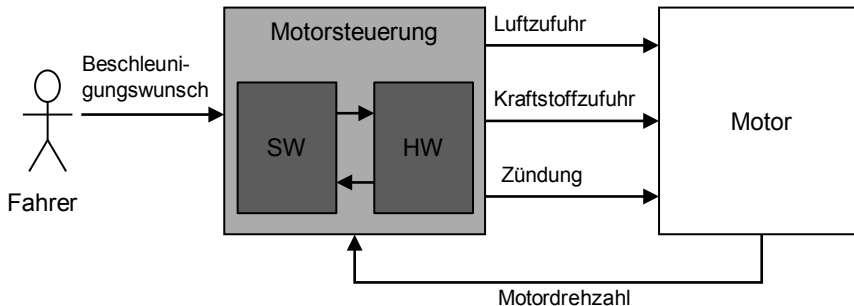


Abb. 3 Stark vereinfachtes, partielles Kontextmodell einer Motorsteuerung

### 3.2 Evaluationskriterien

Fokus der Evaluation von COSMOD-RE war die Durchführung des RE für die Anwendungssoftware (d.h. die Steuerungsfunktionen) der Motorsteuerung. Ziel war es somit, die Eignung von COSMOD-RE für diesen Teilbereich des Entwicklungsprozesses einer Motorsteuerung zu untersuchen. Zur Unterstützung der Evaluation des Ansatzes legten die Domänenexperten unter anderem die folgenden Evaluationskriterien fest:

1. Der Ansatz sollte die Spezifikation von Anforderungen auf zwei Abstraktionsebenen unterstützen (Produktebene und Systemebene).
2. Der Ansatz sollte den Übergang zwischen den Abstraktionsebenen unterstützen.
3. Der Ansatz sollte die Modellierung von zwei verschiedenen Problemklassen adressieren: diskrete und kontinuierliche Problemklassen.
4. Der Ansatz sollte die Modellierung verschiedener Technologien unterstützen: Software, digitale Hardware, Regelungstechnik und Mechanik.
5. Der Ansatz sollte einen durchgängigen RE-Prozess unterstützen.
6. Der Ansatz sollte einfach erlernbar sein.

### 3.3 Dokumentbasierte Evaluation

Die dokumentbasierte Evaluation von COMSOD-RE wurde von Domänenexperten basierend auf einer umfassenden Beschreibung des Ansatzes durchgeführt und erfolgte gegen die zuvor festgelegten Evaluationskriterien. Die Beschreibung des Ansatzes umfasste Definitionen der Grundprinzipien des Ansatzes, der zu erstellenden Entwicklungsartefakte, eine grobe Prozessdefinition sowie ein Beispiel, das den Ansatz veranschaulicht. Tab. 2 zeigt einen Überblick über die Ergebnisse der dokumentbasierten Evaluation des Ansatzes. Diese Ergebnisse spiegeln die subjektiven Eindrücke wieder, die die Domänenexperten aufgrund der dokumentbasierten Beschreibung des Ansatzes gewonnen haben.

Nr.	Kriterium	Erfüllungsgrad
1	Spezifikation von Produkt- und Systemanforderungen	erfüllt
2	Übergang von Produkt- zu Systemebene	teilweise erfüllt
3	Modellierung diskreter und kontinuierlicher Probleme	teilweise erfüllt (diskrete Probleme)
5	Modellierung der relevanten Technologien	teilweise erfüllt (Software)
6	Durchgängiger RE-Prozess	erfüllt
7	Einfache Erlernbarkeit	erfüllt

Tab. 2 Überblick über wesentliche Ergebnisse der dokumentbasierten Evaluation

### 3.4 Geführte Anwendung von COSMOD-RE

Da die dokumentbasierte Evaluation nur begrenzte Einblicke in Eigenschaften wie zum Beispiel die Durchgängigkeit des Ansatzes erlaubt, wurde zusätzlich ein gemeinsamer Workshop von Methoden- und Domänenexperten durchgeführt, bei dem der Ansatz exemplarisch zur Entwicklung von Anforderungen an die Anwendungssoftware einer Motorsteuerung eingesetzt wurde. Die Methodenexperten stellten dabei den Ansatz vor und unterstützten die Domänenexperten bei der Anwendung des Ansatzes. Im Folgenden werden die während der geführten Anwendung gewonnenen Erkenntnisse dargestellt. Die Erkenntnisse sind anhand der Bestandteile des COSMOD-RE-Ansatzes strukturiert.

#### Abstraktionsstufen

Die Unterstützung von Abstraktionsebenen war eine zentrale Anforderung an den Ansatz. Bei der geführten Anwendung des Ansatzes zeigte sich, dass während der Gewinnung von Anforderungen ein flexibler Umgang mit dem Abstraktionsstufenmodell von COSMOD-RE erforderlich ist:

- Der Ansatz muss auch dann anwendbar sein, wenn der Startpunkt des RE-Prozesses auf einer niedrigen Abstraktionsstufe liegt. In dem Beispiel wurde die Anwendungssoftware der Motorsteuerung als Gegenstand der Entwicklung gewählt. Somit lag der Startpunkt auf der Abstraktionsstufe „Hardware-Software-Partitionierung“. Die für die höheren Abstraktionsstufen relevanten Informationen wurden erst nach und nach ausgehend von diesem Startpunkt identifiziert. Dabei zeigte sich die Notwendigkeit, klare, domänenspezifische Kriterien vorzugeben, um die gewonnenen Informationen über das System jeweils in die richtige Abstraktionsstufe einzuordnen.
- Der Ansatz muss berücksichtigen, dass im Entwicklungsprozess binnen kurzer Zeit Informationen für mehrere verschiedene Abstraktionsstufen anfallen. Zum Beispiel kann es vorkommen, dass Ziele und Szenarien auf der höchsten Abstraktionsstufe und unmittelbar darauf lösungsorientierte Anforderungen auf einer sehr niedrigen Abstraktionsstufe definiert werden. Dies kann unter anderem damit begründet werden, dass Systemingenieure über ein umfassendes Wissen und Erfahrungen in Bezug auf die jeweilige Domäne verfügen und daher beispielsweise die Auswirkungen von Anforderungen einer höheren Abstraktionsstufe auf niedrigere Abstraktionsstufen unmittelbar, ohne mehrere Zwischenschritte bewerten können.

## Anforderungsartefakte

In Bezug auf die von COSMOD-RE unterstützten Artefakttypen wurden die folgenden Erkenntnisse gewonnen.

- ❑ Die Identifikation von Stakeholdern und die explizite Definition der Ziele der Stakeholder haben sich als sehr wichtig und hilfreich herausgestellt, um eine Ausgangsbasis für die Identifikation weiterer, detaillierter Anforderungen zu erhalten. Unter anderem wurden dabei wesentliche Entwicklungsziele der Motorsteuerungs-Domäne wie zum Beispiel das Ziel „Ressourcen-schonendes Fahren“ frühzeitig aufgedeckt und dokumentiert.
- ❑ Für eine erfolgreiche Anwendung der ziel- und szenariobasierten Vorgehensweise war es wesentlich, die Szenarien mit dem richtigen Abstraktionsgrad zu definieren. Dabei wurde von den Domänenexperten ein Abstraktionsgrad als hilfreich angesehen, bei dem die aus der Perspektive eines Systemingenieurs essenziellen Schritte zur Erfüllung des jeweiligen Ziels dokumentiert werden.
- ❑ Das Funktionsmodell, das Verhaltensmodell und das Architekturmodell erwiesen sich als sehr hilfreich, um die gewonnenen Informationen zu dokumentieren. Bei der Entwicklung dieser Modelle erwies sich ein Vorgehen sinnvoll, bei dem für jeweils ein spezifisches Ziel mehrere Szenarien definiert und dann für jedes einzelne Szenario die Architektur erweitert wird und die benötigten Softwarefunktionen und die benötigten Modi in den entsprechenden Modellen definiert werden.

## Co-Design-Prozess

Mit den von COSMOD-RE definierten Co-Design-Prozessen wurden die folgenden Erfahrungen gemacht:

- ❑ Prinzipiell wurden alle Subprozesse als sinnvoll erachtet und die Übergänge zwischen den Schritten als hinreichend durchgängig angesehen.
- ❑ Der Subprozess  $SP_2$  (Entwicklung einer initialen Architektur) wurde aufgrund des Startpunkts auf einer niedrigen Abstraktionsstufe (der Motormanagement-Anwendungssoftware) in einer modifizierten Form durchgeführt. Die zentrale Frage bei diesem Schritt war, wie die Einbettung der Software in das System gestaltet werden muss, um die identifizierten Szenarien erfüllen zu können. Dabei wurden beispielsweise notwendige Schnittstellen zu Sensoren, Aktuatoren sowie anderen Systemen im Fahrzeug identifiziert.
- ❑ Die identifizierte (System-)Architektur und die Szenarien wurden als gute Ausgangsbasis angesehen, um Softwarefunktionen zu identifizieren. Dabei zeigte sich die Notwendigkeit, die ausgehend von mehreren Szenarien identifizierten Softwarefunktionen in einem Folgeschritt zu konsolidieren (zum Beispiel durch Generalisierung oder Aggregation).

## 3.5 Evaluation mittels Fragebögen

Im Anschluss an die dokumentbasierte Evaluation und die geführte Anwendung von COSMOD-RE wurden mittels eines Fragebogens der wahrgenommene Nutzen und die Verständlichkeit des Ansatzes evaluiert. Der Fragebogen enthielt neben Fragen zum Hintergrund des Befragten jeweils Aussagen zum wahrgenommenen Nutzen und zur

Verständlichkeit des Ansatzes, die von den Domänenexperten auf einer 7-Punkt-Skala beantwortet wurden. Die Aussagen zum Nutzen wurden aus Hypothesen über die Vorteile des Ansatzes für den RE-Prozess abgeleitet. Zum Beispiel lautete eine Aussage „Durch die Verwendung des Ansatzes werden Fehler bei der Verfeinerung von Anforderungen vermieden“. Die Bewertung der Verständlichkeit des Ansatzes erfolgte mittels Aussagen wie zum Beispiel „Es wäre für mich einfach, den Ansatz anderen Mitarbeitern in meiner Abteilung zu erläutern.“

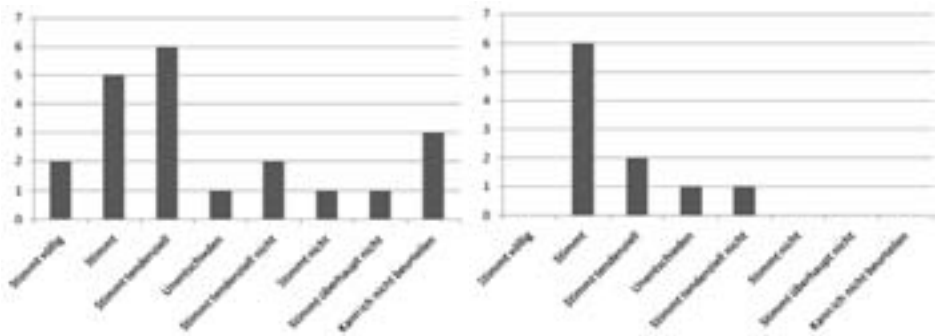


Abb. 4 Aggregierte Antworthäufigkeiten (*links*: Fragen zum wahrgenommenen Nutzen; *rechts*: Fragen zur Verständlichkeit und Erlernbarkeit)

Die aggregierten Häufigkeiten der Antworten auf Fragen zum wahrgenommenen Nutzen und zur Verständlichkeit sind in Abb. 4 dargestellt. Die Antworthäufigkeiten zeigen eine überwiegend positive Beurteilung des Ansatzes, weisen in einigen Detailfragen jedoch auch auf Verbesserungsmöglichkeiten hin. Beispielsweise wurde von den Befragten angeregt, eine Unterstützung für den Architekturentwurf basierend auf Zielen und Szenarien in den Ansatz zu integrieren. Zudem wurde der Bedarf nach einer Unterstützung für die automatisierte Prüfung von Qualitätseigenschaften wie Konsistenz und Vollständigkeit der Anforderungsmodelle festgestellt.

## 4 Schlussfolgerungen

Die eingesetzte Vorgehensweise zur Evaluation bestehend aus einer dokumentbasierten Evaluation, einer geführten Anwendung des Ansatzes durch Domänenexperten und einer schriftlichen Befragung hat sich bewährt, da jeder Schritt zu einem signifikanten Erkenntnisgewinn über den Nutzen und die Anwendbarkeit des Ansatzes für die betrachtete Domäne erbrachte. Die in Abschnitt 3 beschriebenen Evaluationsergebnisse legen folgende, zentrale Schlussfolgerungen nahe:

- Die Kombination aus Zielmodellierung, Szenariomodellierung, Funktionsmodellierung und Architekturmodellierung bildet eine gute Basis für die Realisierung eines durchgängigen RE-Prozesses in der anvisierten Domäne.
- Die explizite Unterstützung mehrerer Abstraktionsebenen wirkt sich in der betrachteten Domäne sehr positiv auf die Akzeptanz eines Ansatzes aus. Allerdings



können während der Entwicklung der Anforderungsmodelle Unklarheiten über die Einordnung der im Entwicklungsprozess anfallenden Informationen in die verschiedenen Abstraktionsebenen auftreten. Vor der Anwendung von COMSOD-RE empfiehlt es sich daher, mit den Domänenexperten einen Satz von Richtlinien zum Umgang mit Abstraktionsebenen während des RE-Prozesses zu definieren.

- Regelungstechnische Problemstellungen können durch Modellierungssprachen wie UML [OMG09] und SysML [OMG09] nur unzureichend abgebildet werden. Diese Problemstellungen sind für die Definition von System- und Softwareanforderungen in der betrachteten Domäne jedoch von zentraler Bedeutung. Zukünftige Versionen von COSMOD-RE sollten daher eine erweiterte Unterstützung für die Einbeziehung von regelungstechnischen Problemklassen in den RE-Prozess anbieten wie zum Beispiel Modellierungstechniken für hybride (diskret-kontinuierliche) Systeme.

Die weiteren Arbeiten an COSMOD-RE umfassen die systematische Evaluation des Ansatzes in unterschiedlichen Domänen, unter anderem, Avionik, Energietechnik und Automatisierungstechnik. Zudem wird der Ansatz basierend auf den gewonnenen Erkenntnissen und Erfahrungen überarbeitet und verbessert.

### **Danksagung**

Für die Mitwirkung an den beschriebenen Arbeiten danken wir Ulrich Freund, Franz Grzeschniok, Christine Haas, Peter Heidl, Igor Menzel, Mark Müller und Christoph Störmer. Wir danken Markus Ciolkowski für die kritische Begutachtung unseres Evaluationsfragebogens.

### **Literatur**

- [BDJ03] Bauer, H.; Dietsche, K., Jäger, T.: Gasoline-Engine Management – Motronic Systems. Yellow Jackets, Robert Bosch GmbH, 2003.
- [Po10] Pohl, K.: Requirements Engineering – Foundations, Principles, Techniques. Springer, erscheint im August 2010.
- [PK04] Puschnig, A.; Kolagari, R.T.: Requirements Engineering in the Development of Innovative Automotive Embedded Software Systems. In: Proc. 12th IEEE Int. Req. Eng. Conf., RE'04, Kyoto, Japan, IEEE Comp. Society, 2004, 328–333.
- [OMG08] Object Management Group: OMG Systems Modeling Language (OMG SysML). Version 1.1, 2008. <http://www.omg.org/spec/SysML/1.1/PDF/>
- [OMG09] Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.2, 2009. <http://www.omg.org/spec/UML/2.2/Superstructure/>
- [PS09] Sikora, E; Pohl, K.: COSMOD-RE – Verzahnung des Architekturentwurfs mit dem Requirements Engineering. OBJEKTSpektrum, Online-Themenspecial: Architekturen, 2009.

# A System for Seamless Abstraction Layers for Model-based Development of Embedded Software\*

Judith Thyssen, Daniel Ratiu, Wolfgang Schwitzer, Alexander Harhurin, Martin Feilkas  
Technische Universität München

Institut für Informatik, Lehrstuhl für Software und Systems Engineering  
{thyssen,ratiu,schwitze,harhurin,feilkas}@in.tum.de

Eike Thaden

OFFIS - Institute for Information Technology  
eike.thaden@offis.de

**Abstract:** Model-based development aims at reducing the complexity of software development by the pervasive use of adequate models throughout the whole development process starting from early phases up to implementation. In this paper we present a conceptual framework to holistically classify developed models along different levels of abstraction. We do this by defining adequate abstractions for different development stages while ignoring the information that is not relevant at a particular development step or for a certain stakeholder. The abstraction is achieved in terms of the granularity level of the system under study (e. g. system, sub-system, sub-sub-system) and in terms of the information that the models contain (e. g. specification of functionality, description of architecture, deployment on specific hardware). We also present the relation between models that describe different perspectives of the system or are at different granularity levels. However, we do not address the process to be followed for building these models.

## 1 Motivation

Central challenges that are faced during the development of today's embedded systems are twofold: firstly, the rapid increase in the amount and importance of functions realized by software and their extensive interaction that leads to a combinatorial increase in complexity, and secondly, the distribution of the development of complex systems over the boundaries of single companies and its organization as deep chain of integrators and suppliers. Model based development promises to provide effective solutions by the pervasive use of adequate models in all development phases as main development artifacts.

Today's model-based software development involves different models at different stages in the process and at different levels of abstraction. Unfortunately, the current approaches do not make clear which kinds of models should be used in which process steps or how the transition between models should be done. This subsequently leads to gaps between

---

\*This work was mainly funded by the German Federal Ministry of Education and Research (BMBF), grant SPES2020, 01IS08045A.

models and thereby to a lack of automation, to difficulties in tracing the origins of modeling decisions made in different stages, and to difficulties in performing global analyses or optimizations that transcend the boundaries of a single model.

Our aim is to enable a seamless integration of models into model chains. A first step to achieve this is to understand which kinds of models are built, who are their main stakeholders, and the relation between them. To this end, we introduce a conceptual framework to holistically classify models that describe the system.

**Outline.** In Section 2 we introduce our modeling framework. In the following sections we detail the framework along two dimensions: on the vertical dimensions model elements specified on different granularity levels are mapped on each other. The specifications of elements on the lower granularity level form a refinement of the specification of their counter parts on the higher granularity level. A special kind of mapping is the mapping of one element on the higher level to a set of elements on the lower level which is similar to the classical decomposition (Section 3). On the horizontal dimension different development perspectives are related to each other (Section 4). In Section 6 we tackle the issue of crossing the different abstraction layers. Section 7 presents related work and Section 8 concludes the paper giving an outlook on future work.

## 2 Achieving abstraction

Abstraction means the reduction of complexity by reducing details. There are two typical possibilities to reduce details:

**Whole-part decomposition.** One manner to deal with complexity is to apply the “divide and conquer” principle and to decompose the whole system into smaller, less complex parts. These parts can be regarded as full-fledged systems themselves at a lower level of granularity. We structurally decompose the system into its sub-systems, and sub-systems into sub-sub-systems until we reach basic blocks that can be regarded as atomic.

**Distinct development perspectives.** The second manner to deal with complexity is to focus only on certain aspects of the system to be developed while leaving away other aspects that are not interesting for the aims related to a given development perspective. The essential complexity of a system is given by the (usage) functionality that it has to implement. By changing the perspective from usage functionality towards realization, the complexity is increased by considering additional implementation details (e. g. design decisions that enable the reuse of existing components).

Consequently, our modeling framework comprises two different dimensions as illustrated in Figure 1: one given by the *level of granularity* at which the system is regarded and the second one is given by different *software development perspectives* on the system.

**Levels of granularity.** A system is composed of sub-systems which are at a lower granularity level and which can themselves be regarded as systems (Section 3). Often, the sub-systems are developed separately by different suppliers and must be integrated afterwards. Especially for system integrators, the decomposition of a system into sub-systems

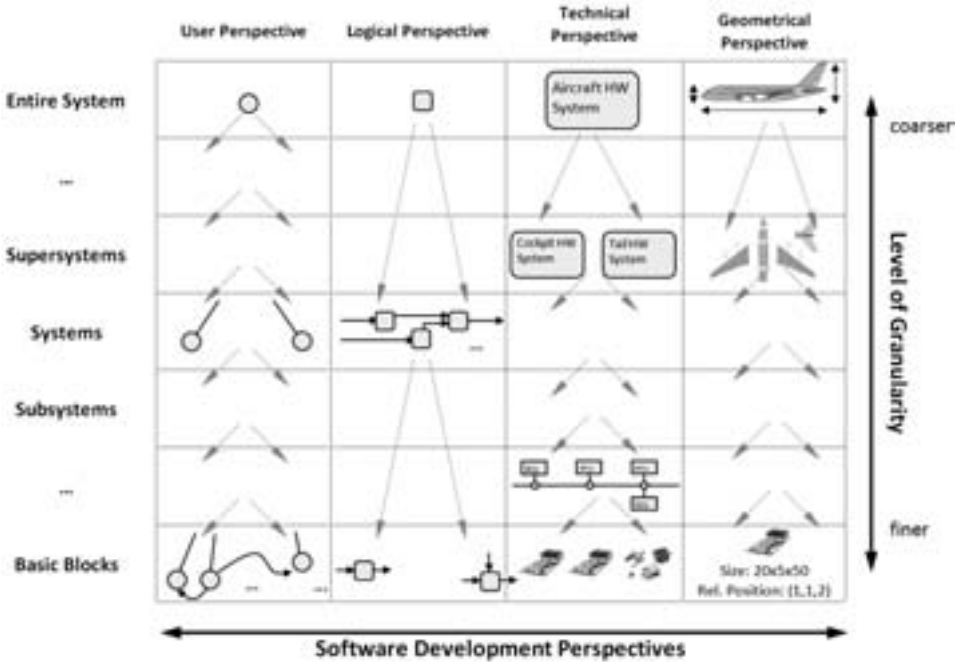


Figure 1: Two approaches to achieve abstraction: a) different granularity levels (vertical), and b) different software development perspectives (horizontal)

and subsequently the composition of the sub-systems into a whole are of central importance. As a consequence, we explicitly include the different granularity levels of systems in our framework (Figure 1 - vertical).

**Software development perspectives.** A system can be regarded from different perspectives, each perspective representing different kinds of information about the system. (Section 4). Our framework contains the following development perspectives: the user perspective, the logical (structural) perspective, the technical perspective, and the geometrical perspective. The perspectives aim to reduce the complexity of the software development by supporting a stepwise refinement of information from usage functionality to the realization on hardware: early models are used to capture the usage functionality and are step-by-step enriched with design and implementation information (Figure 1 - horizontal).

### 3 Granularity Levels

In order to cope with the complexity of today's systems, we decompose them into sub-systems. Each sub-system can be considered as a system itself and can be further decomposed until we reach basic building blocks. As a result we obtain a set of granularity levels upon which we regard the system, e. g. system level – sub-system level – sub-sub-system

level – ... – basic block level. These granularity levels enable us to seamlessly regard the systems at increasingly finer levels of granularity. Since the system is structured into finer parts which can be modeled independently and aggregated to the overall system afterwards, the granularity levels allow us to reduce the overall complexity following the “divide and conquer” principle.

Each system can be decomposed into sub-systems according to different criteria that are many times competing. For example, we might choose to decompose the system into sub-systems that map at best either the user, or the logical, or the technical architecture (these software development perspectives are presented in Section 4 in detail). Depending on which of these software development perspectives is prioritized, the resulting system decomposition looks different: the more emphasis is put on decomposing based on one of the software development perspectives, the more the modularization concerning the other software development perspectives is neglected and other aspects of modularity are lost (phenomenon known as “tyranny of dominant decomposition”).

On the one hand, for models situated at coarser granularity levels, the decomposition is many times driven by established industry-wide domain architectures. These domain architectures are primary determined by the physical/technical layout of a system, that is subsequently influenced by the vertical organization of the industry and division of labor between suppliers and integrators (most suppliers deliver a piece of hardware that contains the corresponding control software). On the other hand, for the models situated at finer granular layers, the decomposition is influenced by other factors, e.g. an optimal modularization of the logical component architecture in order to enable reuse of existing components. However, since the logical architecture acts as mediator between the user and technical perspective (see Section 4), we strongly believe that the logical modularization of the system decomposition should be reflected mainly in the part-whole decomposition of the system.

**Integrators vs. suppliers.** According to the level of granularity at which we regard the system, we can distinguish between different roles among stakeholders (illustrated in Figure 2): the end users are interested only in the top-level functionality of the entire system, the system integrators are responsible for integrating the high-granular components in a whole, while the suppliers are responsible for implementing the lower level components. An engineer can act as a system integrator with respect to the engineers working at finer granularity levels, and as supplier with respect to the engineers working at a higher granularity level. This top-down division of work has different depths depending on the complexity of the end product – for example, in the case of developing an airplane the granularity hierarchy has a high depth, meanwhile for developing a conveyor the hierarchy is less deep.

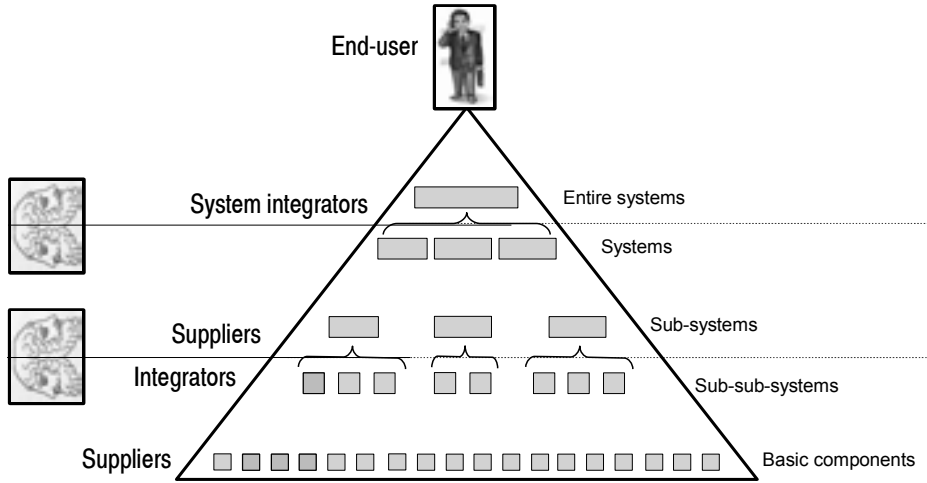


Figure 2: Suppliers vs. Integrators

## 4 Software Development Perspectives

The basic goal of a software system is to offer the required user functionality. There can be other goals that need to be considered such as efficiency, reliability, reuse of other existent systems or integration with legacy systems. However, in our opinion the functional goals are primordial since it is meaningless to build a highly efficient or reliable system that does not perform the desired functionality. Regarding the system purely from the point of view of the user functionality that it implements offers the highest level of abstraction since implementation, technical details and the other concerns required by the non-functional requirements are ignored (abstracted away). By changing the perspective from usage functionality towards implementation, we add more (implementation) details that are irrelevant for the usage and thus, the complexity of the system description is higher. We therefore propose a system of software development perspectives that allows to incrementally add more information to the system models. This refinement of models is inspired by the goal-oriented approach introduced by [Lev00], in which the information at one level acts as the goals with respect to the model at the next level. Our software development perspectives are: the *user perspective* that represents the decomposition of the system according to the behavior needed by its users, the *logical perspective* that represents the logical decomposition of the system and the realization of the software architecture, and the *technical perspective* that represents the technical implementation of the system. Additionally, the *geometrical perspective* represents the geometrical (physical) layout of the system, e. g. information about the shape of a hardware element or its concrete position in an airplane. From a systems engineering point of view, the geometrical perspective is essential. In this paper, however, we concentrate on the development of the software of the future systems. Thus, the geometrical perspective is not further detailed in the following.

**User perspective.** The user perspective describes the usage functionality that a system offers its environment/users. Thereby, a user can be both, the end user of the system (at the highest level of granularity) or a system integrator (at a lower level of granularity). The functionality that is desired from the system represents the most abstract – but nevertheless, most important – information about the system. During design and implementation, we add realization details that are irrelevant for the usage of the system (but nevertheless essential for its realization).

The central aims of the user perspective are: Hierarchical structuring of the functionality from the point of view of the system’s users; Definition of the boundary between the system functionality and its environment: definition of the syntactical interface and abstract information flow between the system and its environment; Consolidation of the functional requirements by formally specifying the requirements on the system behavior from the black-box perspective; Understanding of the functional interrelationships and mastering of feature interaction.

**Logical perspective.** The logical perspective describes how the functionality is realized by a network of interacting logical components that determines the logical architecture of the system. The design of the logical component architecture is driven by various considerations such as: achieving maximum reuse of already existent components, fulfilling different non-functional properties of the system, etc. The logical architecture bridges the gap between functional requirements and the technical implementation means. It acts as a pivot that represents a flexibility point in the implementation.

The main aims of the logical perspective are: Describing the architecture of the system by partitioning the system into communicating logical components; Supporting the reuse of already existent components and designing the components such that to facilitate their reuse in the future; Optionally: Definition of the total behavior of the system (as opposed to the partial specifications in the user perspective) and enabling the complete simulation of all desired functionalities; Mediation between the structure of the function hierarchy and that of the already existing technical platform on which the system should run.

Since the functions of the user perspective are defined by given user requirements and the prerequisites of the technical layer are primarily given a priori, from a software development point of view, the main engineering activities are concentrated on the logical component architecture. Thereby, the logical architecture should be designed in order to capture the central domain abstractions and to support reuse. As a consequence, the logical architecture should be as insensitive as possible to changes of the desired user functionality or technical platform. It should be the artifact in the development process with the highest stability and with the highest potential of reuse.

**Technical perspective.** The technical perspective comprises the hardware topology on which the logical model is to be deployed. For us hardware means entities on which the software runs (ECUs), or that directly interact with the software (sensors/actors). On higher granularity levels hardware entities can also be abstractions/aggregations of such entities.

In the technical perspective engineers need to consider hardware related issues such as throughput of communication, bandwidth, timing properties, the location of different hardware parts, or the exact interaction of the software system with the environment.

The main aims of the technical perspective are: Describing the hardware topology on which the system will run including important characteristics of the hardware; Describing the actuators, sensors, and the HMI (human-machine interaction) that are used to interact with the environment; Implementation and verification of real-time properties in combination with a deployment of logical components; Ensuring that the behavior of the deployed system (i. e., the hardware and the software running on it) conforms to the specifications of the logical layer.

*Note:* One of the main advantages of the clear distinction between the logical and technical architecture is that it enables a flexible (re-)deployment of the logical components to a distributed network of ECUs. If the hardware platform changes, the logical components only need to be (re-)deployed, but the logical architecture does not need to be redesigned.

## 5 Core-models and their Decorations (Viewpoints)

In order to enable the extensibility of models with additional information relevant for the realization of non-functional requirements (e. g. failure models) or even other development disciplines (e. g. mechanical information), we enable the use of decorators (also called Viewpoints).

Each of the three software development perspectives is usually represented by a dominating *core-model* (e. g. functions hierarchies in the user perspective, networks of components in the logical perspective) and may provide a number of additional *decorator-models*. Decorator-models are specialized for the description of distinct classes of functional and non-functional requirements that are relevant to their respective software development perspectives. Decorator-models enrich the core-models with additional information that is necessary for later steps in the software development process or for the integration with other disciplines. For example, this could be safety analyses, scheduling generation or deployment optimizations. The complexity of decorator-models is arbitrary and their impact on the overall system functionality may be significant. Failure-models are an example of usually quite complex decorator-models to the models of the logical architecture. The impact of failure-models to the overall functionality is usually relatively critical, too. Other examples for existing decorations concerning the technical perspective are information concerning physical, mechanical or electrical properties of the technical system under design.



## 6 Relating the Models

In the previous sections (Sections 3 and 4) we detailed two different manners to achieve abstraction: by providing mappings between elements on different granularity levels and by using different software development perspectives. Thereby we obtain models (contained in each cell of the table in Figure 1) that describe the system either at different granularity levels or from different software development points of view. In this section we discuss the relation between models in adjacent cells (horizontally from left to right, and vertically between two consecutive rows).

**Horizontal allocation (mapping models at the same granularity level).** In general, there is a many-to-many (n:m) relation between functions (user perspective) and logical components (logical perspective) that implement them, respectively between logical components and hardware on which they run. However, in order to keep the relations between models relatively simple, we require the allocation to be done in a many-to-one manner. Especially, we do not allow a function to be scattered over multiple logical components or a logical component to run on multiple hardware entities, respectively. If necessary, the user perspective should be further decomposed in finer granular parts until an allocation of each function on individual components is possible. In a similar manner, the logical components should be fine granular enough to allow a many-to-one (n:1) deployment on hardware units.

Allocation represents decision points since each time a transition between an abstract to a more concrete perspective is done engineers have to decide for one of several possible allocations/deployments.

*Note.* It can happen (especially at coarse levels of granularity) that there is an isomorphism between the decompositions realized in different perspectives (e. g. that the main sub-functions of a complex product are realized by dedicated logical components that are run on dedicated hardware).

**Vertical allocation (transition from systems to sub-systems).** Vertical allocation means the top down transition from systems to sub-systems that happens typically at the border between suppliers and integrators – sub-systems built by suppliers are integrated into larger systems by integrators (see Figure 2).

In Figure 3 we illustrate the transition between two subsequent granularity levels generically named “system” and “sub-systems”. The sub-systems of a system can be determined by the structural decomposition in the logical or hardware perspectives (see Section 3). More complex mappings between elements on different granularity levels are possible, too, but are not further considered in this paper. The structure of the system defines its set of components and how they are composed. Each leaf component of the system structure determines a new system at the next granularity level. For example, in Figure 3 (top-right) the structural decomposition at the system level contains three components. Subsequently, at the next level of granularity we have three sub-systems that correspond to the components.

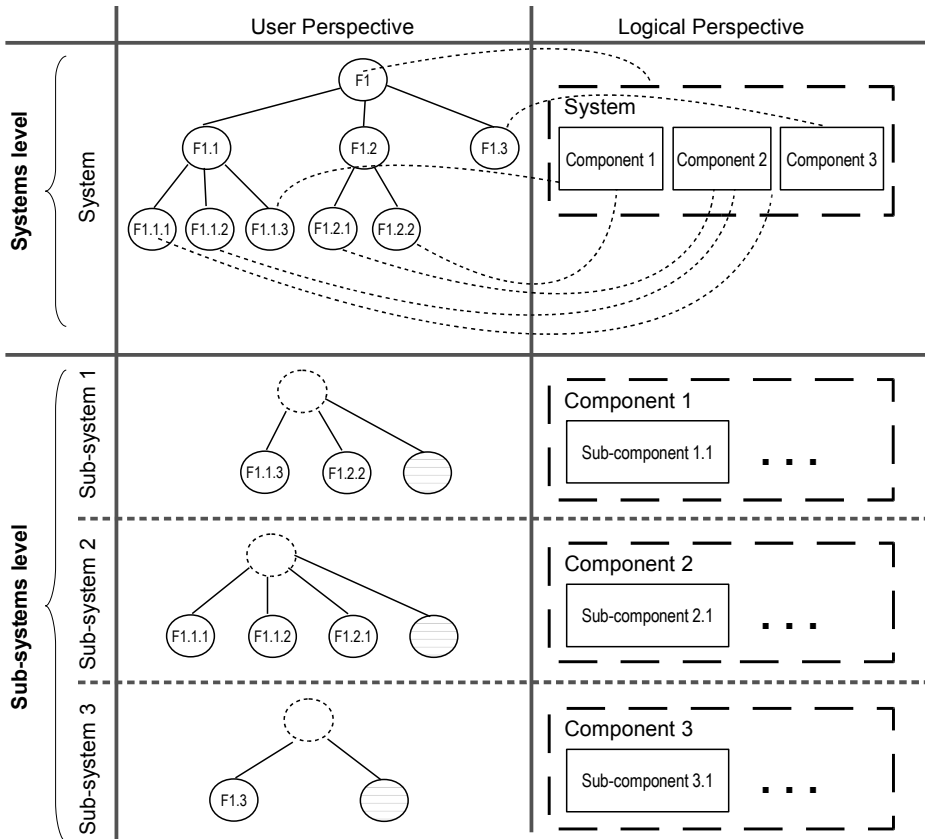


Figure 3: Structural decomposition of the system is one way to define the sub-systems situated at the subsequent granularity level

Generally, the functionality allocated to one of the components of the system defines the functional requirements for its corresponding sub-system. Each sub-system carries the user functionality allocated to its corresponding component at a higher granularity level. For example, in Figure 3 to the “Component 1” component was allocated the F1.1.3 and F1.2.2 functions. These functions should be implemented by the “Sub-system 1” that corresponds to “Component 1” on the next level of granularity. In addition to the original user functions allocated to components at the system level, at the sub-system level new functionality is required due to design (and implementation) decisions taken at the system level. This functionality is needed in order to allow the integration of the sub-systems into the systems and can be seen as a “glue” functionality. In Figure 3 we pictured the glue functionality by hashed circles. The root representing the entire functionality of a sub-component (not-existent at the system level) is pictured through a dotted circle.

## 7 Related Work

Our approach to reduce complexity by a systematic software development along different abstraction layers is part of a more general area of research about the pervasive and disciplined use of models along the development process.

MDA [MM03] represents a well known approach to master the complexity of today's systems by describing the system on different levels of abstraction starting with an informal description of the system (known as "Computation Independent Model"). Based on this, the Platform Independent Model (PIM) defines the pure system functionality independently from its technical realization and at last the PIM is translated to one or more Platform Specific Model (PSM) that can run on a specific platform. While we are aiming at a modeling framework for the development of embedded systems or even specific domains, the MDA is a general purpose approach. The different development perspectives of our modeling framework can be seen as instantiation of the MDA-layers. Furthermore, the MDA layers do not address issues related to the vertical decomposition of systems into sub-systems down to basic components.

The EAST ADL (Electronics Architecture and Software Technology – Architecture Definition Language, [ITE08]) has been designed for the automotive domain and describes software-intensive electric/electronic systems in vehicles on five different abstraction layers starting from high-level requirements and features which are visible to the user to details close to implementation, such as constructs of operating systems (e. g. tasks) and electronic hardware. In contrast to our approach, the EAST ADL does not clearly distinguish the different dimension of abstraction, namely the decomposition in sub-systems and the different perspectives on a system. However, the EAST ADL can be seen as instantiation of our modeling framework. With regard to contents and aims the Vehicle Feature Model and the Functional Analysis Architecture of the EAST ADL can be seen as a counterpart of the functional perspective, the Functional Design Architecture vaguely corresponds to the logical perspective and the abstraction levels of the Function Instance Model, the Platform Model and the Allocation Model vaguely correspond to the abstraction level, which can be found on the technical perspective.

The idea to describe a system from different perspectives is not new. For example, the 4+1 View Model by Kruchten [Kru95] provides four different views of a system: the logical, process, development and physical views. The views are not fully orthogonal or independent – elements of one view are connected to elements in other views. The elements in the four views work together seamlessly by the use of a set of important scenarios. This fifth view is in some sense an abstraction of the most important requirements. However, since this approach is not based on a proper theory and sufficient formalization, a deeper benefit is not achieved. As a result, possibilities to do analysis with these views are weak, and the models are applied only at particular stages of the development process without a formal connection between them. In contrast, the introduced approach aims at "theoretical foundations of a strictly model based development in terms of an integrated, homogeneous, but yet modular construction kit for models" [BR07].

In [BFG<sup>+</sup>08], a first step has been made to integrate the existing research results into an

integrated architectural model for the development of embedded software systems. The architectural model comprises three subsequent layers, namely the service layer, the logical layer, and the technical layer. This architectural model served as basis for the modeling framework presented here. The different layers are reflected by the different software development perspectives of our modeling framework. In the current work we extended the architectural model by introducing granularity levels as a second dimensions.

## 8 Future Work

In this paper we presented different abstraction layers at which the models used to realize a software product should be categorized. There are, however, many open issues that are subject to current and future work: 1) *methodology* – the steps that should be performed to instantiate the layers is of capital importance for the realization of the software product; 2) *allocation* – based on which criteria is the allocation of functionalities on logical components and of logical components on technical platform made; 3) *models* – define which modeling techniques would fit at best to describe particular aspects of the system (e. g. functionality) at particular granularity level (e. g. entire system).

**Acknowledgements:** Early ideas of this paper originate from a discussion with Carsten Strobel, Alex Metzner, and Ernst Sikora.

## References

- [BFG<sup>+</sup>08] Manfred Broy, Martin Feilkas, Johannes Grünbauer, Alexander Gruler, Alexander Harhurin, Judith Hartmann, Birgit Penzenstadler, Bernhard Schätz, and Doris Wild. Umfassendes Architekturmodell für das Engineering eingebetteter Software-intensiver Systeme. Technical Report TUM-I0816, Technische Universität München, June 2008.
- [BR07] Manfred Broy and Bernhard Rumpe. Modulare hierarchische Modellierung als Grundlage der Software- und Systementwicklung. *Informatik Spektrum*, 30(1):3–18, 2007.
- [ITE08] ITEA. EAST-EEA Website. <http://www.east-eea.net>, January 2008.
- [Kru95] Philippe Kruchten. Architecture Blueprints - the "4+1" View Model of Software Architecture. In *TRI-Ada Tutorials*, pages 540–555, 1995.
- [Lev00] Nancy G. Leveson. Intent Specifications: An Approach to Building Human-Centered Specifications. *IEEE Trans. Software Eng.*, 26(1):15–35, 2000.
- [MM03] J. Miller and J. Mukerji. MDA Guide Version 1.0.1. Technical report, Object Management Group (OMG), 2003.



# Eine erweiterte Systemmodellierung zur Entwicklung von softwareintensiven Anwendungen in der Automobilindustrie\*

Jörg Holtmann, Jan Meyer, Wilhelm Schäfer und Ulrich Nickel

chrome|janny|wilhelm@upb.de

Ulrich.Nickel@hella.com

**Abstract:** Heutige Transportsysteme, wie z. B. Automobile sind gekennzeichnet durch eine Vielzahl von Funktionalität, die sehr häufig durch Software realisiert wird. Hiermit ist eine Zunahme der Komplexität festzustellen. Zur Beherrschung dieser Komplexität und damit einhergehend die Aufteilung des Systems in verschiedene Komponenten, ist eine Gesamtmodellierung des Systems inklusive des Verhaltens unerlässlich. Ein besonderer Augenmerk liegt, aufgrund der Echtzeitsysteme in dieser Domäne, auf der Modellierung von Zeiten auf Systemebene. Die derzeitigen Modellierungskonzepte, wie beispielsweise die Systems Modeling Language (SysML), sind dafür aber noch nicht ausreichend. In dem hier vorgestellten Ansatz wird eine erweiterte Systemmodellierung vorgestellt, die zusätzlich eine formale Spezifizierung von Zeiten erlaubt. Durch diese Modellierung sind weitere Analysemethoden, wie z. B. Simulationen oder Verifikationen möglich, die zum einen die sicherheitsrelevante Funktionalität sicherstellen und zum anderen die Qualität der Software steigern.

## 1 Einleitung

Eingebettete Systeme sind nicht wegzudenkende Bestandteile des täglichen Lebens. Sie lassen sich in einfachen Haushaltsgeräten bis hin zu komplexen Transportsystemen etc. finden. Bei der Entwicklung aller eingebetteten Systeme treten vergleichbare Probleme auf. So werden beispielsweise immer mehr Sensoren und bestehende Systeme miteinander vernetzt, um eine möglichst genaue Steuerung der Systeme zu ermöglichen. Die hiermit einhergehende Komplexität, insbesondere bei der zeitlichen Spezifikation des Verhaltens, erschwert die Entwicklung. Zur Darstellung dieser Problemstellung und einer entsprechenden Lösung wird hier stellvertretend auf die Entwicklung eines eingebetteten Systems aus dem Automobilbereich eingegangen.

Die Vielzahl von Innovationen im Automobilbereich, welche zu großen Anteilen durch Software realisiert werden, lässt die Komplexität des Gesamtsystems „Kraftfahrzeug“ bekanntermaßen ansteigen [Gri03]. Grund hierfür ist die hohe Anzahl von Steuergeräten

---

\*Diese Arbeit entstand im Rahmen des SPES2020 Projektes gefördert durch das Bundesministerium für Bildung und Forschung (BMBF), "SPES2020,01IS08045H"

(Embedded Control Units - ECUs), welche über heterogene Netzwerke miteinander kommunizieren und letztendlich ein komplexes Netzwerk von Funktionen realisieren. Betrachtet man die einzelnen ECUs eines Fahrzeugs näher, so ist auch hier ein Zuwachs an Komplexität zu erkennen. Leistungsfähigere Hardware ermöglicht die Integration von Funktionen auf einer ECU, welche vormals auf mehrere ECUs verteilt waren. Darüber hinaus machen einige Funktionen, z. B. Fahrerassistenzsysteme, welche auf einer Echtzeit-Bilddatenverarbeitung basieren, zudem den Einsatz mehrerer Rechenkerne (Mikrocontroller, DSPs oder FPGAs) unerlässlich.

Hieraus ergeben sich neue Problemstellungen, die es zu lösen gilt. Zunächst müssen die einzelnen Funktionen möglichst optimal auf die jeweiligen Rechenkerne aufgeteilt werden. Somit ergibt sich ein, für die Performance wichtiges, Optimierungsproblem. Die Zuordnung von unterschiedlichsten Funktionen auf einen Rechenkern, die sich ggf. noch die Rechenleistung gegenseitig wegnehmen, während gleichzeitig ein anderer Rechenkern nicht voll ausgelastet ist, spricht dafür, dass das System insgesamt nicht optimal implementiert ist. Eine weitere Vorgabe bei der steigenden Funktionalität ist die Sicherstellung des korrekten Aufrufs der Funktionen, der Bereitstellung der Daten und der Kommunikation zwischen ihnen [Bro06]. Diese Problemfelder müssen frühzeitig bei der Entwicklung erkannt und überprüft werden. Spätere Anpassungen können größere Umstrukturierungen und somit auch höhere Kosten verursachen. Von daher müssen bereits in frühen Entwicklungsphasen Methoden zur formalen Spezifikation des Systemverhaltens inklusive der zeitlichen Anforderungen bereitgestellt werden. Nur diese bieten dann auch Möglichkeiten für Analysemethoden bieten, beispielsweise um die Verteilung der Funktionen zu unterstützen. Derzeit werden diese Probleme mit dem Erfahrungswissen der Entwickler und aufwendigen Integrationsphasen behoben. Dies ist aber ein kostenintensiver Ansatz.

Eine Besonderheit in der Automobilbranche ist, dass große Teile der Entwicklung an Zulieferer zwecks Realisierung ausgelagert werden. Deshalb werden vom Automobilhersteller (OEM) Anforderungen erstellt und mit dem jeweils beauftragten Zulieferer ausgetauscht [WW03]. Die Anforderungen sind zum Teil noch unvollständig. Dies ist vor allem der Fall, wenn das Wissen und die Experten beim Zulieferer zu finden sind. Alle Anforderungen werden in der Anforderungsanalyse genauer analysiert und ggf. durch das explizite Erfahrungswissen des Zulieferers ergänzt. Aus den Anforderungen wird schließlich ein initiales Systemmodell (Architektur und Verhalten) erstellt. Diese ist die Basis für die weitere Entwicklung und für Designentscheidungen, wie z. B. die Aufteilung der Funktionalität.

Bei einem eingebetteten System in der Automobilindustrie spielt die Echtzeitfähigkeit und somit die Einhaltung der zeitlichen Anforderungen eine wesentliche Rolle. Dies ist besonders bei sicherheitskritischen Funktionen (Airbag, Steuerung, Bremssystem, etc.) notwendig. Die zeitlichen Anforderungen lassen sich auf allen Ebenen und in unterschiedlichsten Detailstufen wiederfinden. Beispiele für die Systemebene werden anhand eines Komfortsteuergerätes in Kapitel 4 vorgestellt. Die für das Gesamtsystem relevanten zeitlichen Anforderungen müssen in die Systemmodellierung integriert werden. Dies bedeutet, dass bei globalen zeitlichen Anforderungen diese auf einzelne Systembestandteile aufgeteilt werden müssen. Dies kann mitunter in Form von Zeitbudgets erfolgen. Damit die Aufteilung unterstützt durch Analysen bzw. automatisch erfolgen kann, müssen natürlich das Gesamt-

systemverhalten und auch die zeitlichen Anforderungen formal spezifiziert sein.

Mit den heutigen Modellierungssprachen ist es nur bedingt möglich, eine Systemarchitektur mit gleichzeitiger formaler Spezifikation des Systemverhaltens zu modellieren. Dies wird in Kapitel zwei (Stand der Technik) dargelegt. Es werden die verschiedenen Systemmodellierungssprachen, die derzeit in der Industrie eingesetzt werden, mit ihren Vor- und Nachteilen gegenübergestellt. Aus den dort dargelegten Argumenten und den Problemstellungen aus Kapitel eins ergibt sich, dass die bisherigen Ansätze nicht ausreichen und daher erweitert werden müssen. Von daher wird im dritten Kapitel dieses Papiers ein Ansatz einer erweiterten Systemmodellierung vorgestellt, der eine formale Spezifizierung von Verhalten ermöglicht. Hierfür verwenden wir die Systems Modeling Language (SysML) durch eine formale Zeitmodellierung ergänzt. Der Ansatz ermöglicht darüber hinaus auch weitergehende Analysetechniken, beispielsweise Simulationen oder Model Checking Techniken. Die Ergebnisse der Analyseverfahren können z. B. dazu genutzt werden das Verteilungsproblem der einzelnen Funktionen zu lösen. Im vierten Kapitel wird anhand eines Beispiels die Verwendung des vorgestellten Ansatzes aufgezeigt. Zum Schluss wird eine Zusammenfassung und ein Ausblick gegeben, inwieweit der Ansatz in Zukunft noch ausgebaut werden kann, um auch zukünftige Problemstellungen abzudecken.

## 2 Stand der Technik

Es existieren verschiedene Techniken, um ein softwareintensives System zu modellieren und zu analysieren. In diesem Kapitel werden die gängigsten Modellierungssprachen auf ihre Nutzung bezüglich eingebetteter Systeme hin untersucht. In [GH06] werden verschiedene Modellierungssprachen auf ihre Eignung bezüglich der Modellierung von softwareintensiven Systemen untersucht. Auch die in der Industrie weitverbreiteten Sprachen MATLAB<sup>1</sup> und SysML wurden näher untersucht. Als Ergebnis dieser Studie bleibt festzuhalten, dass die Modellierungssprachen für zukünftige softwareintensive Systeme sowohl die Struktur, aber auch das Verhalten formal spezifizieren müssen. Ebenso ist die Modellierung von Zustandsautomaten mit hybridem Verhalten und Uhren notwendig [GH06]. Damit einhergehend ist eine Anbindung an Analysewerkzeuge, exemplarisch sei hier auf das Model Checking hingewiesen, erforderlich. Unter diesen Gesichtspunkten hat sich gezeigt, dass der Ansatz „Mechatronic UML“ mit den darin enthaltenen Real-Time Statecharts am Besten abschneidet. Deshalb werden diese im Ansatz auch wiederverwendet.

### MATLAB/Simulink

Im Bereich der Regelungstechnik wird in der Industrie sehr häufig das Werkzeug MATLAB/Simulink verwendet. Mit der zugrunde liegenden Modellierungssprache ist es möglich, kontinuierliche Systeme zu modellieren. Zugleich ist häufig der Ansatz zu finden, die Sprache auch für die Spezifizierung von diskreten Anteilen und sogar ganzen Systemen zu nutzen. Es ist somit möglich hybride Systeme zu entwickeln. Hierbei können auch zeitliche Anforderungen berücksichtigt werden. Diese Modellierungsvariante bietet weiterhin den Vorteil, dass eine Simulation des spezifizierten Verhaltens möglich ist. Daher ist es

---

<sup>1</sup>[www.mathworks.com](http://www.mathworks.com)



grundsätzlich möglich, ein zeitliches Verhalten einer Komponente bzw. eines Systems zu spezifizieren. Es werden aber keine Uhren mit diesem Ansatz unterstützt [GH06]. Jedoch sind das Werkzeug und somit auch die Sprache nicht dafür konzipiert, die Struktur (Architektur) zu erstellen. Dies ist zwar mit verschiedenen Hilfsmethoden durchaus machbar, jedoch entstehen dann sehr unübersichtliche und kaum mehr wartbare Modelle. Von daher ist von der Modellierung der Struktur eines Systems mit MATLAB/Simulink abzuraten.

**Systems Modeling Language (SysML)** Die Systems Modeling Language (SysML) ist eine Modellierungssprache zur Spezifizierung von Systemen und wurde von der Object Management Group (OMG)<sup>2</sup> entwickelt [Sys08]. Die SysML beschreibt die Systeme dabei auf einer abstrakten Art, sodass noch nicht entschieden ist, in welcher Ingenieursdisziplin sie später realisiert werden. So können sowohl Hardware, Software, aber auch mechanische Bestandteile modelliert werden. Sie ist auf Basis der UML 2.0 entwickelt worden. Dabei ist ein Teil der UML, vor allem die Verhaltensmodellierung, übernommen worden. Zur Anpassung an die Systemmodellierung sind zusätzliche Elemente und Diagramme hinzugekommen, wie beispielsweise das Anforderungs- das Blockdefinitions- und das Zusage- und Sicherungsdiagramm. Im Gegensatz zur UML 2.0 gehören jedoch Diagramme, die nur zur Beschreibung von Software genutzt wurden, wie das Objektdiagramm, nicht zur SysML [Wei08]. Insgesamt ist zur SysML zu sagen, dass ihre Vorteile besonders bei der Modellierung der Architektur zu sehen sind. Es ist zwar auch möglich das Verhalten zu modellieren, aber die SysML bietet nur eingeschränkte Möglichkeiten zur Zeitmodellierung. Es beschränkt sich zum größten Teil auf das When- und das After Konstrukt. Dies ist für die Spezifizierung von Echtzeitsystemen jedoch nicht ausreichend.

**Zusätzliche Profile für die UML/SysML** Die SysML besitzt, wie auch die UML, einen Erweiterungsmechanismus in Form von Profilen. Hierdurch können die Sprachen erweitert und an bestimmte Domänen angepasst werden. Zwei von der OMG spezifizierte Profile befassen sich mit der Performance Modellierung. Diese können auch für eingebettete Systeme genutzt werden. Die beiden Profile sind das Profil für Schedulability, Performance and Time (SPT) [Spt05] und das Nachfolgerprofil Modeling and Analysis of Real-time and Embedded Systems (MARTE) [OMG07]. Beide Profile bieten die Möglichkeit der Modellierung von Zeiten, Nebenläufigkeit und Performance. In den Profilen wird aber nirgends ausgesagt, welcher Stereotyp eindeutig für welche Information verwendet werden muss. Somit sind unterschiedlichste Modellierungsformen möglich. Ferner fehlen Aussagen darüber, wie die neuen Informationen in den verschiedenen Verhaltensdiagrammen integriert werden können, beispielsweise in den Zustandsdiagrammen [HH07]. Außerdem bieten sie keine Möglichkeit, mangels einer präzisen semantischen Definition, die Modelle automatisch auf Korrektheit und Konsistenz zu überprüfen. Somit ist eine eindeutige und formale Modellierung nur bei rigiden Modellierungsvorschriften gegeben und die Verwendung der Profile zu einer anschließenden Systemanalyse nur eingeschränkt möglich.

Es existiert auch noch ein an die Automobilindustrie angepasstes Profil der UML. Dies ist die sogenannte EAST-ADL2 [Ead08]. Dies Profil stellt Stereotypen und Eigenschaftswerte zur Verfügung, die typische Elemente aus der Automobilindustrie, wie beispielsweise die verschiedenen Bussysteme, repräsentieren. Zur Beschreibung von Verhalten können verschiedene Modellierungssprachen genutzt werden, wie Anwendungsfälle und Zustandsau-

---

<sup>2</sup>[www.omg.org](http://www.omg.org)

tomaten [HKM07]. Somit können auch dort nur eingeschränkt Zeiten modelliert werden.

**Fazit** Ausgehend von diesen Untersuchungen bleibt festzuhalten, dass bei allen bestehenden Ansätzen Defizite vorhanden sind. Während einige Ansätze besser die Architektur darstellen können, sind andere besser geeignet, das Verhalten zu modellieren. Eines ist aber bei allen Ansätzen gemeinsam. Bei der Spezifikation von Zeiten und der Anbindung an Verifikations- und Simulationsumgebungen gibt es noch Handlungsbedarf, denn es fehlt ein geeigneter Formalismus zur Spezifikation von Zeiten.

### 3 SysML ergänzt um Real-Time Statecharts

Wie aus der Motivation zu erkennen ist, bedarf es zur Modellierung von softwareintensiven Systemen in der Automobilindustrie sowohl der Systemarchitektur, als auch des Systemverhaltens. Diese können, wie in Kapitel 2 beschrieben, zurzeit nur unzureichend mit den bisherigen Konzepten modelliert werden. Deshalb wird im Folgenden ein Ansatz für eine Systemmodellierung vorgestellt, der sowohl die Systemarchitektur als auch das Verhalten formal modellieren kann. Wie bereits beschrieben, bietet sich zur Modellierung der Systemarchitektur die SysML an. Vor allem auch, weil sie ein internationaler Standard im Bereich der Systemmodellierung ist. Die angesprochenen Nachteile (Verhaltens- und Zeitmodellierung) lassen sich durch eine Erweiterung der SysML beheben. Hierzu werden die sogenannten Real-Time Statecharts (RTSC) verwendet [GB03] [GTB<sup>+</sup>03].

Die Real-Time Statecharts wurden im Sonderforschungsbereich 614 „Selbstoptimierende Systeme des Maschinenbaus“ [SFB09] entwickelt. Der dort entwickelte Ansatz ermöglicht neben der Modellierung der Struktur auch die Modellierung des Verhaltens von selbstoptimierenden mechatronischen Systemen. Für die Spezifikation des Echtzeitverhaltens werden Real-Time Statecharts genutzt. Diese definieren zeitbehaftete UML Zustandsdiagramme auf der Grundlage von Timed Automata. Sie sind somit formal definiert. Im Gegensatz zu den UML Zustandsdiagrammen bringen die Real-Time Statecharts die Möglichkeit zur Modellierung von Zeiten, wie zum Beispiel Uhren, Worst Case Execution Times (WCET) etc. mit. Die RTSCs vereinen die Konstrukte der Statecharts mit der Semantik der Timed Automata und ermöglichen einen besseren Überblick über das Verhalten als Timed Automata an sich [GB03].

Im folgenden Absatz werden die Möglichkeiten der Real-Time Statecharts dargestellt. In der Abbildung 1 ist ein Beispiel eines Real-Time Statecharts zu sehen, welches aus zwei Zuständen besteht. Das Beispiel dient dazu die verschiedenen Modellierungsmöglichkeiten zu erläutern, die mit einem Real-Time Statechart möglich sind, wobei der letzte Zustand nicht mehr verlassen werden kann. Der Zustand S1 besitzt die Zeitinvariante  $t_0 \leq 5$  ms. Bei Eintritt in diesen Zustand wird die Uhr  $t_0$  auf 0 zurückgesetzt und die Operation `entryS1()` mit der WCET  $w = 1$  ms ausgeführt. Während des Aufenthaltes in S1 wird die Methode `doS1()` mit einer WCET von  $w = 1$  ms und einer Periode, die zwischen 2ms und 3ms liegt, ausgeführt. Die dargestellte Transition schaltet, wenn das Ereignis `e` anliegt, der Guard  $x \leq 2$  und der Zeitguard  $1ms \leq t_0$  wahr sind, wobei `x` eine beliebige Integer Variable ist. Beim Schaltvorgang wird als Seiteneffekt `action()` ausgeführt, der eine WCET

von  $w = 2\text{ms}$  besitzt. Er muss spätestens  $10\text{ms}$  nach Transitionsaktivierung, aber auch zum Zeitpunkt  $t_1 = 6\text{ms}$  beendet sein. Der Seiteneffekt darf frühestens bei  $t_1 = 3\text{ms}$  beendet werden. Zum Aktivierungszeitpunkt der Transition wird die Uhr  $t_2$  zurückgesetzt. Die Invariante des Zustands S2 lautet  $t_0 \leq 20\text{ms}$  und  $t_1 \leq 13\text{ms}$ . Bei Verlassen des Zustands werden die Uhren  $t_0$  und  $t_1$  wieder zurückgesetzt.

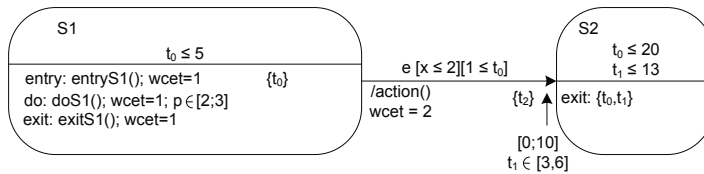


Abbildung 1: Beispiel eines Real-Time Statecharts [Bur02]

Nachdem anhand des Beispiels in Abbildung 1 die Möglichkeiten aufgezeigt wurden, die mit einem Real-Time Statechart möglich sind, wird im Folgenden ein Konzept beschrieben, wie die Statecharts der SysML erweitert werden können, um die zeitlichen Konzepte der Real-Time Statecharts umzusetzen. Diese ermöglichen später eine Analyse der vorgenommenen Informationen. Die Erweiterungen der Real-Time Statecharts, die sich mit der Modellierung von hybriden Systemen beschäftigen, werden dabei zunächst außer Acht gelassen, da sie für die frühen Modellierungsphasen nicht benötigt werden [GBS04].

In der SysML wird zur Spezifikation des Verhaltens u.a. das Zustandsdiagramm der UML 2 übernommen. Darum ist dieses Metamodell zu untersuchen, um Erweiterungspunkte zu finden, an denen die Konzepte der Real-Time Statecharts anknüpfen können. Vergleicht man die beiden Metamodelle von UML- und Real-Time Statecharts, so bleibt festzuhalten, dass die RTSCs alle - mit Ausnahme des After- und des When-Konstruktes - Eigenschaften der UML Statecharts aufweisen. Die Aussagen dieser beiden Konstrukte ist aber implizit in den Real-Time Statecharts gegeben. Jedoch sind sie in zusätzlichen Elementen verborgen, die eine darüber hinausgehende Spezifikation von Zeit erlauben. Von daher lassen sie sich nach modellieren.

Ein RTSC Zustand erweitert einen Zustand der UML um folgende Konstrukte: Zeitinvarianten, Uhren-Resets, WCETs und ein Periodenintervall. Aber nicht nur der Zustand muss erweitert werden. Auch die UML Transitionen müssen im Metamodell ergänzt werden und zwar um die folgenden Konstrukte: Zeitguards, Uhren-Resets, Prioritäten, Deadlines, WCETs und Synchronisationskanäle mit Synchronisationsarten.

Ausgehend von diesen Ergänzungen des SysML bzw. des UML Zustandsdiagramms ergibt sich das folgende erarbeitete Metamodell (vgl. Abbildung 2) eines erweiterten Zustandsdiagramms, in dem auch die Möglichkeit zur Modellierung von Zeiten und verteilter Kommunikation möglich ist.

In der Abbildung 2 ist das erweiterte Metamodell der SysML bzw. der UML zu sehen. In diesem sind die eben herausgearbeiteten notwendigen Erweiterungen zu sehen, wie exemplarisch die Invarianten [MH09]. Die neu hinzugekommenen Elemente sind dabei gelb (grau schraffiert) und die neuen Assoziationen sind dicker und in grün dargestellt.

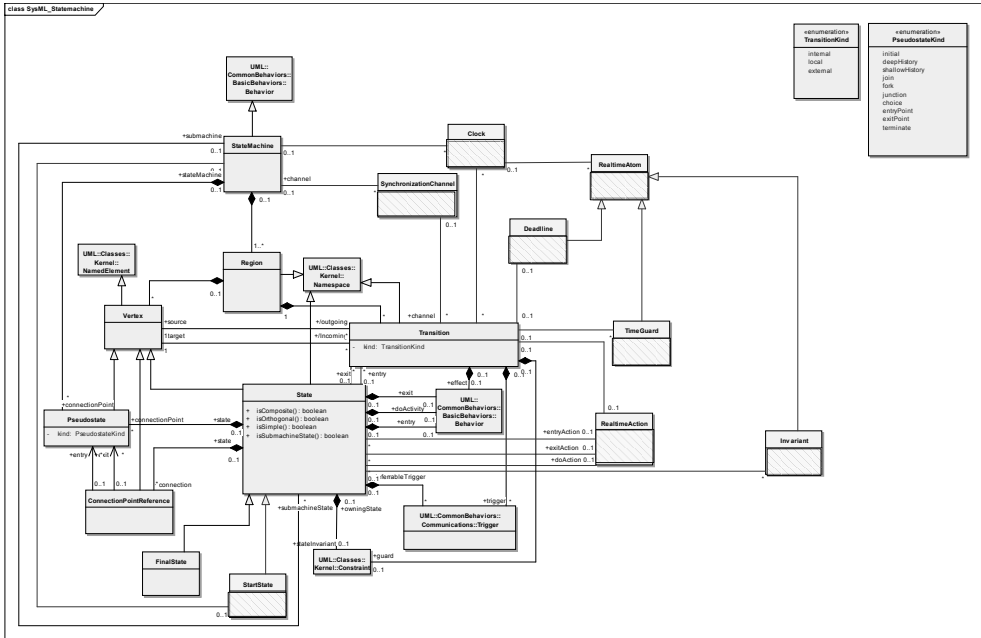


Abbildung 2: Erweitertes MetaModell des UML Zustandsdiagramms

Die Erweiterungen beziehen sich hauptsächlich auf die Zeitmodellierung, wie z. B. durch Uhren etc. Mit diesem erweiterten Metamodell ist die Kombination von SysML und Real-Time Statecharts möglich.

## 4 Verwendung des Ansatzes

In dieser Ausarbeitung wird als Beispiel für die Benutzung des eben vorgestellten Ansatzes von SysML mit einer Erweiterung von Real-Time Statecharts ein Komfortsteuergerät verwendet. Die Hauptaufgabe eines solchen Steuergerätes ist die Überwachung der Zentralverriegelung inklusive Empfang und Auswertung von Signalen der Funkfernbedienung. Die Funktionalität ist in den letzten Jahren jedoch deutlich gestiegen. So verwaltet ein Komfortsteuergerät auch die Diebstahlwarnanlage und wird für das Öffnen und Schließen der Fenster bzw. des Schiebedaches eingesetzt. Ebenso ist es für die Ansteuerung der Innenbeleuchtung und der Ausstiegswarnleuchten verantwortlich. Dabei ist auch grundsätzlich eine Kombination der verschiedenen Funktionalität möglich. Zur Darstellung, des eben beschriebenen Konzeptes, werden die fünf nachfolgenden Anforderungen verwendet:

1. Das Komfortsteuergerät regelt das Innenlicht beim Öffnen bzw. Schließen der Türen.

2. Nach Öffnen einer Tür muss das Innenraumlicht innerhalb von 50 ms hoch gedimmt werden und für 5 Sekunden leuchten.
3. Nach dem Schließen aller Türen wird das Innenraumlicht nach 2 Sekunden ausgeschaltet.
4. Das Innenraumlicht bleibt höchstens 10 Sekunden an.
5. Falls beim Öffnen einer Tür das Innenraumlicht bereits brennt, so wird die Dauer um 5 Sekunden verlängert.

Diese Anforderungen müssen in die erweiterte Systemmodellierung übernommen werden. Die Anforderung eins wird in der SysML durch einen Block namens „Komfortsteuergerät“ realisiert. Die Abbildung 3 zeigt das aus den Anforderungen 2 - 5 modellierte Real-Time Statechart, welches das Verhalten des Blocks spezifiziert. Es besteht aus zwei Zuständen und vier Transitionen. Die zeitlichen Informationen aus den Anforderungen lassen sich in den Uhren bzw. der Invariante wiederfinden. So findet man beispielsweise die Anforderung 4 in der Zeitinvariante  $Licht\_gesamt \leq 10000$  wieder. Während die Anforderung 5 durch die Transition „Tür geöffnet“ beim Zustand „Innenraumlicht\_an“ repräsentiert wird. Die Anforderung 2 und 3 werden durch die Zeitguards bzw. die Perioden  $[0;50]$  bzw.  $[2000;2005]$  umgesetzt. Der Dimm Vorgang wird dabei innerhalb der Funktion „aktiviereLicht“ behandelt. Bei der letzten Periode ist ein Zeitpuffer von 5ms hinzugefügt worden.

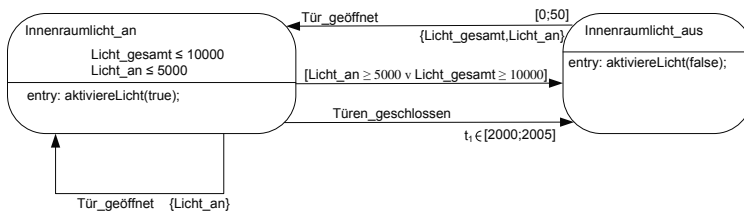


Abbildung 3: Real-Time Statechart der Anforderungen an das Komfortsteuergerät

Das formal spezifizierte zeitliche Verhalten kann nun weiter verwendet werden. So kann mittels formaler Verifikationstechniken (Model Checking) überprüft werden, ob das Komfortsteuergerät die Sicherheitseigenschaften erfüllt, z. B. das keine Verklemmung (Deadlock) vorliegt [Gie05]. Dabei kann nicht nur das Verhalten der Komponente an sich, sondern auch die Kommunikation zwischen ihnen mittels der Real-Time Statecharts beschrieben werden [BGS05]. Hierdurch können die zeitlichen Anforderungen überall formal spezifiziert werden. Ebenso können die Daten auch genutzt werden, um sie innerhalb einer Simulationsumgebung weiter zu analysieren. Hierdurch lassen sich unter anderem Zeitüberschreitungen feststellen. Dafür können beispielsweise Werkzeuge wie ChronSim<sup>3</sup> oder SymTA/S<sup>4</sup> [HHJ<sup>+</sup>05] eingesetzt werden, um die verschiedenen Zeitanforderungen und ihre Aufteilung zu analysieren und zu überprüfen. Die Analyseergebnisse dienen als

<sup>3</sup>www.inchron.de

<sup>4</sup>www.symtavision.com

Grundlage für spätere Designentscheidungen. So kann bei dem eben vorgestellten Komfortsteuergerät beispielsweise die CPU-Last analysiert werden. Wenn sie zu hoch ist, muss ggf. eine leistungsstärkere Hardware genutzt werden. Ebenso kann durch die Analyseergebnisse entschieden werden, ob die Funktion zur Steuerung des Innenlichts auf mehrere Rechenkerne verteilt werden kann. Die Ergebnisse ergänzen somit das Erfahrungswissen der Entwickler, was bisher ausschließlich die Grundlage für Designentscheidungen war. Die Erweiterung ist exemplarisch umgesetzt und wird derzeit im Rahmen des SPES2020 Projektes weiter evaluiert [SPE09].

## 5 Resümee und Ausblick

Bei eingebetteten Systemen und vor allem bei Systemen im Automobilbereich hat die Einhaltung von zeitlichen Anforderungen eine wesentliche Bedeutung. Aufgrund der steigenden Funktionalität und der damit einhergehenden steigenden Komplexität muss bereits in den frühen Entwicklungsphasen dafür Sorge getragen werden, dass diese Anforderung beachtet und erfüllt werden. Dies wird zukünftig immer schwieriger, wenn nicht geeignete Modellierungsmöglichkeiten zur Verfügung stehen, um diese Komplexität beherrschbar zu gestalten.

Mit dem hier vorgestellten Ansatz einer Systemmodellierung mit SysML - erweitert durch Real-Time Statecharts - ist es nun möglich, das System- und Kommunikationsverhalten inklusive zeitlicher Bedingungen formal zu spezifizieren. Hierdurch bekommt der Entwickler sowohl einen besseren Überblick als auch die Möglichkeit sein Modell zu analysieren und zu überprüfen. Dies kann sowohl durch Simulationen als auch durch Model Checking erfolgen. Das Model Checking kann auch bei größeren Systemen durchgeführt werden, da die Real-Time Statecharts eine Dekomposition ermöglichen. Hierdurch kann bereits in frühen Entwicklungsphasen das Modell verifiziert werden. Fehler und Intoleranzen werden frühzeitig erkannt und treten nicht erst im Integrationstest auf. Dementsprechend können sie früh und ohne großen Ressourcenaufwand behoben werden. Dies ist eine deutliche Verbesserung der jetzigen Entwicklung und in der Zukunft auch notwendig.

Die Anbindung des hier vorgestellten Ansatzes an einen Model Checker ist bereits möglich [GBS04]. Eine vollständig automatisierte Anbindung an eine Simulationsumgebung ist jedoch noch nicht gegeben. Es existieren bereits erste Ansätze und Prototypen, die noch ergänzt werden müssen. Hieran wird aber im SPES2020 Projekt [SPE09] weiter geforscht. Denn ein Großteil der Informationen, die für eine Simulation notwendig ist, ist bereits im erweiterten Systemmodell vorhanden. Somit ist eine passende Abbildung zu definieren, um eine semi-automatische Wiederverwendung zu ermöglichen. Dieser Automatisierungsschritt bedeutet, dass auf der einen Seite eine schnellere Anbindung an die Simulationsumgebung ohne redundante Datenbestände notwendig ist. Auf der anderen Seite wird so auch eine Fehlerquelle, nämlich das manuelle Übertragen von Informationen, ausgeschaltet.

## Literatur

- [BGS05] Sven Burmester, Holger Giese und Wilhelm Schäfer. Model-Driven Architecture for Hard Real-Time Systems: From Platform Independent Models to Code. *Model Driven Architecture Foundations and Applications ECMDA-FA*, 2005.
- [Bro06] Manfred Broy. Challenges in Automotive Software Engineering. *International Conference on Software Engineering (ICSE)*, 2006.
- [Ead08] EAST-ADL2 Specification, 2008.
- [GB03] Holger Giese und Sven Burmester. Real-Time Statechart Semantics. Bericht, Universität Paderborn, 2003.
- [GBS04] Holger Giese, Sven Burmester und Wilhelm Schäfer. Modular Design and Verification of Component-Based Mechatronic Systems with Online-Reconfiguration. *Foundations of Software Engineering (FSE)*, 2004.
- [GH06] Holger Giese und Stefan Henkler. A survey of approaches for the visual model-driven development of next generation software-intensive systems. *Journal of Visual Languages & Computing*, 2006.
- [Gie05] Holger Giese. Modeling and Verification of Cooperative Self-adaptive Mechatronic Systems. *Monterey Workshop*, 2005.
- [Gri03] Klaus Grimm. Software Technology in an Automotive Company - Major Challenges. *25th International Conference on Software Engineering*, 2003.
- [GTB<sup>+</sup>03] Holger Giese, Matthias Tichy, Sven Burmester, Wilhelm Schäfer und Stephan Flake. Towards the Compositional Verification of Real-Time UML Designs. *Proceedings of the 9th European software engineering conference*, 2003.
- [HH07] Matthias Hagner und Michaela Huhn. Modellierung und Analyse von Zeitanforderungen basierend auf der UML. *GI-Workshop: Automotive Software Engineering*, 2007.
- [HHJ<sup>+</sup>05] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter und Rolf Ernst. System Level Performance Analysis - the SymTA/S Approach. *IEEE Proceedings of Computers and Digital Techniques*, 152, 2005.
- [HKM07] Martin Hobelsberger, Stefan Kuntz und Jürgen Mottok. Architekturmodellierung: Vergleich von EAST-ADL und SAE AADL. *Automotive, Hanser Verlag*, 2007.
- [MH09] Jan Meyer und Jörg Holtmann. SPES Deliverable AU-D.3.3.A, 2009.
- [OMG07] OMG. UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE), 2007.
- [SFB09] Sonderforschungsbereich 614 „Selbstoptimierende Systeme des Maschinenbaus“, [www.sfb614.de](http://www.sfb614.de), 2009.
- [SPE09] SPES. Software Plattform Embedded Systems (SPES2020), [www.spes2020.de](http://www.spes2020.de), 2009.
- [Spt05] UML Profile for Schedulability, Performance and Testing (SPT), 2005.
- [Sys08] SysML Specification 1.1, 2008.
- [Wei08] Tim Weilkiens. *Systems Engineering mit SysML/UML*. dpunkt-Verlag, 2008.
- [WW03] Matthias Weber und Joachim Weisbrod. Requirements Engineering in Automotive Development: Experiences and Challenges. *IEEE Software*, 20:16 – 24, 2003.

SE | 10  
SOFTWARE ENGINEERING

**Grid Workflow Workshop  
(GWW)**





# 3. Grid Workflow Workshop (GWW 2010)

Wilhelm Hasselbring<sup>1</sup>, André Brinkmann<sup>2</sup>

<sup>1</sup> Universität zu Kiel, wha@informatik.uni-kiel.de

<sup>2</sup> Universität Paderborn, brinkman@uni-paderborn.de

## 1 Einleitung

Im Bereich des Grid Computing stehen Workflows im Fokus zahlreicher Projekte. Allein auf europäischer Ebene wurden und werden viele Projekte zur Entwicklung von geeigneten Werkzeugen, Sprachen und Laufzeitumgebungen für Workflows im Grid- Computing gefördert.

Der dritte Grid Workflow Workshop adressiert sowohl wissenschaftliche als auch betriebliche Workflows im Umfeld des Grid-Computing. Diese dritte Auflage legt den Schwerpunkt auf Fragen des Software Engineering für betriebliche und wissenschaftliche Workflows, Workflow- Sicherheitsinfrastrukturen und die Integration/Migration bestehender betrieblicher und Grid-spezifischer Infrastrukturen.

## 2 Themenfelder

Die Themenfelder der dritten Auflage des Grid Workflow Workshops sind:

- Kommerzielle und wissenschaftliche Grid Workflows
- Ausführungsumgebungen für Grid Workflows
- Grid Workflow-Sicherheit
- Workflows und Grid-Middleware
- EAI und Grid-Computing
- SOA mittels Grid-Technologien
- Grid Service Orchestrierung
- Scheduling und Workflows
- Workflow-Sprachen für Grid Workflows
- Domänenorientierte Grid Workflow-Definition
- Formale Modelle für die Grid Workflow-Analyse

### 3 Beiträge

Die folgenden Beiträge wurden auf dem Workshop präsentiert:

- Andreas Hoheisel, Michael Fellmann and Thorsten Dollmann:  
Dynamische Prozesse und deren Abbildung auf ausführbare Workflows description
- Dirk Mühlenberg and Sandro Leuchter  
Semantisch beschriebene Grid-Dienste in einem Semantik Webservice Framework
- Georg Birkenheuer, Sebastian Breuers, André Brinkmann, Dirk Blunk, Gregor Fels, Sandra Gesing, Sonja Herres-Pawlis, Oliver Kohlbacher, Jens Krüger and Lars Packschies:  
Grid-Workflows in Molecular Science
- Stefan Gudenkauf, Guido Scherp, André Höhing, Wilhelm Hasselbring and Odej Kao:  
Workflow Modeling for WS-BPEL-based Service Orchestration in SMEs
- Ernst Juhnke, Tim Dörnemann, Roland Schwarzkopf and Bernd Freisleben:  
Security, Fault Tolerance and Modeling of Grid Workflows in BPEL4Grid
- Guido Scherp and Wilhelm Hasselbring:  
Ein modellgetriebener Ansatz zur Nutzung von WS-BPEL für Scientific Workflows
- Mirko Sonntag, Dimka Karastoyanova and Frank Leymann: The Missing Features of Workflow Systems for Scientific Computations

In der Abschlussdiskussion wurden insbesondere die (tatsächlichen) Anforderungen an Scientific Workflows diskutiert.

# Dynamische Prozesse und deren Abbildung auf ausführbare Workflows

Andreas Hoheisel<sup>1</sup>, Michael Fellmann<sup>2</sup>, Thorsten Dollmann<sup>3</sup>,

<sup>1</sup> Fraunhofer-Institut für Rechnerarchitektur und Softwaretechnik (FIRST)  
Kekuléstraße 7, 12489 Berlin  
andreas.hoheisel@first.fraunhofer.de

<sup>2</sup> Universität Osnabrück  
Institut für Informationsmanagement und Unternehmensführung (IMU)  
Lehrstuhl für Informationsmanagement und Wirtschaftsinformatik  
Katharinenstraße 3, 49069 Osnabrück  
michael.fellmann@uos.de

<sup>3</sup> Institut für Wirtschaftsinformatik (IWi) im DFKI  
Stuhlsatzenhausweg 3, 66123 Saarbrücken  
thorsten.dollmann@iwi.dfki.de

**Zusammenfassung:** Der Beitrag beschreibt Überlegungen und neue Ansätze zur Unterstützung der Dynamik von Prozessen in IT-gestützten Geschäftsprozessmanagementsystemen für SOA, Grid- und Cloud-Umgebungen. Hierbei werden häufige Änderungen von Prozessen sowohl auf der fachlichen Ebene als auch auf der Ebene der Implementierung und Ausführung berücksichtigt. Der dargestellte Ansatz basiert auf einer automatischen Transformation von fachlichen in ausführbare Prozessmodelle sowie deren Ausführung und Beobachtung mit dem *Grid Workflow Execution Service* (GWES).

## 1 Einleitung

Aktuelle Systeme zum IT-gestützten Geschäftsprozessmanagement (*Business Process Management – BPM*) gliedern den Ablauf des Prozessmanagements in der Regel in drei getrennte, aufeinanderfolgende Phasen: Prozessmodellierung, Prozessautomatisierung und Prozessausführung [Pe09]. In der Prozessmodellierung werden fachliche Modelle der Geschäftsprozesse erstellt und analysiert. In der Prozessautomatisierung werden aus fachlichen Modellen technische Prozesse abgeleitet, welche als Grundlage für die IT-gestützte Automatisierung dienen. Der Begriff „Automatisierung“ bezieht sich hierbei auf das Prozessmanagement; ein Prozess gilt somit auch dann als automatisiert, wenn einzelne Aktivitäten durch Personen ausgeführt werden. In der Prozessausführung schließlich erfolgt die Ausführung der Prozesse, zum Beispiel in einer SOA, Grid- oder Cloud-Umgebung. Für die drei Phasen der Modellierung, Automatisierung und Ausführung kommen meist jeweils eigenständige Methoden und Werkzeuge zum Einsatz, welche die Durchlässigkeit zwischen den Phasen behindern. Ein etablierter Ansatz ist dabei

die Verwendung der *Business Process Modeling Notation* (BPML) [Ob09] zur Prozessmodellierung und die anschließende Überführung in die *Web Services Business Process Execution Language* (WS-BPEL) [Oa07] anhand der im Anhang A des BPMN-Standards enthaltenen Transformationsregeln. Allerdings bringt dieser Ansatz im Kontext dynamischer und verteilter Prozesse zwei grundlegende Probleme mit sich. Zum einen gestaltet sich eine Transformation von BPMN-Modellen in WS-BPEL-Skripte aufgrund der umfangreichen BPMN-Konstrukte komplex und fehleranfällig. Zum anderen wird eine Virtualisierung bei der Prozessautomatisierung im Kontext von WS-BPEL nicht explizit berücksichtigt.

Der vorliegende Artikel beschreibt ein Vorgehen, das die Dynamik von Prozessen durch eine engere Verzahnung der drei Phasen des Geschäftsprozessmanagements besser unterstützen soll. Basis hierfür ist die Verwendung eines über weite Teile einheitlichen Prozessbeschreibungsformalismus, der eine einfache Abbildung zwischen den Phasen ermöglicht und somit eine direkte Interaktion zwischen Modellierung, Automatisierung und Ausführung zulässt. Dadurch soll eine bessere Abstimmung zwischen fachlichen und technischen Prozessen realisiert werden, was insbesondere für dynamische Prozesse zweckdienlich ist. Ad-hoc-Änderungen in fachlichen Prozessen (z. B. „Unterauftragnehmer insolvent“) können dadurch einfacher an die technische Ebene durchgereicht werden. Umgekehrt ist eine schnelle Reaktion auf Änderungen in der technischen Ebene (z. B. „Datenspeicher voll“) eine wichtige Voraussetzung für die effektive Anwendbarkeit von Geschäftsprozessmanagement.

Dieser Artikel ist eine Weiterführung der Arbeiten aus [Ho09], in welchen der Fokus auf den technischen Aspekten der Abbildungsmethodik zur Überführung von ereignisgesteuerten Prozessketten in Petrinetze lag. Der Schwerpunkt des vorliegenden Artikels liegt darüber hinaus in allgemeinen Überlegungen zur Unterstützung der Dynamik von Prozessen. Diese Arbeit wurde gefördert durch die BMBF-Projekte der D-Grid-Initiative BauVOGrid (Förderkennzeichen: 01IG07001) und PneumoGRID (Förderkennzeichen: 01IG09002D).

## **2 Dynamische Prozesse und Ressourcen**

Die Dynamik von Prozessen und Ressourcen lässt sich anhand der Variabilität von Umwelt und Prozess in vier Kategorien unterteilen:

1. Konstanter Prozess in konstanter Umwelt. Ein Beispiel hierfür sind Unternehmen, die in stabilen Märkten mit einem konstanten Prozess immer gleichartige Leistungen anbieten.
2. Konstanter Prozess in dynamischer Umwelt. Ein Beispiel hierfür ist ein Industrieunternehmen, dessen Fertigungsprozess tolerant ausgelegt sein muss gegenüber dem Ausfall einer Ressource wie einer Maschine oder eines IT-Dienstes.
3. Dynamischer Prozess in konstanter Umwelt. Ein Beispiel hierfür ist ein Unternehmen, dessen Produkte aufgrund einer Unternehmensübernahme ggf. mit anderen (zeitlich konstanten) Ressourcen erstellt werden müssen.

4. Dynamischer Prozess in dynamischer Umwelt. Ein Beispiel hierfür ist ein Unternehmen, dessen Geschäftsprozess ständig an sich ändernde Märkte und Ressourcen angepasst werden muss.

Die für unsere Methodik relevanten Anwendungsszenarien stammen aus den Projekten BauVOGrid und PneumoGRID, in denen sowohl variable wissenschaftliche als auch kommerziell nutzbare IT-Prozesse auf unregelmäßig verfügbaren und fehleranfälligen Grid-Knoten ausgeführt werden sollen. Schwerpunkt unseres Ansatzes ist daher die Berücksichtigung von Prozess- und Ressourcendynamik gemäß Kategorie 4. Bezüglich der drei Phasen Modellierung, Automatisierung und Beobachtung ergeben sich dabei folgende Lösungsansätze:

*Phase 1 – Prozessmodellierung:* Die Auswirkung von Änderungen in fachlichen Prozessen werden direkt während der Modellierung analysiert und simuliert. Falls umgekehrt ein fachlicher Prozess auf Grund der variablen Umwelt modifiziert werden muss – z. B. wegen eines Ressourcenengpasses – so werden die hierfür notwendigen Informationen von der technischen Ebene zurück auf die Ebene der fachlichen Prozessmodellierung abgebildet und dienen dort als Entscheidungshilfe.

*Phase 2 – Prozessautomatisierung:* Änderungen von fachlichen Prozessmodellen werden handhabbar durch eine möglichst automatische Abbildung auf technische, ausführbare Prozessmodelle sowie eine späte Bindung der Fachfunktionalitäten an Ausführungskomponenten. Durch Einführung einer Virtualisierungsschicht, die es ermöglicht, ausführbare Prozesse unabhängig von der verfügbaren Infrastruktur zu formulieren, wird zusätzliche Unabhängigkeit gegenüber einer variablen Umwelt gewonnen.

*Phase 3 – Prozessausführung und -beobachtung:* Variable Ressourcen und Komponenten werden handhabbar durch Mechanismen der Lastverteilung und dynamischen Ressourcenverteilung sowie durch zeitnahe Ressourcenüberwachung und fehlertolerante Ausführung. Unterstützt wird dies durch eine flexible, z. B. Constraint-basierte Ressourcenplanung, die ein dynamisches und automatisches Nachplanen bei Änderungen der Prozesse oder der Umwelt ermöglicht.

Die folgenden Abschnitte stellen unsere bislang realisierten Arbeiten bezüglich Modellierung (Abschnitt 2.1), Automatisierung (Abschnitt 2.2) sowie Ausführung und Beobachtung (Abschnitt 2.3) von dynamischen Prozessen kurz vor.

## 2.1 Modellierung fachlicher und technischer Prozesse

Zur Modellierung **fachlicher Prozesse** existieren im Wesentlichen zwei etablierte Modellierungsnotationen: Die *Business Process Modeling Notation* (BPMN) [Ob09] sowie die *Ereignisgesteuerten Prozessketten* (EPK) [KS92]. Aufgrund der einfacheren Struktur sowie der semantischen Nähe zu Petrinetzen – welche in dieser Arbeit der Prozessautomatisierung zugrunde liegen – basiert die Geschäftsprozessmodellierung in der vorliegenden Arbeit auf EPK. Zum Austausch von Prozessmodellen, die mit Hilfe einer Ereignisgesteuerten Prozesskette beschrieben werden, hat sich die *Event-driven Process Mar-*

*kup Language (EPML)* durchgesetzt [MN04]. Die XML-basierte Notation gestattet eine plattform- und sprachunabhängige Repräsentation von EPK.

Die Modellierung **technischer Prozesse** basiert auf High-Level-Petrinetzen gemäß ISO/IEC 15909-1 [Is04] und dem entsprechenden XML-Austauschformat *GWorkflowDL* [Ho09, Al06, Al06b, HA06], welches neben der Modellierung und Analyse von technischen Prozessen auch eine direkte Automatisierung und Überwachung der Prozesse unterstützt. Für eine formale Spezifikation der *GWorkflowDL* sei auf das XML-Schema der *GWorkflowDL* [Ho09] verwiesen.

## 2.2 Überführung von fachlichen Geschäftsprozessen in ausführbare IT-Prozesse

Abbildung 1 zeigt den gewählten Lösungsansatz für die schrittweise und zum großen Teil automatisierte Abbildung von fachlichen Prozessbeschreibungen auf die jeweils geeigneten und verfügbaren Ressourcen. Die Geschäftsprozesse werden zunächst annotiert und dann auf abstrakte technische Prozesse (gelb) abgebildet, in diesem Fall durch die Abbildung von EPK auf *GWorkflowDL*-Dokumente [Ho09]. Jeder Aktivität werden dann zunächst mit Hilfe des ResourceMatcher-Dienstes Service-Kandidaten (blau) zugeordnet, welche die entsprechende Funktionalität bereitstellen. Ein Scheduler wählt einen der Kandidaten aus (grün) und führt die Aktivität auf den entsprechenden Ressourcen aus. Ein Beispiel einer Überführung eines EPK-Modells in einen ausführbaren *GWorkflowDL*-Prozess ist in Abbildung 2 dargestellt.

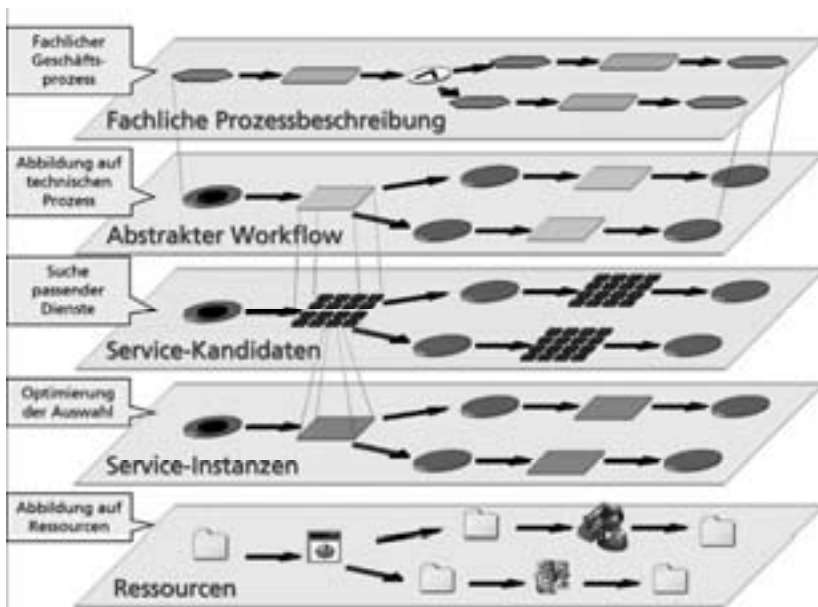


Abbildung 1: Überführung von fachlichen Prozessbeschreibungen in automatisierte Prozesse

### 2.3 Automatisierung und Beobachtung der Prozesse

Der *Grid Workflow Execution Service (GWES)* ermöglicht die Automatisierung und das interaktive Management von komplexen und dynamischen Prozessabläufen in Service-orientierten Architekturen oder Grid-Umgebungen [Ho08, Ho06, GH07, Vo08]. Durch die Marken des Petrinetzes wird nicht nur ein Prozessmuster, sondern auch der Zustand einer jeden Prozessinstanz beschrieben. Dies ermöglicht die direkte Überwachung der Prozesse sowie eine einfache Realisierung von Fehlertoleranzmechanismen.

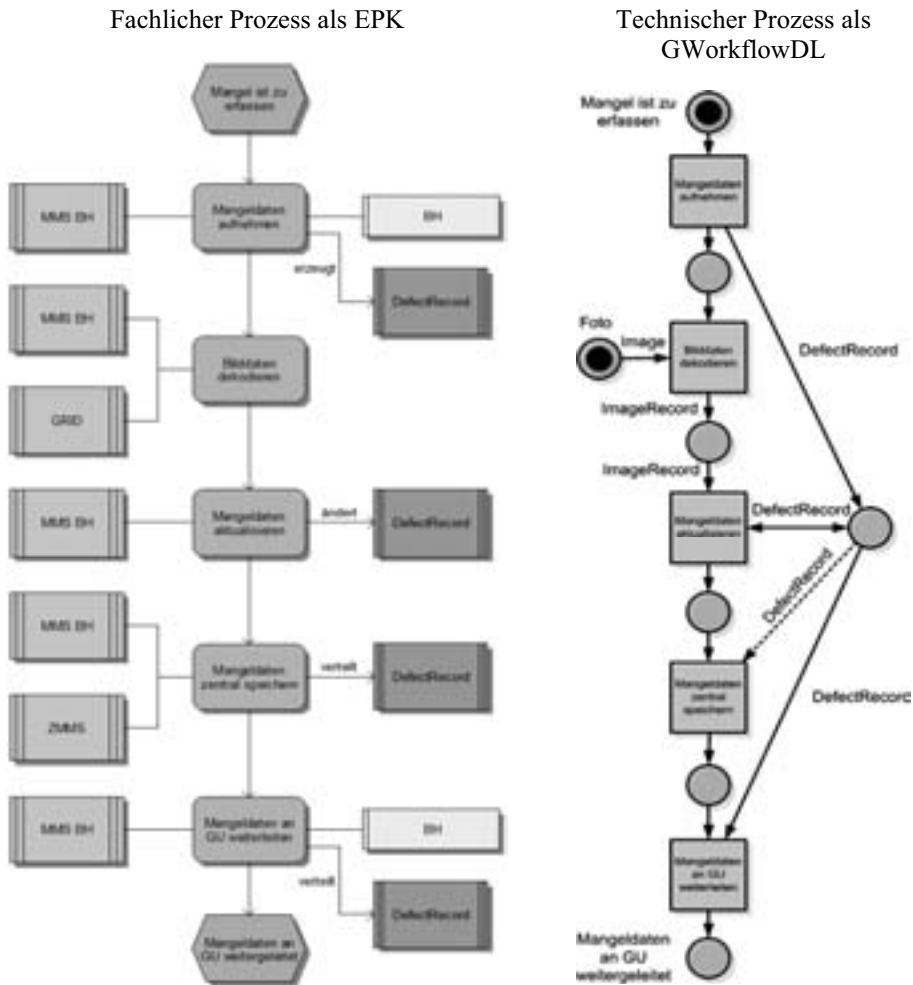


Abbildung 2: Beispiel einer Überführung eines einfachen Geschäftsprozesses aus dem Bereich Mängelmanagement im Bauwesen in einen ausführbaren Grid-Workflow



### 3 Schlussfolgerungen und Ausblick

Der Beitrag hat am Beispiel der Sprachen EPK und GWorkflowDL gezeigt, wie ein durchgängiger Ansatz von der fachlichen Beschreibung von Prozessen bis hin zu deren Ausführung unter Berücksichtigung von Dynamik, Lastverteilung und Ausfallsicherheit erreicht werden kann. Dieser Ansatz ist besonders für sich häufig ändernde Prozesse sowie für verteilte Umgebungen im Bereich des Grid- und Cloud-Computing geeignet, in denen nur partiell eine Kontrolle über die zur Ausführung notwendigen Ressourcen möglich ist. Zukünftiger Forschungsbedarf besteht insbesondere hinsichtlich einer verbesserten Werkzeugunterstützung.

### 4 Literaturverzeichnis

- [Al06] Alt M., Gorlatch S., Hoheisel A., Pohl H.-W.: Using High-Level Petri Nets for Hierarchical Grid Workflows. 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006), Amsterdam, Netherlands, IEEE, 2006.
- [Al06b] Alt M., Hoheisel A., Pohl H.-W., Gorlatch S.: A Grid Workflow Language Using High-Level Petri Nets. PPAM05, LNCS 3911, Springer, 2006; S. 715-722
- [GH07] Gubala T.; Hoheisel A.: Highly Dynamic Workflow Orchestration for Scientific Applications. CoreGRID Technical Report Number TR-0101, CoreGRID, 2007.
- [HA06] Hoheisel A.; Alt M.: Petri Nets. In (Taylor I.J., Gannon D., Deelman E., Shields M.S. Hrsg.): Workflows for e-Science – Scientific Workflows for Grids, Springer, 2006.
- [Ho06] Hoheisel, A.: User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2006.
- [Ho08] Hoheisel, A.: Grid-Workflow-Management. In (Weisbecker, A.; Pfreundt, F.-J.; Linden, J.; Unger, S. Hrsg.): Fraunhofer Enterprise Grids – Software. Fraunhofer IRB Verlag, Stuttgart, 2008. ISBN: 978-3-8167-7804-2
- [Ho09] Hoheisel, A.: XML-Schema der Grid Workflow Description Language (GWorkflowDL) – Version 2.0. Fraunhofer FIRST, 2009; [http://www.gridworkflow.org/kwfggrid/src/xsd/-gworkflowdl\\_2\\_0.xsd](http://www.gridworkflow.org/kwfggrid/src/xsd/-gworkflowdl_2_0.xsd) [Zugriffsdatum 20.09.2009]
- [Ho09b] Hoheisel, A.; Dollmann, T.; Fellmann, M.: Überführung von EPK-Modellen in ausführbare Grid- und Cloud-Prozesse. In: Tagungsband der EPK2009, 2009
- [Is04] ISO/IEC 15909-1: High-level Petri nets – Part 1: Concepts, definitions and graphical notation. 2004.
- [KS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). In: Scheer A-W (Hrsg): Veröffentlichungen des Inst. für Wirtschaftsinformatik, Nr. 89, Universität des Saarlandes. 1992
- [MN04] Mendling, J.; Nüttgens, M.: Exchanging EPC Business Process Models with EPML. In (Nüttgens, M.; Mendling, J. Hrsg.) Proc. of the 1st GI Workshop XML4BPM - XML Interchange Formats for Business Process Management; Modellierung 2004, Marburg Germany, 2004, S. 61-79.
- [Oa07] OASIS: Web Services Business Process Execution Language Version 2.0. OASIS, 2007; <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html> [Zugriffsdatum 20.09.2009]
- [Ob09] Object Management Group: Business Process Modeling Notation (BPMN) – Version 1.2., 2009; <http://www.omg.org/spec/BPMN/1.2/> [Zugriffsdatum 20.09.2009]
- [Pe09] Peisl, R.: Business Process Management mit IBM WebSphere. IBM, 2009.
- [Vo08] Vossberg, M.; Hoheisel, A.; Tolxdorff, T.; Krefting, D.: A Workflow-based Approach for Fault-tolerant Medical Image Transfer in Health Grids. In: Future Generation Computer Systems, Elsevier, 2008.

# Semantisch beschriebene Grid-Dienste in einem Semantik Webservice Framework

Dirk Mühlenberg & Sandro Leuchter

Abt. Interoperabilität und Assistenzsysteme  
Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB  
Fraunhoferstraße 1  
76131 Karlsruhe  
{vorname.nachname}@iosb.fraunhofer.de

**Abstract:** Grid-Middleware erlaubt den verteilten Zugriff auf Software- und Hardwareressourcen, indem sie prinzipiell solche Ressourcen beschreibt und einem Anwender über Protokoll und API auf seinem Arbeitsplatz zur Verfügung stellt. Ressourcen liegen auf entfernten Knoten, ihr Zustand ist jederzeit ermittelbar und Daten werden über definierte Protokolle ausgetauscht. Höherwertige Anwendungen werden durch Aggregation einzelner Ressourcen in Workflows komponiert, die durch entsprechende Werkzeuge vom Anwender modellierbar sind. Im Bereich „Kooperative Informationssysteme“ auf Basis einer SOA benötigt man aber eine Architektur, die es dem Anwender erlaubt, durch Formulierung eines Informationsinteresses einen dynamischen evtl. zyklischen Workflow zu starten, der zum Anfragezeitpunkt noch gar nicht bekannt ist, sondern vom System durch Vergleich der verfügbaren Ressourcen mit den angefragten Informationsobjekten erzeugt wird. Statische Workflowsysteme z.B. auf Basis von WS-BPEL (Web Services Business Process Execution Language) eignen sich nicht für diese Anforderung. Stattdessen verwenden wir einen Ansatz aus dem Bereich semantische Webservices, um zielgetriebene verteilte Aufgabenverarbeitung in einem Anwendungsbereich kooperativer Informationssysteme für die Luftbildauswertung zu realisieren. Jedoch berücksichtigen die Standardframeworks aus diesem Bereich keine Grid-Dienste und die dort eingeführten Formate, eine Integration ist aufgrund derselben technischen Schnittstelle (WSRF – Web Service Resource Framework) aber möglich. Die Hauptaufgabe dafür liegt in der Integration der Beschreibungselemente der Grid-Ressourcen in den Erzeugungsprozess des Workflows als nicht-funktionale Parameter. Die hier vorgestellten Arbeiten sind Bestandteil zu einem möglichen Semantik Grid Ansatz als Basis für eine service-orientierte Architektur.

# 1 Einleitung

Moderne Informationssysteme mit verteilten Ressourcen wie Datenbanken und Fachanwendungen erlauben es heutzutage dem Nutzer in einem Portal transparent auf Daten und Dienste zuzugreifen ohne genaues Wissen über den Ort der Daten oder der Anwendungen voraus zu setzen. Wird eine service-orientierte Architektur (SOA) basierend auf Webservices genutzt, so gibt es unterschiedliche Ansätze, um Ressourcen in einem Netz zu beschreiben. Die Beschreibung von IT-Ressourcen ist gerade im Konzept eines Software-Grids ein entscheidendes Basiselement, Grid-Ressourcen werden anhand wohl definierter Parameter beschrieben und zur Laufzeit durch entsprechende Dienste (sogenannte Information Provider) veröffentlicht und so dauerhaft Konsumenten zur Auswertung zur Verfügung gestellt. Diese Beschreibungen sind jedoch ausgelegt auf die Bedürfnisse von naturwissenschaftlichen wie simulationslastigen Anwendungen, d.h. Parameter zur Beschreibung von Hardware, Betriebssystem und Laufzeitgrößen spielen hier die wesentliche Rolle, um entsprechende Jobs an ausgesuchte Knoten im Grid zu delegieren. Die vorhandenen Beschreibungen können aber prinzipiell um anwendungsnahe Elemente erweitert werden. Die Nutzung von Ontologien ermöglicht die Ressourcen hierfür semantisch zu beschreiben.

## 2 Semantische Beschreibung von Grid-Ressourcen

Ein Framework, das sich zur Ontologie-basierten Dienstbeschreibung anbietet, ist das „Semantic Metadata Discovery and Monitoring System“ (S-MDS), das im Rahmen des AIST-GTRC Database Grids entwickelt wurde [MK06]. Diese Software erlaubt es, in einem Globus Toolkit (GT4) Grid Dienste mit Metadaten in Form von Ontologieinstanzen zu versehen. Die Informationen werden zur Laufzeit über GT4 Standardmechanismen (Monitoring und Discovery Service - MDS) gesammelt und auf diese Ontologie-Instanzen abgebildet. Die Beschreibungen und ihre Zuordnungen können mit Editoren erstellt und im GT4 standardkonform über Information Provider veröffentlicht werden. Eine Untermenge der Metadaten ist dann z.B. über eine SPARQL Anfrage von Konsumenten zu ermitteln. Dieses Konzept ist für eine Integration in eine semantische Grid-basierte SOA besonders geeignet, da es konform ist mit den GT Mechanismen und eine flexible und gut erweiterbare Basis für die Modellierung von Dienstbeschreibungen darstellt.

### 2.1 Dienstbeschreibung mittels Ontologien als nicht-funktionale Parameter

Die Anforderungen an einen Fachdienst zerfallen in funktionale und nicht-funktionale Anteile. Der funktionale Anteil beschreibt die benötigten und die zurückgelieferten Daten sowie Seiteneffekte und werden in einer technischen Schnittstelle realisiert. Der nicht-funktionale Anteil ist in seiner Beschreibung davon unabhängig und spiegelt die Eigenschaften eines Dienstes wider, die es z.B. erlauben, einen Dienst aufzufinden oder Kosten, Qualität und seinen Nutzen zu bewerten. Solche Beschreibungen können unabhängig von einer Ausführung des Dienstes sein (wie etwa die Beschreibung der Fähigkeiten eines Detektionsverfahrens) oder aber von der Ausführung abhängen, wie

etwa Payload-abhängige Tarife. Erst die nicht-funktionalen Parameter ermöglichen die korrekte und effiziente Nutzung von Diensten, etwa gesteuert durch Präferenzen in einem Nutzerprofil über eine aufgabenangepasste Kosten-Nutzen-Rechnung oder aber die Qualitätsbewertung eines Dienstes durch Einzel- oder Gemeinschaftsbewertung der Nutzer im Netz. Die Repräsentation der Parameter als Wissensbasis erlaubt zudem regelbasiert zu arbeiten und über die verschiedenen Informationen Schlussfolgerungen zu ziehen.

Als Basisontologie muss im Sinne der Kompatibilität im GT4 Umfeld ein Modell basierend auf GLUE-CE (Grid Laboratory for a Uniform Environment) [An09] gewählt werden. „CE“ steht für Computing Element ein Basiskonzept für eine Entität, die eine Grid-Ressource managt. Diese Basis wurde im Rahmen eines Experimentalsystems für die Anwendung Luftbildauswertung erweitert um Anteile, die auf den Grundkonzepten „anwendungsrelevantes Objekt“ (ARO), „Information“, „Dienst“, „Experte“ und „Nutzereigenschaften“ basieren. Die Fähigkeiten eines Experten oder Dienstes wurden aus Netzen von AROs modelliert, die Güte oder Qualität aus dem Übereinstimmungsgrad der Fähigkeit mit der geforderten Fähigkeit aus der Nutzeranfrage ableiten. Der funktionale Anteil der Schnittstelle wird im Webservice/Grid Umfeld in einer WSDL (Web Service Description Language) Datei gemäß dem zugehörigen W3C Standard beschrieben, der nichtfunktionale Anteil wird im nächsten Abschnitt ausgeführt.

## **2.2 Umsetzung in S-MDS**

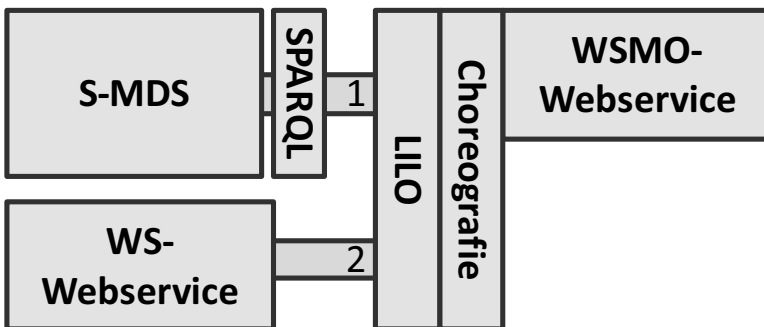
S-MDS erlaubt es, Standard Grid Dienste mittels einer ontologiebasierten Beschreibung (Metadaten) zu hinterlegen. Die Ontologie ist in OWL (Web Ontology Language) zu formulieren. Eine Ausgangsform basierend auf GLUE-CE ist in OWL verfügbar, die mit einem Ontologieeditor wie Protégé zu erweitern ist. Die eigentlichen Laufzeitinformationen der nicht-funktionalen Eigenschaften werden durch einen Grid Resource Provider gestellt, der je nach Anwendung (also Satz von Dienstparametern) zu implementieren ist. Für die Standard Grid-Parameter gibt es bereits vorgefertigte Klassen. Hat man die anwendungsspezifische Ontologie erstellt, so lassen sich anhand eines festgelegten Ablaufs in einem Editor die für einen Dienst wichtigen Konzepte mit ihren Werten verbinden. Die derzeitige Implementierung von S-MDS lässt als Zuordnung von Parameter und Wert derzeit nur XPath (XML Path Language) Ausdrücke [Be07] zu. Die Werte müssen dabei einer XML Datei entstammen, die vom eigentlichen Wertelieferant (Resource Provider) zu befüllen ist. Hervorzuheben ist die Trennung von Metadatenlieferant und der eigentlichen Darstellung der Metadaten als Ontologieinstanzen innerhalb von S-MDS. Somit ist es prinzipiell dem Anbieter eines Dienstes im Grid überlassen, wie er seine Metadaten Grid-konform präsentiert, S-MDS sorgt für die Übersetzung in eine Semantik Web konforme Darstellung und die zugehörige Zugriffsmöglichkeit.

### 3 Zielgetriebene Problemlösung für Web Services

Als Middleware für eine zielgetriebene Problemlösung wurde das WSMO (Web Service Modelling Ontology) Framework [Ro06] gewählt, da es basierend auf Standards im Bereich Semantik Web sowie W3C Webservices entwickelt wurde und durch seinen flexiblen modularen Aufbau eine leichte Erweiterungsmöglichkeit für zusätzliche Dienstschnittstellen und -protokolle bietet. WSMO erlaubt es, Webservices mit OWL-konformen Ontologien zu beschreiben. Das Framework besteht aus der Laufzeitumgebung WSMX (Web Service Modelling eXecution environment), auf der Webservices, die mittels WSMML dargestellt sind, ausgeführt werden. Es gibt zwei Werkzeuge zur Modellierung: „Studio“ und „Toolkit“. Mit beiden können Dienste, Ontologien und Workflows entwickelt, getestet und in Repositories veröffentlicht werden. Webservices in WSMO können anhand ihrer Beschreibungen gesucht und für eine bestimmte Aufgabe ausgewählt werden (Matchmaking). Die eigentliche Stärke von WSMO liegt aber in der Art und Weise, wie Workflows ausgeführt werden können. Zum einen gibt es die Möglichkeit, Workflows zu designen oder auch aus anderen Darstellungsformen (z.B. EPML – Event-driven Process Chain Model Markup Language in ARIS) zu importieren, zum anderen aber kann man ein Ziel angeben, zu dem es keinen Workflow gibt. Die Laufzeitumgebung versucht anhand der zur Verfügung stehenden Ressourcen (Dienste und Abbildungsvorschriften), dieses Ziel zu erreichen. Hierbei müssen evtl. komplexe Aufgaben erledigt werden. Beispiele sind das Mapping von unterschiedlichen Ontologien aufeinander, das Auswerten komplexer Choreografien oder das Mapping unterschiedlicher Protokolle (durch Prozessabbildung innerhalb der Choreografie) aufeinander. Ein Ziel (Goal) kann einfach die Erzeugung einer Ontologie-Instanz (im einfachsten Fall über den Aufruf eines Dienstes) in einer Wissensbasis oder die Auswertung eines komplizierten logischen Ausdrucks sein.

### 3.1 Integration der Grid Dienste in WSMO

Ein Webservice in WSMO ist eine Abstraktion von der eigentlichen darunterliegenden Implementierung, die z.B. als W3C Web Service vorliegt. Die Servicebeschreibung wird in WSML [Ro06] formuliert, zur Ausführung des Services muss der abstrakt beschriebene Dienst und seine Eingangsdaten einem Grounding unterzogen werden, d.h. die funktionalen und nicht-funktionalen Eigenschaften, die als Ontologieinstanzen in WSMO vorliegen, werden durch ein *Lowering*<sup>1</sup> in XML Instanzen gemäß der Spezifikation unterzogen. Resultate werden durch ein entsprechendes *Lifting* in Ontologieinstanzen gewandelt. Dieser Prozess ist abgetrennt in einem eigenen Modul verfügbar und für jeden Service zu implementieren. Das *Lowering* muss ausprogrammiert werden, das *Lifting* lässt sich durch ein XSL Transformation darstellen. Ein Grid-Dienst ist ebenfalls ein Webservice, das WSRF basierte Interface wird jedoch über eine eigene Grid-konforme Schnittstelle innerhalb der *Lowering-Lifting*-Komponente (LILO) angesprochen, da z.B. im Grid die Sicherheitsanforderungen voraussetzen, dass der Anfragende authentifiziert ist und eine Authorisierung für die angefragte Ressource besitzt.



Abbildungung 1: Integration von S-MDS in WSMO über Choreografie

<sup>1</sup> Lowering ist der Übergang (*Grounding*) vom Domänenmodell (in WSML) nach SOAP (in XML), Lifting der umgekehrte Vorgang.

### 3.2 Nutzung der Dienstparameter in WSMO

Die Dienstparameter (Metadaten) können auf der Gridseite über einen eigenen Metadatendienst angefragt werden, der sich auf den S-MDS Services abstützt. Der Konsument muss den interessierenden Dienst benennen und erhält die gewünschten Metadaten zurück. Dies ist auf der WSMO Seite über einen eigenen Webservice realisiert, der schließlich den Metadaten Grid Dienst aufruft. Somit lässt sich in der Choreografie von WSMO Services ein Ziel angeben, das die Metadaten zu dem eigentlichen Grid Dienst bereitstellt (Schritt 1 in Abbildung 1 **Fehler! Verweisquelle konnte nicht gefunden werden.**) und mittels Schlussfolgerung oder anderer Mechanismen das Auffinden, die Choreografie oder Orchestrierung von Diensten innerhalb von WSMO beeinflusst (Schritt 2 in Abbildung 1). Insbesondere bei Trennung von in funktionaler Hinsicht gleichartiger Dienste durch nicht-funktionale Parameter wie Fähigkeitsbeschreibungen (z.B. bei Verfahren zur Objektidentifikation) liegt der Wert der Repräsentation eines solchen Parameters durch Ontologien in der Mächtigkeit der Beschreibung und Möglichkeit zur logischen Verarbeitung der Wissensbausteine bzgl. der Anforderungen an diesen Parameter (wie Detektion von Fahrzeugen im Gegensatz zu Detektion von Personen) durch die Nutzeranfrage.

## 4 Verwandte Arbeiten

In [Pa06] ist eine Übersicht über Semantik Grid Ökosysteme. Die Autoren bemühen ein Softwaremuster der ‚Cooperative Semantic Aware and Semantic Provisioning Services‘ sowie den Methodenansatz der evolutionären bzw. der kompletten Neuentwicklung. In dieser Arbeit werden die gewählten Ansätze einer Analyse bzgl. eines umfangreichen Kriterienkatalogs unterzogen, sowie in einem weiteren Kapitel näher beschrieben.

Ein Ansatz, der von den gewählten Frameworks (WSMO + Grid) dem hier vorgestellten am nächsten kommt, ist in [Sh09] zu finden. Das Konzept und der Anspruch sind jedoch verschieden, der Autor erweitert WSMO (und somit WSML) um Konzepte für Grid-Services (sind WSMO Web-Services) und Jobs (sind WSMO Goals) und erläutert anhand eines Testfalls die Interaktion von WSMO mit dem darunterliegenden Grid-Framework. Der Schwerpunkt in [Sh09] liegt in der Spracherweiterung und den dafür notwendigen Erweiterungen der Laufzeitumgebung WSMX und nicht in der semantischen Annotation der Ressourcen und der Nutzung dieser Metadaten in WSMO. Außerdem war es wichtig, die verwendeten Frameworks unverändert zu lassen, um neue Versionen leichter nutzen zu können.

## 5 Zusammenfassung und Ausblick

Im Rahmen des hier beschriebenen experimentellen kooperativen Informationssystems zur Unterstützung der Luftbildauswertung wurden Beiträgen zu einem Semantik Grid Ansatz gezeigt, mit denen anwendungsnahe Metadaten von Grid Diensten bei zielgetriebenen Problemlösungen einbezogen werden können. Ein Schwerpunkt der

Arbeiten lag in der Wiederverwendung bestehender Konzepte zu kooperativen Informationssystemen in einer neuen Architektur, die auf Standards in den Bereichen Semantik Web und Software-Grid beruht. Ebenso sollten die verwendeten Frameworks keinerlei Veränderungen und Einschränkungen unterzogen werden, um Migration und vollen Funktionsumfang zu garantieren. Eine Besonderheit der hier beschriebenen Integration ist, dass Grid Dienste und Semantic Web Services gleichermaßen genutzt werden können.

Zukünftig steht die Integration dieser Anteile in den Semantic Service Bus [Ka07] an, wodurch sogar die Einbindung weiterer Protokolle und Programmierschnittstellen ermöglicht wird.

## 6 Literaturverzeichnis

- [An09] Sergio Androozzi et.al: Glue specification v. 2.0. Technical report, Open Grid Forum, March 2009.
- [Be07] Anders Berglund et.al: XML Path Language (XPath) Version 2.0, W3C Recommendation, January 2007.
- [Ka07] Dimka Karastoyanova et.al: Semantic Service Bus: Architecture and Implementation of a Next Generation Middleware. ICDE Workshops 2007: 347-354.
- [MK06] S.Mirza and I.Kojima, "S-MDS: A Semantic Information Service for Advanced Resource Discovery and Monitoring in WS-Resource Framework", GGF16 3rd Semantic Grid Workshop, Feb., 2006.
- [Pa06] Pinar Alper et al: Semantic Aware Grid Services, Deliverable D1.4, OntoGrid project, August 2006.
- [Ro06] D. Roman et.al: WWW: WSMO, WSML, and WSMX in a nutshell. In Proceedings of the first Asian Semantic Web Conference (ASWC 2006), Beijing, China, September 3-7, 2006.
- [Sh09] M. Omair Shafiq: Semantic Grid: Extending Semantic Web, Master Thesis in Computer Science, March 2009, University of Innsbruck.





# Grid-Workflows in Molecular Science \*

Georg Birkenheuer<sup>1</sup>, Sebastian Breuers<sup>2</sup>, André Brinkmann<sup>3</sup>, Dirk Blunk<sup>4</sup>,  
Gregor Fels<sup>5</sup>, Sandra Gesing<sup>6</sup>, Sonja Herres-Pawlis<sup>7</sup>,  
Oliver Kohlbacher<sup>8</sup>, Jens Krüger<sup>9</sup>, and Lars Packschies<sup>10</sup>

*Paderborn Center for Parallel Computing, Universität Paderborn*

<sup>1</sup>birke@uni-paderborn.de   <sup>3</sup>andre.brinkmann@uni-paderborn.de

*Department für Chemie, Universität zu Köln*

<sup>2</sup>breuerss@uni-koeln.de   <sup>4</sup>d.blunk@uni-koeln.de

*Department Chemie, Universität Paderborn*

<sup>5</sup>fels@uni-paderborn.de   <sup>9</sup>mercutio@uni-paderborn.de

*Zentrum für Bioinformatik, Eberhard-Karls-Universität Tübingen*

<sup>6</sup>sandra.gesing@uni-tuebingen.de

<sup>8</sup>oliver.kohlbacher@uni-tuebingen.de

*Fakultät Chemie, TU Dortmund*

<sup>7</sup>sonja.herres-pawlis@tu-dortmund.de

*Regionales Rechenzentrum, Universität zu Köln*

<sup>10</sup>packschies@uni-koeln.de

**Abstract:** Computational Chemistry gathers information about properties of molecules based on compute intensive simulations. In this area, workflows are an essential instrument for managing complex simulations cascades. The aim of the MoSGrid project is an easy to use Grid integration of such workflows based on a portal that covers the complexity. This paper presents an initial general description of workflows for molecular science and details the result based on two examples for the integration of Gaussian and Gromacs.

## 1 Workflows and workflow management in MoSGrid

Computational Chemistry has evolved during the last two decades into an established discipline in natural sciences. Scientists from many disciplines are using molecular simulations to gather information about the properties of molecules, small and large alike. Computational procedures developed by theoretical chemists and physicists as well as mathematicians and bioinformaticians are now used on a daily bases as high performance computing infrastructures have become available to everyone. However, as computational chemistry allows users to answer increasingly complex scientific questions, more and more

---

\*This work is supported by German Ministry of Education and Research under project grant #01IG09006.

scientists are encouraged to apply these tools, and consequently the demand for computational resources strongly increases.

There are, however, a number of limitations. The number of programs for molecular simulations (electronic structure methods and molecular mechanics/dynamics codes) that are well-established and accepted in the scientific community is still quite small. Furthermore, the usability of these tools is often limited. This is partly due to the design of the user interfaces, since a lot of sophisticated tools lack a graphical and easy to use interface. Altogether, many new users to this field have to be acquainted not only with the large number of methods and chemical theories that are implemented in these programs but also in operating the latter. To lower the hurdle of using these programs, there is the need for intuitive user interfaces.

MoSGrid anticipates to ease the handling of the large complexity of molecular simulation methods. The main goal of this BMBF funded joint research project of the German Grid (D-Grid) Initiative is to make molecular simulation codes available for German Grid resources. Molecular simulation codes and computational resources are intended to be accessible using the MoSGrid portal, which will offer various tools to import molecular information, to setup and submit calculations, and to extract relevant results easily.

The MoSGrid portal is going to offer a (graphical) workflow manager. Commonly used simple and complex workflows can be stored in recipe repositories and be made available for every user. All users can develop, improve, publish and use workflows for their everyday tasks. As a result of these efforts the variety of application cases increases, making the use of computational chemistry tools easier for less experienced users at the same time.

In this paper, we describe the nature of typical workflows and their management in the field of computational chemistry. The requirement analysis is based on a comprehensive survey within the computational chemistry community. Some preliminary results have already been presented in [NBB<sup>+</sup>09].

## 2 Computational Requirements in Molecular Structure Simulations

Molecular structure simulations can be demanding in various respects on several different levels. One level is the problem definition that depends on the chemist. In this process the experience and knowledge of the user about the particular simulation technique and its advantages and drawbacks is a critical element.

Another level is the resource requirements definition. This is more or less a computer science issue that depends on the size of the problem, on the algorithms applied and partly even on the hardware architecture to be used. This leads to a further level.

In *molecular mechanics* the problem size is determined by the number of atoms in the simulation and can be increased in a parallel tempering sampling approach [ED05] by the factor of the number of replicas.

A bulk simulation consisting of a vast amount of molecules and atoms can be reduced

to several smaller problems with interdependences, e. g. by domain decomposition or separated electrostatic field calculations [HKvdSL08].

Replica exchange or parallel tempering simulates different replicas with defined order parameters like temperature or pressure, eventually exchanges them and subsequently continues the simulation on the exchanged replica. Dependent on the number of atoms many different replicas can be necessary to achieve a good sampling. Both approaches require at least several processors and a – shared or distributed – common memory. This can be reasonably achieved by a MPI (Message Passing Interface) implementation that is realised in the Gromacs molecular dynamics package [HKvdSL08].

In *quantum mechanics* the system size is dependent on the number of basis functions and the calculation complexity on the method employed. The solution of such systems can be enhanced by, e. g. the parallel solving of the self consistent field equations.

A screening setup that employs a quantum mechanical approach to calculate certain properties of several hundred up to millions of molecules where each of them in their own simulation can increase the problem size. This screening consists per se of several tasks without internal dependences which means that every task can be solved on a different resource. Depending on the problem, the calculation can be enhanced and accelerated by multiple processors [FTSea09].

### 3 General Workflow Description

One of MoSGrid's aims is to simplify the access to molecular simulation tools. This will be accomplished by providing a collection of recipes to the user of the MoSGrid portal. Chemists are familiar with recipes from their daily work and in this context working with a recipe defines an ontology.

A recipe is a concept that can be implemented using a workflow. The workflow that will be applied to the simulation codes in the MoSGrid project can be depicted as a multi-step process consisting of the following tasks:

**Job definition** that describes the problem the user wants to solve. For this description the user has to provide information with intelligent assistance of the portal. Every chemical question concerns a structure of interest and one or more of its properties. The user starts a workflow by uploading a structure (which is possible in several different formats) or by usage of a data repository that is also part of the MoSGrid developmental efforts. This repository will contain descriptions of molecular structures and results of structure calculations in a meta molecule description language. The input of this information is supported by a continuous

**Metaprocessing** that checks the user input for consistency. This processing will assist the user to provide a complete set of meta information describing the job, e. g. structure, application, method, temperature or basis set. To avoid annoyance of the user by (over-)demanding input masks sensible default values will be provided by the system as preset values which can be overwritten by the user.

Afterwards this molecular simulation meta description (MSMD) will be translated to the user in a totally transparent manner by a

**Preprocessing** step resulting in an application specific input format. For this process adapters will be developed for several quantum chemistry and molecular dynamic codes, e. g. Gaussian and Gromacs. As far as possible the estimation of resource requirements as memory, storage or number of nodes and processors shall also be done by these adapters. Concerning partial platform specificity of input files a differentiation between a portal based and a resource based preprocessing has to be done.

After conversion of the MSMD and generation of the input files, the

**Job Submission** is initiated. This can be understood as queuing a middleware specific container which is an execution environment on a grid resource. All the described information is then transferred to this middleware environment, i. e. the staging in of the generated input files and additional input, like structure format files, as well as the translation of the system requirements into a batch system call.

**Steering** is a feature that in a first implementation will rely on a simple monitoring and abortion mechanism that allows the user to interfere and stop the simulation, if it develops not as expected. For this purpose the user is granted access to the result files of the ongoing simulation and is assisted by tools that generate graphical representations of the current simulation state. The steering option is intended to avoid waste of time and resources. After successful execution of the job a

**Postprocessing** will be performed to extract application independent information from the result files. This information will be reported in an adequate form, e. g. as charts or graphs, on the portal and can be optionally stored in the data repository provided by MoSGrid.

The huge benefit of this approach is the fact that the process of designing a molecular simulation is separated from the application. Thus, the user can concentrate on the definition of the scientific part of the simulation while MoSGrid cares about the tedious steps of preprocessing, job submission, and postprocessing in a totally transparent way.

## 4 Examples for Grid-Workflows in Molecular Science

The following sections describe two selected applications of workflows in molecular science. We focus on the analysis of a typical simulation process in quantum chemistry (Gaussian) and molecular dynamics (Gromacs).

## 4.1 Gaussian

Gaussian is a quantum mechanics package that has been widely used for over three decades by a large scientific community [FTSea09]. The first step in a Gaussian workflow (see Fig. 1) is the definition of a molecular structure. The molecule is typically created by a graphical molecule editor like GaussView. At a later stage of the MoSGrid project, the basic molecule-definitions will be available from a molecular database and the ability of editing the molecule will be added to the portal itself.

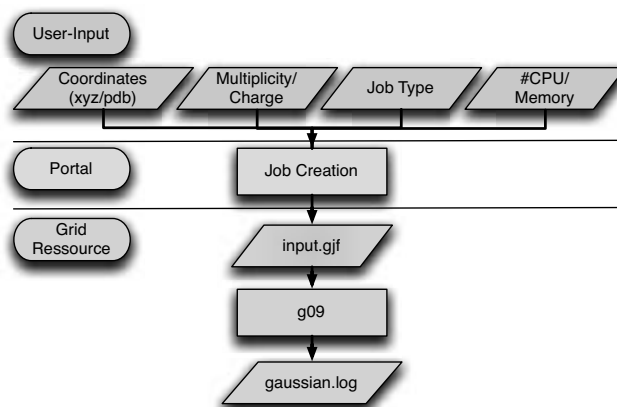


Figure 1: Basic Gaussian workflow showing default input and output.

Furthermore, technical information like the memory and the number of CPU's have to be inserted. This could be added automatically by the preprocessor upon resource selection. Following our recent analysis, all required information can be collected in a Grid portal. Even the preprocessor is likely to be provided by the portal itself, as these steps are not compute-intensive and a direct user input is possible. After molecule creation, job definition, and preprocessor check, a Gaussian job file is submitted into the Grid, and computed with Gaussian *g09*. After the termination of the job, the user analyses the results and mostly submits subsequent calculations based on this first calculation. In a later stage of MoSGrid these subsequent calculations can be added to the workflow as well.

This workflow can be extended by using a huge array of molecules as input which will be processed in parallel. For this purpose, the portal will offer the possibility to select an array of molecule coordinates, convert them into the corresponding gjf-files with pre-defined information and to subsequently submit the bulk to the Grid.

## 4.2 Gromacs

Gromacs 4 is a software package for molecular dynamics (MD) simulations. It is considered one of the fastest MD codes available [HKvdSL08]. Examples of MD applications, covering only a small portion of possible applications, are the study of drug-receptor interaction, shear viscosities of technical fluids, folding of pharmaceutically interesting proteins, or ion permeation through ion channels [KF09]. Gromacs is an open-source project originated from the University of Groningen, now maintained by an active community of users and developers from all over the world [HKvdSL08].

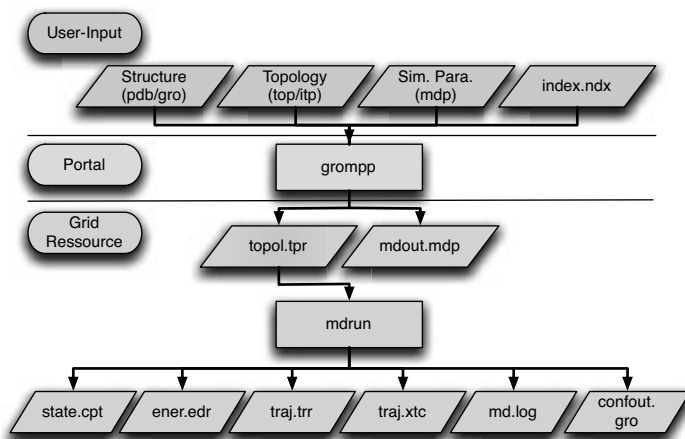


Figure 2: Basic Gromacs workflow showing default input and output.

The most simple Gromacs workflow (2) consists of the call of the preprocessor *grompp*, which joins input data such as the atom coordinates and system topology with the calculation specification into a single input file (*topol.tpr*). The actual calculation is performed by the (parallel) MD application (*mdrun*) taking great advantage of high performance compute resources with low latency network interconnects. Real life examples for anticipated Gromacs workflows for the MoSGrid projects are (i) full equilibrations of globular proteins in explicit water, (ii) semi-automatic free energy perturbation simulations to determine the free energy of solvation of drugs and other small molecules, (iii) semi-automatic linear interaction-energy based calculations of the binding energy of drugs to their receptors, (iv) semi-automatic property studies of bulk or water solute liquid crystal materials.

## 5 Selected Workflow Engines

In accordance to MoSGrids workflow requirements, several workflow engines could be chosen for implementation. The following will give a brief introduction to some potential workflow engines for MoSGrid. However, which engine to take is not decided yet.

**UNICORE 6** The UNICORE 6 embedded workflow engine follows a data driven workflow concept. The different steps of a workflow are described in an XML format. At execution the workflow service decides which step can be executed and forwards this step to a service orchestrator that assigns the subjob to a compute resource. When a subjob finishes, the workflow manager checks if a following subjob can be started due to predefined and now fulfilled dependencies.

**BPEL** BPEL defines atomic operations as WSDL described web services. This allows the usage of an external web service inside a workflow instead of executing it as a subjob on the resources. The BIS-Grid project incorporated the ActiveBPEL workflow engine into UNICORE 6. This added functionality allows to execute a frequently used parser as a web service either in the portal or as a subjob in the UNICORE workflow.

**jBPM** jBPM from JBoss is a Java-based framework for orchestrating and invoking workflows by using the proprietary process language jPDL (jBPM Process Definition Language) and the workflow engine PVM (Process Virtual Machine). PVM supports multiple process languages and offers to combine computing tasks with human interactions. It is easy to extend with Java expertise and has an active community. One of its drawbacks is that it also requires Java expertise for a basic action like orchestrating workflows, e.g. to integrate web services.

**MSS** MSS is designed for co-allocation of computing, software licenses and networking resources. Important is the reservation of network bandwidth for workflows that require QoS during stage-in, compute, and stage-out. MSS uses WS-Agreement as interface to communicate with external components like the middleware or the user client.

**WSS** The Workflow Scheduling Service (WSS) is a workflow management and scheduling component. WSS deals with workflow management and execution including data extraction, transport, job execution and fully integrates into the provided Globus WSRF framework. Further, WSS enables users to describe workflows with a simple XML-based Workflow Specification Language (WSL) that encapsulates JSDL.

**GWES** The Grid Workflow Execution Service (GWES) is a workflow enactment engine with Meta scheduling capabilities. GWES coordinates the composition and execution process of workflows. It implements a workflow concept by means of the Grid Workflow Description Language (GWorkflowDL), which is based upon the theory of Petri nets. It provides interfaces to the Web Portal, to a command line client, and to Globus Toolkit 4.

**GridWay** The GridWay meta-scheduler describes interdependencies between tasks of a workflow in a “Job Description Document” (JDD) or JSDL. GridWay is part of the Globus Toolkit and its Job Management assigns every job, to a host for execution due to job dependencies, resource requirements and resource limits. A Dispatch Manager



performs the submission and watches over the execution of a job, the Execution Manager is responsible for job execution and management, and a Transfer Manager is responsible for file staging, handling the remote working directory and remote host clean-up.

All workflow engines support the basic workflow functionalities required by MoSGrid. However, during the definition of MoSGrid, it was decided to use UNICORE 6 as Grid middleware as it is widely distributed and already offers graphical user interfaces for an intuitive access to Grid resources even for non computer scientists. Therefore, the further analysis of workflow engines will be restricted to those usable with UNICORE 6 and compare them to MoSGrid's requirements. In focus of this process will be the UNICORE 6 embedded workflow engine and BIS-Grid's ActiveBPEL workflow engine.

## 6 Conclusion and Future Work

MoSGrid is at an early stage of the project. Therefore, this paper describes the work done for the conceptual design of typical workflows. Based on the workflow concept an evaluation of workflow engines will follow. MoSGrid will use UNICORE 6 as Grid middleware restricting the analysis of workflow engines. We will focus on the UNICORE 6 embedded workflow engine and the BPEL integrated workflow engine originating from the BIS-Grid project. A detailed evaluation of the engines will affect the deployment of the MoSGrid portal.

## References

- [ED05] David J. Earl and Michael W. Deem. *Parallel Tempering: Theory, Applications, and New Perspectives*, 2005.
- [FTSea09] M. J. Frisch, G. W. Trucks, H. B. Schlegel, and et. al. *Gaussian 09 Revision A.1*, 2009. Gaussian Inc. Wallingford CT.
- [HKvdSL08] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *Journal of Chemical Theory and Computation*, 2008.
- [KF09] J. Krüger and G. Fels. Potential of Mean Force of Ion Permeation through alpha7 nAChR Ion Channel. In *Proceedings of IWPLS'09, Edinburgh, UK, September 14-15, 2009, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-513/paper06.pdf*, 2009.
- [NBB<sup>+</sup>09] O. Niehörster, G. Birkenheuer, A. Brinkmann, B. Elsässer, D. Blunk, S. Herres-Pawlis, J. Krüger, J. Niehörster, L. Packschies, and G. Fels. Providing Scientific Software as a Service in Consideration of Service Level Agreements. In *Proceedings of the Cracow Grid Workshop (CGW)*, 2009.

# Workflow Modeling for WS-BPEL-based Service Orchestration in SMEs

S. Gudenkauf, G. Scherp, W. Hasselbring  
OFFIS Institute for Information Technology  
Technology Cluster EAI  
Escherweg 2, 26121 Oldenburg, Germany  
*email:* [stefan.gudenkauf, guido.scherp]@offis.de,  
wha@informatik.uni-kiel.de

A. Höing, O. Kao  
Technische Universität Berlin  
Complex and Distributed IT Systems  
Einsteinufer 17, 10587 Berlin, Germany  
*email:* [andre.hoeing, odej.kao]@tu-berlin.de

**Abstract:** The BIS-Grid project<sup>1</sup>, a project in the context of the German D-Grid initiative, investigates Grid and Cloud service orchestration for information systems integration, especially when crossing enterprise boundaries. Small and medium enterprises shall be enabled to integrate heterogeneous business information systems and to use external resources and services with affordable effort. In this paper, we present the workflow modeling method that we developed within the BIS-Grid project.

## 1 Introduction

The integration of heterogeneous information systems is crucial in order to map business processes to the technical system level. To do so, integration is often achieved by service orchestration in service-oriented architectures (SOA). Web services are commonly used to create SOA since they enable service orchestration and hide the underlying technical infrastructure. SOA and Web service technologies are also the basic technologies for the newly emerging Cloud computing paradigm. Cloud computing provides easy access to IT infrastructures, computing platforms, or complete applications. These characteristics are also referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In the BIS-Grid project<sup>2</sup>, we focus on the integration of enterprise applications (Enterprise Application Integration, EAI) using Grid and Cloud service technologies. Our major objective is to prove that these technologies are feasible for information systems integration for small and medium enterprises (SMEs), especially

---

<sup>1</sup>This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant No. 01IG07005 as part of the D-Grid initiative.

<sup>2</sup><http://www.bisgrid.de>

when traversing enterprise boundaries. SMEs shall be enabled to integrate heterogeneous business information systems and to use external resources and services with affordable effort. To do so, we proposed Orchestration as a Service (OaaS) as the primary infrastructure paradigm – a specialization of PaaS where a service orchestration engine is hosted in a Cloud environment, directly to be maintained by the OaaS provider – and provided an adequate OaaS infrastructure, see [GHH<sup>+</sup>09, HSG<sup>+</sup>09].

However, before even regarding the utilization of externally provided Grid and Cloud services, SOA can be regarded as the main requirement for SMEs, both from an organizational and from a technical perspective (cf. [BGJ<sup>+</sup>09]). While the organizational perspective requires SMEs to structurally align with an underlying SOA infrastructure and to define the different layers of service provision, the technical perspective requires to formalize the business processes, to separate the parts of the processes that can or shall be automatized from those that are subject to human activity, and to define the basic services, for example. Within the BIS-Grid project, we evaluate the OaaS approach and our OaaS infrastructure in two business scenarios motivated by our industrial project partners, a services partner for first-class trade brands on the European photographic market supplying stores and internet retailers with photographic products, and a global market leader in the field of wire drawing and draw-peeling for the automotive industry. Both have strong needs for service orchestration, the first one to integrate enterprise data for unified access for call center agents, and the latter to improve access to, and retrieval and maintenance of product and project data across different sites of the company group. However, the industrial partners are SMEs that initially did not have extensive practice and experience in workflow technologies and SOA as well as Grid and Cloud Computing. While the lack of expertise in setting up and maintaining a service-oriented computing infrastructure was compensated by the OaaS scenario in which the technical partners of the project act as service providers for an orchestration engine, process orientation and workflow modeling still remained as requirements for the SMEs. For the pragmatically definition of workflows for such SMEs, a method was needed that is simple to follow, general, and effective. This method is presented in the next section.

## 2 Workflow Modeling Method

Within our application scenarios, we developed and employed a top-down workflow modeling method that is small enough to target service orchestration in prototype SOA scenarios. Figure 1 presents an overview of this method. The upper half of the figure shows the creative activities of the method, and the lower shows the involved infrastructure components. Associated business roles are annotated to both the activities and the components, and the arrows informally depict the main dependencies between the respective activities and components. The design of a workflow mainly depends on a process model that is delivered by *business analysts* in the form of Business Process Modeling Notation diagrams (<http://www.bpmn.org/>) as a high-level abstraction to describe the business process, and on services that are provided by *service developers* – either by identifying existing ones or by genuine implementation. There is an information exchange between the individual

roles, for example, to gain a common view of the global data model. To provide an executable workflow, the workflow must be deployed on an adequate workflow engine (here, the BIS-Grid Workflow Engine) and, of course, all used services must be made available at their respective service execution environments.

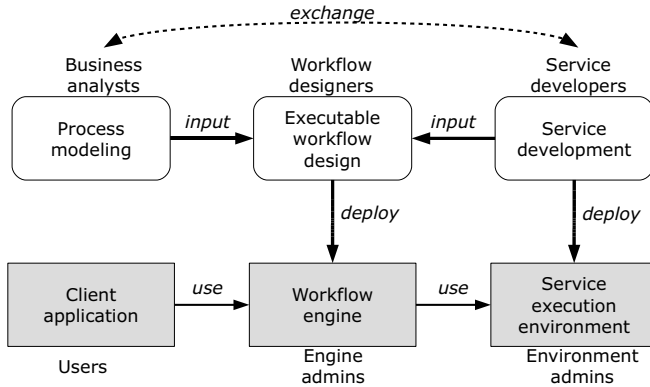


Figure 1: Overview of the workflow modeling method.

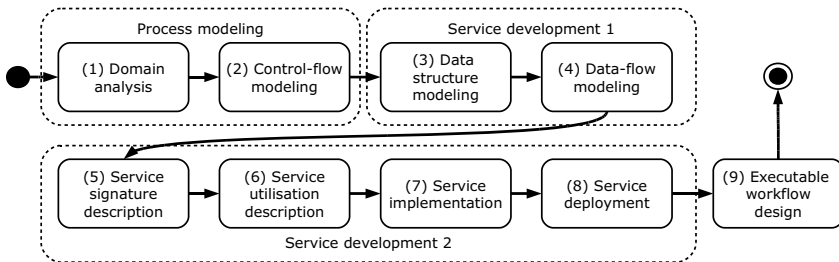


Figure 2: Applied workflow development process.

Figure 3 presents an UML object diagram that gives a more specific overview of the individual artifacts of the method and the conceptual background. It represents a method model that focuses on the *RepresentationOf* ( $\mu$ ) and *DecomposedIn* ( $\delta$ ) relations between the objects, while the objects themselves represent systems. While the  $\mu$  relations identify a given system as the *model* of another system that is called system under study (*sus*), the  $\delta$  relation simply identifies a system as a *part* of a *composite* system. We find that such  $\mu\delta$  graphs provide a good overview on the elements of a method, ranging from the conceptual problem decomposition behind the method (i.e., the abstract systems, AS), to the digital systems (DS) to be developed, to the physical systems (PS) – i.e., diagrams and formal models – that are used to alleviate the development of the digital systems by focusing on specific aspects of the problem. Please note that the *RepresentationOf* ( $\mu$ ) relation is not necessarily transitive [Fav04a], and that automated transformations between systems are not in the focus. Also, the roles of the respective systems are only depicted exemplar-

ily (blue). For more information on this kind of modeling-in-the-large, see, for example, [Fav04a, Fav05, Fav04b].

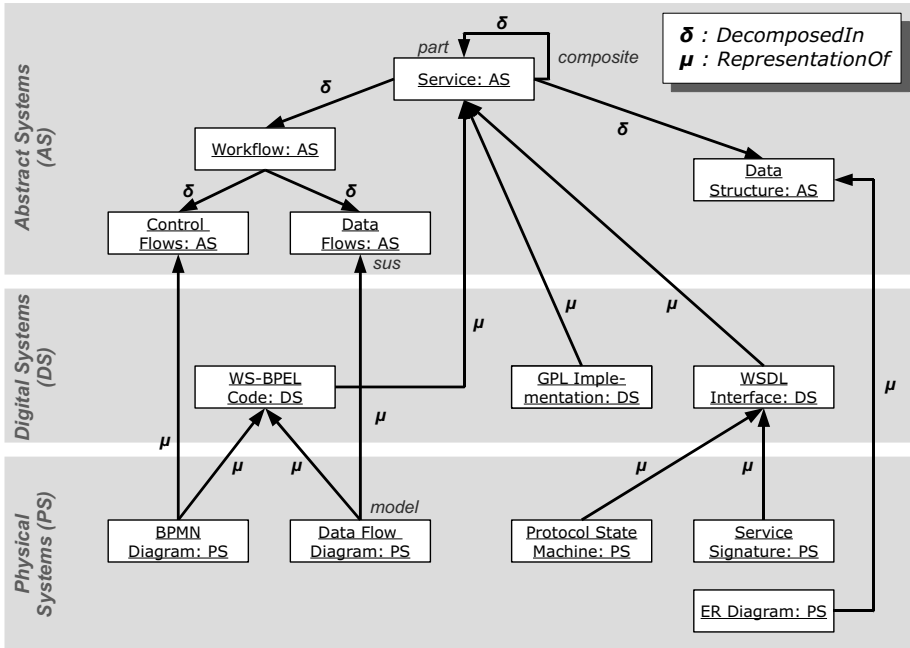


Figure 3: Model of the workflow modeling method.

The fundamental distinction is that a computer program consists of computation that has to be coordinated (cf. [GC92]). Thereby, the fundamental representation of a computer program is a service, which itself can be decomposed in services. We consider a workflow as the concrete embodiment of the coordination structure of the program, and deployed services as the concrete embodiment of computation, since they are then regarded as black boxes. An executable service can consist of, for example, the WS-BPEL process description, primitive service implementations given in general programming language (GPL) code, and the WSDL interface descriptions. In order to alleviate the development of an executable service, various diagrams and models are employed that represent parts of a service while focusing on specific aspects of the service: (1) BPMN diagrams focus on the control flows of the workflow. (2) Data flow diagrams (DFD) focus on the data flows of the workflow. (3) Protocol state machines describe the utilization of services to be orchestrated. (4) Signature descriptions describe the interfaces of services with multiple in and out-parameters. (5) Entity relationship diagrams describe the data structures to be used within the services. Based on this model, the individual process steps to identify, model, and deploy business workflows are as follows (see Fig 2).

- (1) *Domain analysis.* The respective business domains have to be analyzed to gain a thorough domain understanding. This includes, for example, the analysis of the

current enterprise architecture, expert interviews, on-site investigations, and requirements analysis.

- (2) *Control-flow modeling*. This activity is composed of the following sub-activities:
  - (a) The current business processes (as-is state) is described using the Business Process Modeling Notation (BPMN).
  - (b) From the as-is state a first version of the to-be processes are developed and described using the BPMN, too.
  - (c) Data sources and simple data-flows were annotated in the to-be BPMN diagrams as far as possible using the BPMN (see the fourth lane from the top in Fig. 4).
  - (d) The to-be BPMN diagrams are iteratively expanded to regard different layers of abstraction. Thereby, we consider the operational layer, the services layer, the business process layer, and the consumer layer. We especially find this activity to be very helpful in order to separate concerns at an early stage of development (cp. SOA reference architecture in [BLJ<sup>+</sup>08]). Figure 4 illustrates a such-layered call center process from the first application scenario for read-only data retrieval (layers are ordered from the bottom to the top).
- (3) *Data structure modeling*. Upon the relevant information systems and databases, the logical structure of the required information is modeled. To do so, we use entity-relationship (ER) diagrams that represent the relevant data structures whereas the information system/database origin of the structures is annotated.
- (4) *Data-flow modeling*. In addition to control-flow modeling, we model the data-flows of business processes using data flow diagrams (DFDs).
- (5) *Service signature description*. Based upon the results of the previous activities, we textually describe the signatures of the services of the respective business processes as a basis for service interface definition.
- (6) *Service utilization description*. In addition to signature description, we describe the usage protocol of the services regarded as black-boxes using protocol state machines. Although this activity represents an overhead for services with small signatures, we think that this activity is of great value for services that provide several operations and where the operations have strong service lifecycle dependencies.
- (7) *Service implementation*. Starting with WSDL interface design, the services are implemented.
- (8) *Service deployment*. The services are deployed under consideration of the enterprise architecture and the scenario requirements, for example, security requirements.
- (9) *Workflow design*. Finally, we implement WS-BPEL workflows for the modeled business processes.



### 3 Related Work

The Quasar Enterprise (QE) Method is a coherent method for realizing service-oriented architectures [EHH<sup>+</sup>08]. It provides detailed guidelines for designing and providing services which is driven by the identification of business goals and requirements. There are also many other approaches for SOA, for example, [Mel08, KBS04, AGA<sup>+</sup>08]. We regard them to be targeted at a much broader SOA realization than our approach, but possibly not as broad and comprehensive as QE.

Currently, each step in our method relies on manual execution. Particularly with regard to the concrete service development steps (e. g. WS-BPEL process and Web service implementation), there already exist approaches for (semi-)automatic transformation. In [OvdAD06], for example, a method is described to translate BPMN diagrams to WS-BPEL processes. On the technical side, model driven technologies [RH09] can be used to perform such transformation steps [YZZ<sup>+</sup>07].

In contrast to full-fledged SOA methods, the method presented in this paper is much more narrow, and specifically focuses on soft-wired service orchestration in SMEs, where the targeted SMEs are supposed to have limited resources for and little experience with service orchestration in order to undertake a complete SOA introduction in one single commitment. From this point of view, the presented method can be seen as a simple method for a first step in SOA introduction to be applied in a prototype service orchestration scenario.

### 4 Conclusion

In this paper, we presented the workflow modeling method that we developed within the BIS-Grid project. Although the project is aimed specifically at Grid and Cloud service orchestration that can cross enterprise boundaries, the method targets WS-BPEL service orchestration in general.

### References

- [AGA<sup>+</sup>08] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Gariapathy, and K. Holley. SOMA: A method for developing service-oriented solutions. *IBM Syst. J.*, 47(3):377–396, 2008.
- [BGJ<sup>+</sup>09] Melanie Baier, Gernot Gräfe, Melanie Jekal, Florian Röhr, and Tanja Vörckel. Marktanalyse für kommerzielles Grid-Providing . Technical report, Universität Paderborn and Siemens AG - Siemens IT Solutions and Services, July 2009.
- [BLJ<sup>+</sup>08] Norbert Bieberstein, Robert G. Laird, Keith Jones, Tilak Mitra, and Jason Weisser. *Executing SOA: A Practical Guide for the Service-Oriented Architect*, chapter A Methodology for Service Modeling and Design, pages 57–81. developerWorks Series. IBM Press, May 2008. 2008 Dimensions 7x9-1/4 240 Edition: 1st. 0-13-235374-1 ISBN-13: 978-0-13-235374-8.



- [EHH<sup>+</sup>08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, and Johannes Willkomm. *Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten*. dpunkt, Heidelberg, 2008.
- [Fav04a] Jean-Marie Favre. Foundations of Model (Driven) (Reverse) Engineering : Models - Episode I: Stories of The Fidus Papyrus and of The Solarus. In *Language Engineering for Model-Driven Software Development*, 2004.
- [Fav04b] Jean-Marie Favre. Towards a Basic Theory to Model Model Driven Engineering. In *Workshop on Software Model Engineering, WISME 2004, joint event with UML2004*, 2004.
- [Fav05] Jean-Marie Favre. Foundations of Meta-Pyramids: Languages vs. Metamodels – Episode II: Story of Thotus the Baboon. In Jean Bezivin and Reiko Heckel, editors, *Language Engineering for Model-Driven Software Development*, number 04101 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- [GC92] David Gelernter and Nicholas Carriero. Coordination Languages and their Significance. *Communications of the ACM*, 35:97–107, 1992.
- [GHH<sup>+</sup>09] Stefan Gudenukauf, Wilhelm Hasselbring, André Höing, Guido Scherp, and Odej Kao. Workflow Service Extensions for UNICORE 6 - Utilising a Standard WS-BPEL Engine for Grid Service Orchestration. *Lecture Notes in Computer Science: Euro-Par 2008 Workshops - Parallel Processing*, 5415:103–112, 2009.
- [HSG<sup>+</sup>09] André Höing, Guido Scherp, Stefan Gudenukauf, Dirk Meister, and André Brinkmann. An Orchestration as a Service Infrastructure Using Grid Technologies and WS-BPEL. In *ICSOC/ServiceWave*, pages 301–315, 2009.
- [KBS04] Dirk Krafzig, Karl Banke, and Dirk Slama. *Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
- [Mel08] Ingo Melzer, editor. *Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis*. Spektrum, Heidelberg, 3. edition, 2008.
- [OvdAD06] C. Ouyang, WMP. van der Aalst, and AHM. ter Hofstede M. Dumas. Translating BPMN to BPEL. Technical report, 2006.
- [RH09] Ralf Reussner and Wilhelm Hasselbring, editors. *Handbuch der Software-Architektur*. dpunkt, Heidelberg, 2. edition, 2009.
- [YZZ<sup>+</sup>07] Xiaofeng Yu, Yan Zhang, Tian Zhang, Linzhang Wang, Jianhua Zhao, Guoliang Zheng, and Xuandong Li. Towards a Model Driven Approach to Automatic BPEL Generation. In *ECMDA-FA*, pages 204–218, 2007.

# Security, Fault Tolerance and Modeling of Grid Workflows in BPEL4Grid\*

Ernst Juhnke, Tim Dörnemann, Roland Schwarzkopf, Bernd Freisleben  
Dept. of Mathematics and Computer Science, University of Marburg  
{ejuhnke, doernemt, rschwarzkopf, freisleb}@informatik.uni-marburg.de

**Abstract:** BPEL is the de facto standard for business process modeling in today's enterprises and is a promising candidate for the integration of business and scientific applications that run in Grid or Cloud environments. In this paper, selected components of *BPEL4Grid*, a scientific workflow system for developing service-oriented Grid applications based on BPEL, are presented. The focus of the paper is on security aspects of workflow composition and fault tolerance mechanisms for long-running workflows. Furthermore, two workflow modeling tools targeted at users with different levels of knowledge about BPEL are briefly described.

## 1 Introduction

The Business Process Execution Language for Web Services (BPEL4WS or WS-BPEL [ACD<sup>+</sup>03]) is the de facto standard for web service composition in business applications. It enables the construction of complex web services composed of other web services that act as the basic activities in the newly constructed service. Access to a *process* is exposed by the execution engine through a web service interface (Web Services Description Language, WSDL), allowing the process to be accessed by web service clients or to be used as a basic activity in other processes.

This paper presents an overview of some of the key components of BPEL4Grid (see Figure 1), a scientific workflow system for developing service-oriented Grid applications, built on BPEL and related standards (SOAP, WSDL, Eclipse). In particular, security aspects of workflow composition and fault tolerance mechanisms for long-running workflows are discussed. Furthermore, two workflow modeling tools that we have developed are briefly described: (1) DAVO (Domain-Adaptable Visual Orchestrator which is the foundation for ViGO (Visual Grid Orchestrator), a graphical modeling tool that supports the BPEL standard and our Grid extensions. (2) SimpleBPEL, a tool that allows domain experts with little or no knowledge about BPEL to compose workflows from predefined, domain-specific sub-workflows ("snippets").

---

\*This work is financially supported by the BMBF (D-Grid Initiative)

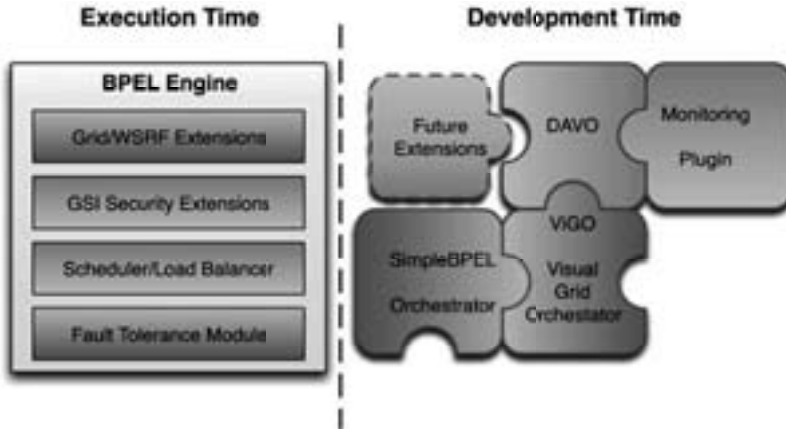


Figure 1: Components of BPEL4Grid – only components highlighted in blue will be described

## 2 Grid Workflow Security

While BPEL works well for traditional web services, it has a number of drawbacks with respect to the more complex world of WSRF-based Grid computing, especially where security is concerned. The BPEL security concept is not equipped to deal with complex multi-protocol Grid environments and does not integrate with the Grid Security Infrastructure (GSI). While BPEL is mainly focused on anonymous HTTPS-based TLS security or manual role-based authentication encoded in SOAP headers, Grid computing has a mandatory user-centric security approach using X.509 certificates, which far exceeds the scope and capability of the BPEL security model.

To support GSI, we have further extended our previously published [DFH<sup>+</sup>07] Grid-related BPEL extension *gridInvoke* by security-related settings [DSF08]. The extended *gridInvoke* activity allows us to define, for example, the security method to be used, choose whether to use encryption or signing of messages, and the delegation level of proxy certificates. The syntax of the extension is described in Listing 1.

```
<gridInvoke ...>
<security
  method="GSItransport | GSISecureMessage | GSISecureConversation"
  level="privacy | integrity"
  authz="none | self | host | anyString"?
  peer-credentials="filename"? anonymous="true | false"?
  delegation="none | full | limited"? />?
</gridInvoke>
```

Listing 1: Syntax of the security settings for invocation

Since the runtime of a process might be unknown and therefore longer than the proxy

certificate's lifetime, automatic renewal of proxy's is offered by the BPEL4Grid engine. The workflow engine monitors both the runtime of the process and the proxy's lifetime. If the proxy's lifetime is about to expire, the engine will renew the certificate if desired by the user of the workflow. The BPEL engine contacts the given MyProxy server and retrieves a proxy certificate with the given lifetime from the server. Figure 2 illustrates this sequence. Since the whole conversation is secured using HTTPS (from client to BPEL engine) and pure TLS (from BPEL engine to MyProxy), it can be considered as secure. Further details, a discussion of the implementation and a performance evaluation can be found in a previous paper [DSF08].

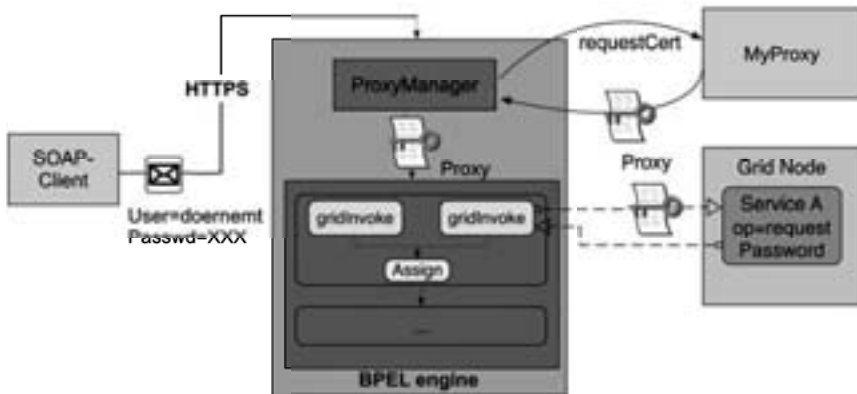


Figure 2: Integration of BPEL and Grid Security Infrastructure

### 3 Fault Tolerance of Grid Workflows

When long-running or computationally-intensive workflows are to be executed, fault handling is very important, since the failure of a single component might lead to an abandonment of the entire workflow. Many faults can be corrected by either simply retrying the failed operation or by substituting the failed component by an equivalent one.

While composition languages like WS-BPEL offer fault handling mechanisms, *infrastructural failures* like network timeouts and server outages should not be handled using the language mechanisms, since this would clutter the composition logic with non-functional aspects. Consequently, we identify classes of faults that can be handled automatically and define a policy language to configure automatic recovery behavior without the need for adding explicit fault handling mechanisms to the BPEL process. Furthermore, the approach provides automatic Cloud-based redundancy of services to allow substitution of defective services.

The developed solution [JDF09] adds a policy-based fault handling mechanism to BPEL without making any changes to the language standard. Using policies, it allows to enable and disable both retry and substitute actions. To reduce the number of required policies, a

Fault Classifier (FC) categorizes faults into groups. Instead of defining a policy for every type of fault, policies can be specified for each group of faults instead. Furthermore, the policies permit to set parameters such as the maximum number of retries, what kind of resources (dedicated hosts, Cloud resources, etc.) may be used for substitution and so on.

The Message Monitor (see Figure 3) monitors the response messages from invoked services and checks whether they contain fault messages or not. In the first case, the corresponding message is passed to the Fault Classifier (step 2). The classifier classifies the fault using rules stored in the Fault Database; the result is then passed (step 3) to the Policy Processor that applies all configured policies. The result contains zero or more recovery strategies with priorities (Retry is to be performed before Substitution, for instance).

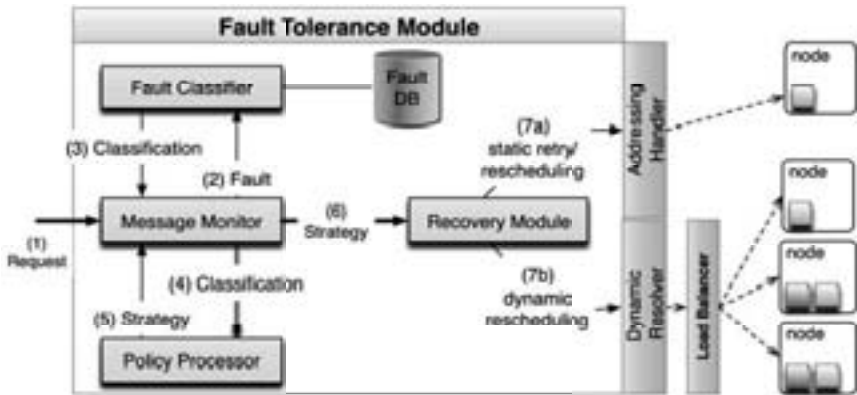


Figure 3: Sub-components of the Fault Tolerance Module

If the Retry strategy is applied, the previously invoked service is invoked again (step 7a). Otherwise, the Dynamic Resolver [DJF09] is executed to perform a dynamic scheduling on the resources declared in the Substitution strategy (step 7b). These resources may either be machines from a pool of available resources or virtual machines that are provisioned on-demand from a Cloud infrastructure like Amazon’s EC2 [ec2] or Nimbus [nim]. Details concerning Cloud-based redundancy and the architecture of the on-demand provisioning framework can be found in [JDF09, DJF09].

## 4 Grid Workflow Modeling

### 4.1 Domain-Adaptable Visual Orchestrator (DAVO)

DAVO is a domain-adaptable, graphical BPEL editor [DMS<sup>+</sup>09]. The key benefits that distinguish DAVO from other editors are the *adaptable data model* and the *user interface*, which permit customization to specific domain needs.

In DAVO, the activities are represented using objects of the class hierarchy shown in Figure

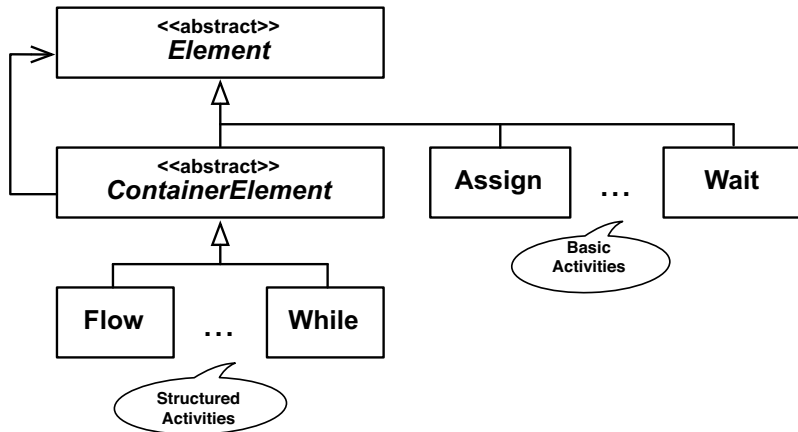


Figure 4: A simplified Element class hierarchy.

4. Element is the parent class of all activities, whereas ContainerElement is the parent class for all structured activities. The actual class hierarchy is more complex than the one displayed here. It contains additional abstractions for activities referring to other web services and sequential structured activities, which are omitted for simplicity since they are not relevant for the data model.

While it is sufficient for a standard BPEL workflow editor to use a simple data model to store the process' information, i.e. information for elements and attributes, this approach is not feasible with regard to extensibility. For the use of BPEL within and the adaptation of DAVO to specific domains, it is necessary to associate additional information with an activity (e.g. Grid Security-related settings). DAVO uses named properties to associate arbitrary information with activities. Besides name and value, these properties also contain an IValidator that can be used to check validity when setValue() is called. Additionally, the properties itself have various meta-properties, such as:

- persistent determines if the property value is stored together with the DAVO data model.
- readOnly and visible, which are used (together with various other meta-properties) to control the automatic creation of property views, as described in [DMS<sup>+</sup>09].

The extension mechanism is illustrated in Figure 5. The core of each DAVO extension is an implementation of the IModelExtender (IME). A specific IME implementation knows the ElementExtension for each Element. After the ElementFactory has created the Element (1), it passes it to the ElementExtender (2), which then asks the IME for extensions for the given Element (3). After the IME has created the extension for the given Element (4), it is added to the Element (5). The ElementExtension can modify the Element in many ways. For example, it can add new properties or hide existing ones.

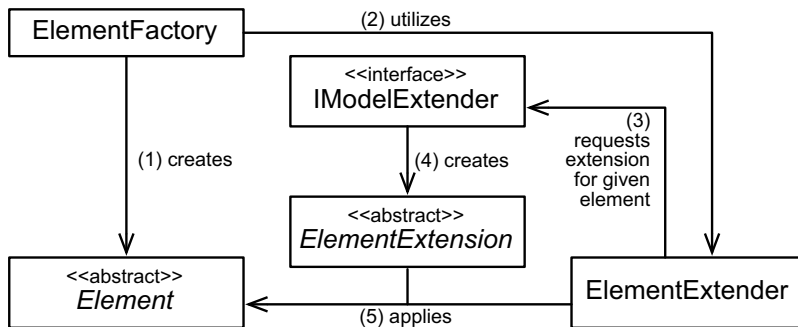


Figure 5: The Element extension mechanism.

New activities can be added by simply inheriting from an arbitrary class from the hierarchy and registering the new activity using one of DAVO’s extensions points. As consequence, the new activity is added to DAVO’s graphical user interface, as described in [DMS<sup>+</sup>09].

The described adaptability and extensibility features were used to implement the data model and visual representation of the aforementioned Grid-specific extensions in ViGO. Besides that, ViGO offers assistants to model the invocation of Grid services, allows us to graphically define security requirements and includes a WSRF-capable WSDL parser to import services into a workflow.

## 4.2 Simplified Modeling with SimpleBPEL

Recently, ViGO has been extended to allow parts of workflows (“snippets”) to be saved in libraries. The newly developed SimpleBPEL Orchestrator allows us to import those libraries. Workflow developers may use the existing snippets as black boxes in their workflows. The tool checks whether snippets the user wants to combine in a workflow actually fit together. This is done by validating whether the output data of the first component fits to the input data of the succeeding component. If so, the tool automatically generates necessary BPEL code (such as *assign* operations) without the need for any action of the developer. Figure 6 illustrates a simple example of a workflow developed using SimpleBPEL and its actual representation in ViGO.

## 5 Related Work

Due to space restrictions, it is not possible to fully cover related work in all areas. We therefore only briefly present some exemplary related work.

Amnuaykanjanasin and Nupairoj [AN05] present a BPEL-based approach for orchestrating OGSI-based Globus Toolkit 3 Grid services using proxy services. Proxy services are

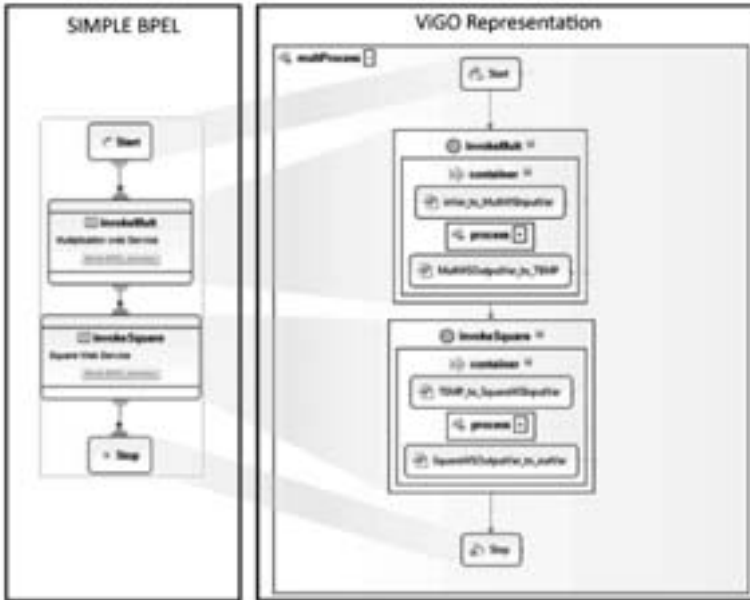


Figure 6: Workflow modeling with SimpleBPEL vs. modeling with ViGO

facade services that hide the Grid service's complexity and are invoked by the workflow engine instead of the original service. The call is then delegated to the Grid service. The approach supports security mechanisms of Globus Toolkit 3 security based on WS-Security. Despite the fact that the complexity of Grid environments is increased by this approach, the solution is interesting, since it allows the usage of security and notification features. However, lifetime management of proxy certificates is not addressed at all.

By introducing a new element (*find.bind*) into the BPEL, Karastoyanova et al. [KHC<sup>+</sup>05] have presented their approach for runtime adaptability. The mechanism is able to find services, e.g. by querying a UDDI registry. Based on policies, it selects suitable services and binds them to process instances. In case a service call fails, a process instance repair is guaranteed by rebinding to another port. Selection criteria can be modified at runtime.

Di Penta et al. [DEV<sup>+</sup>06] present WS Binder that allows the (re-) binding of *partnerLinks* to services during the runtime of a process. For this aim, the authors use a proxy architecture. *PartnerLinks* are bound to proxy services instead of the original target services, meaning that the workflows need to be adapted to run in the environment. If a failure occurs at runtime, the proxy services are rebound to target services determined by the framework's discovery and selection component.



## 6 Conclusion

In this paper, selected components of *BPEL4Grid* were presented: security aspects of workflow composition, fault tolerance mechanisms for long-running workflows, and corresponding workflow modeling tools. Future work will be devoted to extending the functionality of BPEL4Grid with respect to workflow analysis, validation, and monitoring.

## References

- [ACD<sup>+</sup>03] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and Sanjiva Weerawarana. *Business Process Execution Language for Web Services Version 1.1*, 1.1 edition, May 2003.
- [AN05] P. Amnuaykanjanasin and N. Nupairoj. The BPEL Orchestrating Framework for Secured Grid Services. In *ITCC (1)*, pages 348–353. IEEE, 2005.
- [DEV<sup>+</sup>06] M. Di Penta, R. Esposito, M. Villani, R. Codato, M. Colombo, and Elisabetta Di Nitto. WS Binder: a Framework to Enable Dynamic Binding of Composite Web Services. In *Proceedings of the 2006 Int. Workshop on Service-oriented Software Engineering*, pages 74–80. ACM, 2006.
- [DFH<sup>+</sup>07] T. Dörnemann, T. Friese, S. Herdt, E. Juhnke, and B. Freisleben. Grid Workflow Modelling Using Grid-Specific BPEL Extensions. In *Proceedings of German e-Science Conference (GES)*, pages 1–8, 2007.
- [DJF09] T. Dörnemann, E. Juhnke, and B. Freisleben. On-Demand Resource Provisioning for BPEL Workflows using Amazon’s Elastic Compute Cloud. In *Proceedings of the 9<sup>th</sup> IEEE Int. Symposium on Cluster Computing and the Grid*, pages 140–147. IEEE, 2009.
- [DMS<sup>+</sup>09] T. Dörnemann, M. Mathes, R. Schwarzkopf, E. Juhnke, and B. Freisleben. DAVO: A Domain-Adaptable, Visual BPEL4WS Orchestrator. In *Proceedings of the IEEE 23<sup>rd</sup> Int. Conference on Advanced Information Networking and Applications (AINA '09)*, pages 121–128. IEEE, 2009.
- [DSF08] T. Dörnemann, M. Smith, and B. Freisleben. Composition and Execution of Secure Workflows in WSRF-Grids. In *Proceedings of the 8<sup>th</sup> IEEE Int. Symposium on Cluster Computing and the Grid*, pages 122–129. IEEE, 2008.
- [ec2] Amazon Web Services LLC, Amazon Elastic Compute Cloud (EC2). <http://aws.amazon.com/ec2/>.
- [JDF09] E. Juhnke, T. Dörnemann, and B. Freisleben. Fault-Tolerant BPEL Workflow Execution via Cloud-Aware Recovery Policies. In *Proceedings of 35<sup>th</sup> EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, pages 31–38. IEEE, 2009.
- [KHC<sup>+</sup>05] D. Karastoyanova, A. Houspanossian, M. Cilia, F. Leymann, and A. Buchmann. Extending BPEL for Run Time Adaptability. In *EDOC '05: Proceedings of the Ninth IEEE Int. EDOC Enterprise Computing Conference*, pages 15–26. IEEE, 2005.
- [nim] Nimbus, Nimbus IaaS Cloud Computing Solution. <http://www.nimbusproject.org>.

# Ein modellgetriebener Ansatz zur Nutzung von WS-BPEL für Scientific Workflows

G. Scherp<sup>1</sup> und W. Hasselbring<sup>2</sup>

<sup>1</sup>OFFIS Institut für Informatik, Escherweg 2, 26121 Oldenburg

<sup>2</sup>Universität Kiel, Institut für Informatik, 24118 Kiel

**Abstract:** Scientific Workflows haben sich zunehmend in diversen wissenschaftlichen Disziplinen etabliert, um die Modellierung komplexer Prozessierungsketten und deren Ausführung in verteilten Infrastrukturen wie Grids zu unterstützen. Mit der Einführung von Service-orientierten Architekturen im betrieblichen Kontext und deren Adoption in Grid-Infrastrukturen, haben sich Scientific Workflows und Business Workflows auf der technischen Ausführungsebene stark angenähert. Daher ist eine in diesem Bereich etablierte und standardisierte Workflow-Sprache wie WS-BPEL technisch gesehen auch für die Ausführung von Scientific Workflows geeignet, allerdings weniger zur fachlichen Modellierung. In diesem Dokument beschreiben wir einen modellgetriebenen Ansatz in Form eines Transformationsframeworks, um aus einer Scientific Workflow-Beschreibung auf der fachlichen Ebene automatisiert einen ausführbaren Workflow auf Basis einer Business Workflow-Sprache wie WS-BPEL auf der technischen Ebene zu generieren.

## 1 Einleitung

Scientific Workflows haben sich zunehmend zur Beschreibung und Ausführung komplexer Prozessketten in unterschiedlichen wissenschaftlichen Bereichen wie Chemie, Biologie und Physik etabliert. Grid-Infrastrukturen sind auf Grund der üblicherweise großen Anzahl an Ressourcen geeignete Ausführungsumgebungen für Scientific Workflows. Dazu wurden zahlreiche Workflow-Sprachen und -Engines sowie Werkzeuge entwickelt. In der betrieblichen Informationsverarbeitung werden ebenfalls Workflows, im Folgenden als Business Workflows bezeichnet, eingesetzt. Durch die Etablierung von Service-orientierten Architekturen (SOA) sind in einem Business Workflow genutzte Anwendungen wie beispielsweise Datenbanken durch Dienste gekapselt. Letztlich wird die Ausführung eines Business Workflow zu einer so genannten Dienstorchestrierung (engl. *service orchestration*). Die XML-basierte *Web Services Business Process Execution Language* (WS-BPEL)<sup>1</sup> wurde speziell für die Orchestrierung von Diensten bzw. Web Services entwickelt und hat sich als OASIS-Standard im betrieblichen Umfeld etabliert. Ausgehend von der Definition der *Open Grid Service Architecture* (OGSA)<sup>2</sup> wurde das SOA-Konzept in Grid-Technologien übernommen. Sämtliche Zugriffe auf Ressourcen (Rechner, Speicher) sind in einer solchen Grid-Infrastruktur durch Dienste gekapselt. Dazu wurden sowohl eigene Standards

---

<sup>1</sup><http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

<sup>2</sup><http://www.globus.org/ogsa/>

wie das *Web Service Resource Framework* (WSRF)<sup>3</sup> entwickelt, das zustandsbehaftete Web Services ermöglicht, als auch bereits bestehende Web Services-Technologien wie WS-Addressing<sup>4</sup> verwendet. Der WSRF-Standard wurde in Grid-Middlewares wie Globus Toolkit 4<sup>5</sup> und UNICORE 6<sup>6</sup> implementiert. Somit sind technisch gesehen sowohl Business Workflows als auch Scientific Workflows Dienstorchestrierungen und WS-BPEL demnach zur Ausführung von Scientific Workflows geeignet.

## 2 WS-BPEL für Scientific Workflows

Der Begriff Business Workflow bzw. Workflow stammt ursprünglich aus dem betrieblichen Kontext und bezeichnet die vollständige oder teilweise technische Abbildung eines so genannten Geschäftsprozesses eines Unternehmens. Für Business Workflows gibt es anerkannte Definitionen und ein Referenzmodell von der *Workflow Management Coalition* (WfMC)<sup>7</sup> und es wurden entsprechende Standards entwickelt und etabliert. Für Scientific Workflows hingegen haben sich bisher weder eine einheitliche Definition noch Standards durchgesetzt. Eine zutreffende Beschreibung der Ziele von Scientific Workflows ist in [LWMB09] gegeben.

“The main goals of scientific workflows, then, are (i) to save ‘human cycles’ by enabling scientists to focus on domain-specific (science) aspects of their work, rather than dealing with complex data management and software issues; and (ii) to save machine cycles by optimizing workflow execution on available resources.”

Scientific Workflows werden dabei auf zwei Ebenen betrachtet. Erstens auf einer fachlichen Ebene, die die fachliche Modellierung eines Scientific Workflow durch einen Wissenschaftler fokussiert, und von technischen Details abstrahiert. Zweitens auf einer technischen Ebene, auf der ein ausführbarer Workflow vorliegt und auf den verfügbaren Ressourcen ausgeführt wird. Diese Trennung von fachlicher und technischer Ebene ist bei Business Workflows üblich. Beispielsweise kann für ein Business Workflow die fachliche Ebene durch die *Business Process Modelling Notation* (BPMN)<sup>8</sup> abgebildet werden und die technische Ebene durch WS-BPEL. Unterschiedliche Gründe motivieren diese Trennung. Sie ermöglicht die saubere Trennung von Fachlichkeit und IT und somit auch von unterschiedlichen Rollen innerhalb eines Unternehmens (Management und IT-Abteilung). Sie ermöglicht unterschiedliche Abstraktionsebenen für den jeweiligen Zweck. Das Management will sich beispielsweise nicht mit IT-technischen Details beschäftigen, sondern betrachtet einen Business Workflow auf einer fachlichen Ebene mit höherer Abstraktion, der von der IT-Abteilung auf der technischen Ebenen mit niedrigerer Abstraktion um-

---

<sup>3</sup><http://www.oasis-open.org/committees/wsrf/>

<sup>4</sup><http://www.w3.org/Submission/ws-addressing/>

<sup>5</sup><http://www.globus.org/toolkit/>

<sup>6</sup><http://www.unicore.eu>

<sup>7</sup><http://www.wfmc.org>

<sup>8</sup><http://www.bpmn.org/>

gesetzt wird. Üblicherweise sind produktiv eingesetzte Business Workflows sehr stabil, ändern sich selten und werden häufig ausgeführt [LR99]. Die zwangsläufig entstehende Lücke zwischen der fachlichen und technischen Ebene soll beispielsweise durch den Einsatz einer SOA verringert werden, damit Änderungen auf der fachlichen Ebene flexibler und schneller durch die technische Ebene umgesetzt werden können. Diese Trennung der beiden Ebenen macht auch für Scientific Workflows Sinn. Dabei muss allerdings beachtet werden, dass es mit dem Wissenschaftler nur eine Rolle gibt. Dieser erstellt einen Scientific Workflow, der auf “Knopfdruck” ausgeführt werden soll. Scientific Workflows ändern sich häufig, teilweise nach jeder Ausführung. Eine klare Trennung motiviert durch ein Rollenmodell analog zu Business Workflows gibt es bei Scientific Workflows daher nicht. Allerdings spielt der Abstraktionsgrad eine wichtige Rolle. Ein Wissenschaftler hat in der Regel wenig bis gar keine IT-Kenntnisse und benötigt wie in [LWMB09] motiviert eine Domänen-spezifische Abstraktion eines Scientific Workflow, quasi Daten und Prozessoren als grafische Boxen dargestellt und mit Linien verknüpft. Die technische Ausführung erfolgt automatisch im Hintergrund. Die Trennung der fachlichen und technischen Ebene macht von daher Sinn, dass auf jeder Ebene Technologien eingesetzt werden können, die den jeweiligen Zweck erfüllen. Dies setzt allerdings voraus, dass es für Scientific Workflows eine automatisierte Abbildung der fachlichen zur technischen Ebene gibt. Ansätze in diese Richtung gibt es sowohl für Business Workflows als auch für Scientific Workflows und werden kurz in den verwandten Arbeiten in Abschnitt 4 behandelt.

In der Einleitung wurde bereits motiviert, dass WS-BPEL technisch gesehen für die Ausführung von Scientific Workflows geeignet ist, also der technischen Ebene zugeordnet werden kann. Da WS-BPEL ein technischer Standard ist, eignet es sich nicht direkt für die Modellierung eines Scientific Workflows auf der fachlichen Ebenen. Generell enthält WS-BPEL keine direkte Fachlichkeit, es implementiert eine technische Service-Orchestrierung. Neben der technischen Eignung gibt es weitere Argumente für die Nutzung von WS-BPEL für Scientific Workflows. WS-BPEL ist ein etablierter OASIS-Standard und erfährt breite Unterstützung seitens der Industrie. Dadurch wird WS-BPEL durch zahlreiche existierende Workflow-Engines unterstützt. Des Weiteren wurde WS-BPEL für den Einsatz in SOA konzipiert. Infrastrukturen für Scientific Workflows wie Grids orientieren sich zunehmend am SOA-Konzept. Besteht somit die Ausführung eines Scientific Workflows nur noch aus Service-Aufrufen, so bietet sich für die Service-Orchestrierung die Nutzung einer etablierten und für diesen Zweck konzipierten Sprache an, statt evtl. bestehende Workflow-Engines aus dem Scientific Workflow-Bereich umzubauen. Zusammenfassend bietet sich mit WS-BPEL (oder anderen Business Workflow-Sprachen) eine technische Basis an, die sowohl für die Ausführung von Business Workflows als auch für die Ausführung von Scientific Workflows geeignet ist.

Zur Nutzung von WS-BPEL für die Ausführung von Scientific Workflows sind zwei technische Aspekte zu beachten. Erstens die Unterstützung des WSRF-Standards und zweitens die Unterstützung von Sicherheitsmechanismen in Grids. Die WSRF-Unterstützung betrifft zum Einen den Aufruf von WSRF-konformen Web Services. Da es sich dabei um eine bestimmte Reihenfolge von Web Service-Aufrufen handelt (Ressource erzeugen, nutzen und löschen), ist dies mit WS-BPEL umsetzbar. Wir haben dies mit den Grid-Middlewares Globus Toolkit 4 und UNICORE 6 getestet [SHG<sup>+</sup>10]. Zum Anderen

muss sich ein WS-BPEL-Prozess nach außen WSRF-konform verhalten, was prinzipiell möglich ist, aber die Komplexität eines WS-BPEL-Prozesses erheblich vergrößern würde [Ley06]. Sicherheitsaspekte sind im WS-BPEL-Standard nicht vorgesehen und müssen durch eine Workflow-Engine im Rahmen des Deployments unterstützt werden. Im Rahmen des D-Grid<sup>9</sup>-Projektes BIS-Grid<sup>10,11</sup> wird die so genannte BIS-Grid Workflow Engine<sup>12</sup> entwickelt, die ein “Grid-Wrapper” für eine beliebige WS-BPEL-Engine im Hintergrund darstellt. Dabei übernimmt die BIS-Grid Workflow Engine Sicherheitsaspekte des Grid sowie die WSRF-konforme Darstellung eines WS-BPEL-Prozesses. In diesem Szenario muss ein WS-BPEL-Prozess somit lediglich den Aufruf von WSRF-Web Services berücksichtigen. Im folgenden Abschnitt wird ein Transformationsframework vorgestellt, mit dem ein entsprechender WS-BPEL-Prozess generiert werden kann.

### 3 Transformationsframework

Unser Ansatz basiert darauf, dass ein Wissenschaftler einen Scientific Workflow auf der fachlichen Ebene modelliert, und dieser dann über ein Transformationsframework automatisch in eine technische und ausführbare Form übertragen wird, siehe Abbildung 1. Dabei soll die fachliche Ebene mit bestehenden Scientific Workflow-Technologien und die technische Ebenen mit bestehenden Business Workflow-Technologien umgesetzt werden. Die Verbindung dazwischen soll das durch Transformationsframework abgebildet werden. Prinzipiell soll das Transformationsframework so weit wie möglich unabhängig von der verwendeten Workflow-Technologie sein. Um das Transformationsframework zu testen haben wir uns auf WS-BPEL und die BIS-Grid Workflow Engine auf der technischen Seite festgelegt, die Workflow-Technologie auf der fachlichen Ebene ist derzeit noch offen. Letztlich sind mit dem Transformationsframework folgende Ziele verbunden:

- Verknüpfung von bestehenden Scientific Workflow-Technologien zur Modellierung von Scientific Workflows (fachliche Ebene) und bestehenden Business Workflows-Technologien zur Ausführung von Scientific Workflows (technische Ebene).
- Erweiterbarkeit um weitere Workflow-Sprachen sowohl auf der fachlichen als auch auf der technischen Ebene.
- Generierung eines ausführbaren Scientific Workflow auf Basis einer Business Workflow-Sprache. Komplexe Sprachkonstrukte werden automatisch erzeugt, deren manuelle Erstellung oft mühsam und fehleranfällig ist. Dazu gehören beispielsweise auch Konstrukte zur Fehlerbehandlung, die in der Regel auf der fachlichen Ebene nicht modelliert werden (eher durch Konfiguration einstellbar).

---

<sup>9</sup><http://www.d-grid.de>

<sup>10</sup><http://www.bisgrid.de>

<sup>11</sup>Dieses Projekt wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01IG07005 gefördert und ist Teil der D-Grid-Initiative.

<sup>12</sup><http://bis-grid.sourceforge.net>

- Nutzung von etablierten Technologien aus der modellgetriebenen Softwareentwicklung.

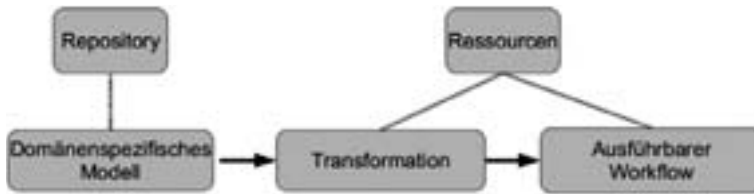


Abbildung 1: Allgemeiner Ansatz

Um eine automatische Abbildung von der fachlichen zur technischen Ebene zu gewährleisten, gehen wir davon aus, dass auf der fachlichen Ebene zumindest grundlegende technische Informationen vorhanden sind, allerdings vor dem Wissenschaftler versteckt. Dazu verwendet der Wissenschaftler zur Modellierung eines Scientific Workflows ein Repository, indem beispielsweise verfügbare Datenquellen, Datensinken und Prozessoren hinterlegt sind. Diese Elemente können über eine einfache grafische Notation verknüpft werden. Hinter jedem Element verbergen sich aber technische Informationen, beispielsweise über Daten und Dienste, die zur konkreten Abbildung auf bestimmte Ressourcen genutzt werden können. Jede technische Information zu einem Element kann dabei konfigurierbare Bereiche enthalten, die als Parameter vom Wissenschaftler modifiziert werden können, beispielsweise um gewünschte Daten zur Prozessierung einzugrenzen. Das Transformationsframework erzeugt somit Quelltext, der zum Einen die generelle Workflow-Ausführung abbildet und zum Anderen das so genannte Binden (engl. *binding*) an konkrete Ressourcen beinhaltet. Das Binden an eine Ressource bzw. einen Dienst ist ein Kernkonzept von SOA und kann statisch schon während des Transformationsprozesses für eine bestimmte Ressource durchgeführt werden, oder es wird Quelltext generiert, der eine dynamische Bindung zur Laufzeit vorsieht. In beiden Fällen muss ein Verzeichnis über verfügbare Ressourcen vorhanden sein. Bei der Auswahl einer Workflow-Technologie für die fachliche Ebene im Rahmen unserer geplanten Tests spielt die Unterstützung eines entsprechenden Repositories eine zentrale Rolle.

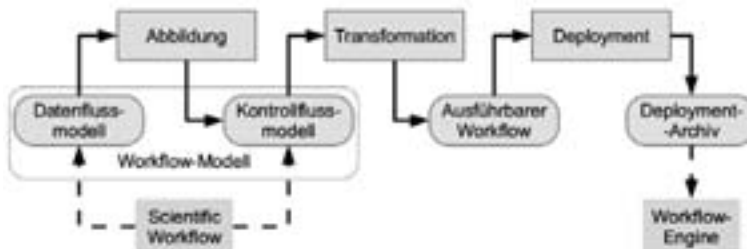


Abbildung 2: Architektur des Transformationsframeworks

Der Transformationsprozess des Transformationsframeworks ist in Abbildung 2 dargestellt. In einem ersten Schritt wird ein Scientific Workflow, der auf Basis einer bestehenden

Scientific Workflow-Sprache vorliegt, in ein internes Workflow-Modell abgebildet. Diese interne Workflow-Modell wird benötigt, um unabhängig von einer bestimmten Scientific Workflow-Sprache zu sein und kapselt sowohl den Datenfluss als auch den Kontrollfluss eines Scientific Workflows. Da Scientific Workflows in der Regel Datenfluss-orientiert sind und in geringem Umfang Kontrollfluss-Elemente zulassen, wird bei der Abbildung in das interne Workflow-Modell der Kontrollfluss nicht vollständig sein. Wenn beispielsweise ein Prozessor Eingangsdaten aus zwei Datenquellen benötigt, so ist in der Regel nicht festgelegt, in welcher Reihenfolge diese Datentransfers durchgeführt sollen oder parallel. Daher muss in dem Kontrollflussmodell die Aufrufreihenfolge von Datentransferdiensten in Form eigenständiger Aktivitäten berücksichtigt werden. Welche Aufrufreihenfolge nachher gewählt wird, kann u. a. von Performanzgründen abhängig sein, was hier nicht weiter behandelt wird. Um diese Erweiterung in dem Kontrollflussmodell auf Basis des Datenflussmodells vorzunehmen, ist der Schritt "Abbildung", siehe Abbildung 2 vorgesehen. Es ist derzeit noch offen, ob das Workflow-Modell aus zwei getrennten Modellen für Kontroll- und Datenfluss besteht, oder ein gesamte hybrides Modell verwendet wird. Des weiteren ist derzeit noch offen, welche Kontroll- und Datenflusselemente unterstützt werden.

Im nächsten Schritt wird das Workflow-Modell in einen ausführbaren Workflow transformiert, siehe "Transformation" in Abbildung 2. Da als Zielsprache Business Workflow-Sprachen genutzt werden, die üblicherweise Kontrollfluss-orientiert sind, wird der Transformationsprozess auf Basis des Kontrollflussmodells bzw. der Kontrollfluss-orientierten Sicht des Workflow-Modells durchgeführt. Dieser Schritt besteht aus zwei Unterschritten. In einem ersten Unterschritt wird der Kontrollfluss erweitert, in dem typische Aktivitäten eines Scientific Workflows um entsprechende Aktivitätsmuster erweitert werden. Die Aktivität zur Einreichung eines Jobs wird beispielsweise durch die Aktivitäten *Prepare*, *Submit*, *Wait*, und *Cleanup* erweitert. Diese Erweiterungen sind unabhängig von der Zielsprache und werden solange fortgeführt, bis keine Aktivitäten mehr erweitert werden können. Im zweiten Unterschritt wird der so erweiterte Kontrollfluss in die Zielsprache übertragen. Im Prinzip wird dabei jede Aktivität durch einen entsprechenden Quelltext-Abschnitt passend zur Ausführungsumgebung ersetzt. Beispielsweise kann für die *Wait*-Aktivität bei der Job-Einreichung Quelltext für ein *Push*-Modell (Benachrichtigungsdienst nutzen; Globus Toolkit 4) oder *Pull*-Modell (Periodische Statusabfrage; UNICORE 6) generiert werden. Am Ende des zweiten Unterschritts liegt der Scientific Workflow in einer ausführbaren Form vor. Sämtliche im Transformationsprozess angewendeten Transformationsregeln oder Templates sind nach unterschiedlichen Aspekten von Scientific Workflows strukturiert. Dazu gehören allgemeine Aspekte wie Fehlerbehandlung und Überwachung einer Workflow-Ausführung, sowie spezifische Aspekte wie die Einreichung eines Jobs oder die Ausführung eines Datentransfers. Dadurch können Anpassungen und Erweiterungen am Transformationsprozess gezielter und einfacher vorgenommen werden.

Im letzten Schritt "Deployment", siehe Abbildung 2, wird der ausführbare Workflow in einem Deployment-Archiv verpackt. Der Aufbau eines Deployment-Archivs ist abhängig vom Hersteller einer Workflow-Engine. Anschließend wird der ausführbare Workflow mittels des Deployment-Archives über die entsprechende Workflow-Engine bereitgestellt und der Scientific Workflow kann ausgeführt werden.

Für die prototypische Implementierung Transformationsframeworks wird das Eclipse Modeling Framework (EMF)<sup>13</sup> verwendet. Das Workflow-Modell und der ausführbare Workflow werden durch Ecore-Modelle repräsentiert. Eine Modell-zu-Modell-Transformation (M2M) beschreibt die Transformation von einem Ecore-Modell zu einem Ecore-Modell, z. B. mit der Query View Transformation (QVT)<sup>14</sup> or ATLAS Transformation Language (ATL)<sup>15</sup>. Eine Modell-zu-Text-Transformation beschreibt die Transformation eines Ecore-Modells zu Quelltext, z. B. mit Xpand<sup>16</sup>. Der Schritt “Abbildung” ist somit eine M2M-Transformation, im Schritt “Transformation” ist der erste Unterschnitt ebenfalls eine M2M-Transformation und der zweite Unterschnitt eine M2T-Transformation.

## 4 Verwandte Arbeiten

Es gibt zahlreiche bestehende Lösungen wie GWES, Kepler, Taverna, Triana usw., die Scientific Workflows bereits erfolgreich umsetzen<sup>17</sup>, aber hier nicht umfassend betrachtet werden können. Unser Ansatz soll kein Ersatz dieser Lösungen darstellen, sondern eine Ergänzung im besonderen Fokus auf der Verwendung von Business Workflow-Technologien zur Ausführung von Scientific Workflows in Service-orientierten Grids.

Die Eignung von WS-BPEL (bzw. dessen Vorgänger BPEL4WS) zur Ausführung von Scientific Workflows in Service-orientierten Grids wird durch zahlreiche Veröffentlichungen belegt, beispielsweise [WHH05, WEB<sup>+</sup>06, DFH<sup>+</sup>07]. In einigen Ansätzen wie [WHH05, DFH<sup>+</sup>07] werden dazu Erweiterungen für WS-BPEL vorgeschlagen. Damit wird ein Sprachdialekt erzeugt, wodurch eine eigene Workflow-Engine entwickelt werden und zukünftige WS-BPEL-Versionen selbst übernommen werden müssen. In unserem Ansatz wird daher ausschließlich Standard-WS-BPEL verwendet. Teilweise sind auf WS-BPEL basierende Werkzeuge entwickelt worden [WEB<sup>+</sup>06, DFH<sup>+</sup>07], die zur Modellierung bzw. Implementierung von Scientific Workflows eingesetzt werden. Diese Werkzeuge sind für Wissenschaftler ohne IT-Kenntnisse nicht geeignet. Daher setzen wir in unserem Ansatz auf eine fachliche Abstraktion mit bestehenden Scientific Workflow-Technologien.

Ansätze zur Abbildung der fachlichen zur technischen Ebene auf Basis von modellgetriebenen Technologien gibt es im Scientific Workflow-Bereich bisher kaum. In [DBG<sup>+</sup>03] wird eine eigene Code-Generierung verwendet, allerdings mit Fokus auf Workflow-Optimierung. Das in [WEB<sup>+</sup>06] beschriebene Werkzeug bietet mit so genannten *Plugins* und *Macros* eine Art Template-Mechanismus zur Code-Generierung an. Im Business Workflow-Bereich wird diese Abbildung stärker fokussiert, z. B. in [ODtHvdA06, YZZ<sup>+</sup>07]. Allerdings handelt es sich dabei um einen Top-Down-Softwareentwicklungsprozess, der schwer automatisierbar ist. Die hier betrachteten Scientific Workflows hingegen stellen auf der fachlichen Ebene eine Komposition aus definierten Elementen eines Repositories dar, woraus ein ausführbarer Workflow generiert wird.

---

<sup>13</sup><http://www.eclipse.org/modeling/emf/>

<sup>14</sup><http://www.omg.org/spec/QVT/1.0/>

<sup>15</sup><http://www.eclipse.org/m2m/atl/>

<sup>16</sup><http://wiki.eclipse.org/Xpand>

<sup>17</sup>Siehe auch <http://www.gridworkflow.org/>.



## 5 Fazit

In diesem Dokument wurde die Trennung von fachlicher und technischer Ebene bei Scientific Workflows sowie der Einsatz von etablierten und standardisierten Business Workflow-Technologien wie WS-BPEL auf der technischen Ebene motiviert. Zur Verbindung der fachlichen und technischen Ebenen wurde ein erweiterbares und auf modellgetriebenen Technologien aufbauendes Transformationsframework vorgestellt, das eine Scientific Workflow-Beschreibung auf der fachlichen Ebene in einen ausführbaren Workflow auf der technischen Ebene transformiert. Ein Prototyp ist auf Basis von EMF geplant.

## Literatur

- [DBG<sup>+</sup>03] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, Kent Blackburn, Albert Lazzarini, Adam Arbre, Richard Cavanaugh und Scott Koranda. Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*, 1(1):25–39, March 2003.
- [DFH<sup>+</sup>07] Tim Dörnemann, Thomas Frieze, Sergej Herdt, Ernst Juhnke und Bernd Freisleben. Grid Workflow Modelling Using Grid-Specific BPEL Extensions. 2007.
- [Ley06] Frank Leymann. Choreography for the Grid: towards fitting BPEL to the resource framework. *Concurr. Comput. : Pract. Exper.*, 18(10):1201–1217, 2006.
- [LR99] Frank Leymann und Dieter Roller. *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, September 1999.
- [LWMB09] Bertram Ludäscher, Mathias Weske, Timothy McPhillips und Shawn Bowers. Scientific Workflows: Business as Usual? In Umeshwar Dayal, Johann Eder, Jana Koehler und Hajo Reijers, Hrsg., *7th Intl. Conf. on Business Process Management (BPM)*, LNCS 5701, Ulm, Germany, 2009.
- [ODtHvdA06] Chun Ouyang, Marlon Dumas, Arthur H. M. ter Hofstede und Wil M. P. van der Aalst. From BPMN Process Models to BPEL Web Services. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, Seiten 285–292, Washington, DC, USA, 2006. IEEE Computer Society.
- [SHG<sup>+</sup>10] Guido Scherp, André Höing, Stefan Gudenkauf, Wilhelm Hasselbring und Odej Kao. Using UNICORE and WS-BPEL for Scientific Workflow Execution in Grid Environments. In *Euro-Par 2009 Workshops - Parallel Processing*, Jgg. Lecture Notes in Computer Science of *Lecture Notes in Computer Science*. Springer, 2010.
- [WEB<sup>+</sup>06] Bruno Wassermann, Wolfgang Emmerich, Ben Butchart, Nick Cameron, Liang chen und Jignesh Patel. *Sedna: A BPEL-Based Environment for Visual Scientific Workflow Modeling*, Kapitel 0, Seiten 427–448. Springer, 2006.
- [WHH05] Yong Wang, Chunming Hu und Jinpeng Huai. A New Grid Workflow Description Language. In *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, Seiten 257–260, Washington, DC, USA, 2005. IEEE.
- [YZZ<sup>+</sup>07] Xiaofeng Yu, Yan Zhang, Tian Zhang, Linzhang Wang, Jianhua Zhao, Guoliang Zheng und Xuandong Li. Towards a Model Driven Approach to Automatic BPEL Generation. In *ECMDA-FA*, Seiten 204–218, 2007.

# The Missing Features of Workflow Systems for Scientific Computations

Mirko Sonntag, Dimka Karastoyanova, Frank Leymann

Institute of Architecture of Application Systems  
University of Stuttgart  
Universitaetsstrasse 38  
70569 Stuttgart, Germany  
{sonntag, karastoyanova, leymann}@iaas.uni-stuttgart.de

**Abstract:** This paper discusses technical aspects of how business workflow management systems can be improved in order to apply them in the field of scientific workflows and reap all their benefits. We give recommendations how to address the discovered gaps in support for scientific applications. The approach we follow addresses the requirements of scientists and scientific applications, which we also identify in this work.

## 1 Introduction

There are two main application areas driving and utilizing workflows – businesses and scientific computations and experimenting. Since the early 90s, workflow management systems (WfMSs) are applied by enterprises to support their business. Over the years, the workflow technology matured and is nowadays established and well-proven in the business area. Business WfMSs are universally designed independent of the concrete business area of employing enterprises. Because of the generic approach the workflow technology follows, typically lots of configuration options for such systems exist which contributes to their complexity. Therefore it is commonplace for IT experts to implement business processes of enterprises and to set up the software infrastructure. Business workflows often represent the products of enterprises (e.g. a loan approval workflow of a bank stands for the product “loan”) [LR00].

In recent years, workflows gained more and more attention in science to support scientists in their work. Scientific WfMSs are not built using the existing workflow technology but are designed and developed from scratch. The reason can be found in the very different requirements of scientific workflows compared to their business counterpart [Gi07]. Scientific WfMSs are often tailored to a particular application domain. In this context, workflows implement scientific simulations, experiments, and computations typically dealing with huge amounts of data. Scientists model, execute, monitor, and analyze workflows. Since they are no IT experts, special attention is paid on the usability aspects of the systems.

Since recently there are endeavors to join the efforts of these two communities. To be more precise, efforts were made to harness the traditional workflow technology for scientific workflows [BG07, Wa07, Ba08, So10]. This approach is promising since it brings the strengths of the technology to scientific applications. Despite their generic properties, business WfMSs are not yet capable of covering some of the challenges arising because of particular needs of scientists and specific existing software and hardware infrastructures. A full-fledged Grid support, for example, is crucial since many scientific applications are conducted in a Grid environment as they depend on large storage resources and computational power a single scientific institute can hardly afford.

To the best of our knowledge, there is no systematic discussion and comparison of the scientific workflow systems and workflow systems used widely in business applications. Moreover, there are only a few research works that report on how the conventional workflow technology needs to be improved in order to accommodate all the requirements of scientist.

This paper provides a more technical view on problems coming up when employing traditional workflow technology in the scientific area. The considered technical aspects are thereby explained and compared to existing scientific WfMSs by exposing similarities and differences. Another main contribution is a set of recommendations about how the discussed barriers can be overcome. The reader is given a means at hand to understand the challenges of building a scientific WfMS on top of the traditional workflow technology.

The rest of the paper is structured as follows: Section 2 briefly discusses related work in the field of investigating scientific workflows and existing scientific workflow systems. Section 3 presents the technical consideration of business WfMS concepts and their application to scientific workflow management (WfM). Finally, Section 4 concludes the paper and gives directions for future work.

## **2 Related work**

In [Gi07] it is investigated which specific requirements scientific workflows impose on a supporting infrastructure. The discussion is held on a system level considering various functional (e.g. reproducibility, automatic adaptation) and non-functional properties (e.g. stability, usability). As opposed to this, our approach is discussing technical aspects of business WfMSs and their applicability in the area of scientific WfM.

In [BG07] typical characteristics of business workflows are explained (e.g. transactions, fault tolerance, use of standards) and their benefits for scientific workflows are presented. Moreover, features unique to scientific workflows are identified (e.g. evolving nature, non-computer experts as workflow designers, data centrality). In [Lu09] an inverse approach is followed: common features of scientific workflows (e.g. data flow orientation) are collected and compared to business workflows. However, both papers argue on the workflow model and language level. In contrast to that, we compare business and scientific WfMSs and their underlying technical concepts.

The considerations in this paper are based on requirements of scientists on scientific WfMSs. These requirements are both gathered out of the mentioned works and derived from several scientific WfMSs: Kepler [Al04], Taverna [Oi06], and Triana [Ch05] are scientific workflow systems that are able to deal with Web services (WS); Pegasus [De04] and SEGL [Cu08] are workflow systems specialized for the use in a Grid environment; Sedna [Wa07] is a workflow modeler based on the conventional workflow technology; Trident [Ba08] and e-BioFlow [Wa09] are systems based on business WfMSs.

### 3 Fitting conventional workflow systems for scientific computations

Currently, there are efforts to establish the conventional workflow technology in the scientific domain [Ba08, So10, Wa09, Wa07]. This promises to bring along a number of advantages: simplified collaboration of different scientific groups through the use of technology standards, already existing tools that can be used as development basis, or a higher robustness compared to existing scientific workflow systems, to name just a few. On the other hand, a lot of problems come up when trying to use traditional WfMSs in science because of very indifferent requirements by the users and the employed infrastructures. Business WfMSs are rather general systems since they are intended to serve enterprises independent of their business model and hardware infrastructure. Furthermore, several types of workflows can be handled by one and the same WfMS (production, administrative, collaborative, and ad hoc) [LR00]. This yields a very high complexity of the systems (e.g. number of tools, involved user roles, configuration options) and heavy IT support is needed to translate business processes into a machine-readable form and to customize the software.

Instead of making use of existing and established solutions scientific WfMSs are often built from scratch. This is because the systems are tailored to the needs of scientists [Gi07] and to the respective operation area. Scientists as non-computer experts require, for example, high usability of the software. Scientific workflows are often operated in a specific scientific domain (e.g. e-BioFlow is preferential for life science workflows). This is mainly manifested in the tools built-in in the workflow editor to solve a certain scientific problem in a domain specific language. There are also scientific WfMSs particularly designed for the operation in Grid environments (e.g. Pegasus, SEGL).

In the following, we dig deeper into the issue of investigating the applicability of business WfMSs in the scientific area. To do so, we account for a number of technical aspects of business WfMSs, analyze them, and give recommendations about how to extend or adapt them to meet the requirements of scientists.

**Life cycle and tool integration.** In business process management (BPM), a very well-known subject is the life cycle of business processes containing several separated and repeatable management phases as shown in Figure 1. Naturally, each phase is conducted by a particular user role with specific tasks, knowledge, capabilities, and working methods. To account for this situation the BPM life cycle is typically supported by many different tools with a complex interplay, e.g. an editor for workflow modeling, or an

engine for workflow execution. As opposed to this, scientific WfMSs often consist of a single tool (e.g. Kepler, Triana, Taverna). This is due to the fact that scientists are typically the only user group of a system. When employing a business WfMS for the use in the scientific domain, the tool portfolio needs to be thinned out and/or integrated so as to be experienced as a single tool for scientists. Such an integrated tool should hide the complexity of the underlying software infrastructure for modeling, configuration, execution, monitoring, and analysis of workflows.

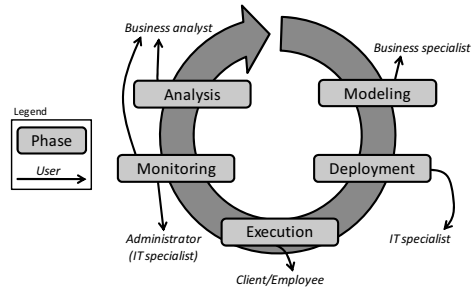


Figure 1: BPM life cycle

**Modeling and deployment.** An important aspect in business WfM is the deployment of process models on a workflow engine. Deployment is a technical step to put process models into production. It requires deep knowledge of the used infrastructure (e.g. workflow engine, web server) and technology (e.g. WSS, partner links in BPEL [AI07]). The purpose of deployment is manifold: translation of process models into a format optimized for execution (typically held in a database); external configuration of processes increases their reusability in different contexts; installation of processes so that they can be frequently executed even late after modeling; providing processes to customers (e.g. as WS). In contrast, most scientific workflows are simply executed without deployment. The execution code of a workflow is already known on workflow editor level making a translation step unnecessary. Instance state is held in main memory only. Configuration is contained directly in the workflow definition. That means to change its configuration (e.g. bind another resource for a task) the workflow itself has to be changed (adapted in the terminology of conventional workflows). Scientific workflows are typically executed by scientists themselves immediately after modeling or even during modeling in a trial-and-error manner [Wa09]. Workflows are mainly not provided to be invoked by third parties (e.g. WSS are only called synchronously so that no callback is needed). An exception is Triana where a workflow can be explicitly deployed and provided as WS. Nevertheless, if a business WfMS needs to be used in the scientific domain, the deployment step should be kept due to the many advantages it yields, such as reusability of workflow parts, efficient execution with the help of an optimized format, robustness when storing instance state in a database. Since on the other hand scientists are no computer experts and are neither able to nor want to cope with the complexity of a deployment step, we propose to hide the deployment. An option is to conduct it transparently for the scientist as part of a “run workflow” operation. Information needed for deployment could be derived from workflow model properties (e.g. in case of an installation as WS the workflow model’s name combined with an activity name could act as service name and could be part of the port).

**Execution.** By joining the Grid technology and Service-Oriented Architectures (SOA) with the help of the Open Grid Services Architecture (OGSA) and the Web Services Resource Framework (WSRF) the first steps towards Grid-awareness in business

WfMSs are taken. WfMSs that make use of the WS technology can now access Grid resources offered as stateful WSs [Sl06]. Since Grid support goes beyond conventional WS invocations, an Enterprise Service Bus (ESB) can be used to keep track of all resources on the Grid, match functional properties, negotiate policies, and eventually find and bind appropriate services/resources on behalf of a workflow [We05]. However, Grid-enabled workflow systems in science (e.g. Pegasus, SEGL) implement additional concepts: e.g. searching for idle resources, code and data shipping/staging, job scheduling, authentication, authorization, or credential delegation, which are not issues solved by conventional workflow systems. For a full-fledged Grid support an ESB must be extended to support such concepts. The incorporation of existing Grid middleware services could simplify this exercise.

**Monitoring and analysis.** Monitoring of business processes is crucial to observe the system state, and to visualize business or performance data on the workflow level with the help of tables and diagrams, for instance. Monitoring is highly customizable and hence a very complex part of WfMSs. Typically lots of instances of a process model are monitored simultaneously because business analysts are mainly interested in aggregated results. In scientific WfM, monitoring of single instances is more important in order to allow scientists to follow the progress of their computations. In existing scientific systems, customizing monitoring is very restricted if possible at all. Adopting business monitoring features in the scientific domain is beneficial due to powerful methods for observing workflow runs, user notifications, and customizability. Nevertheless, the concepts need extensions to fit scientific needs. Monitoring should be focused on single instances and geared towards the workflow model graph structure. Although Oracle's process server [Ora], for example, allows the user to inspect running instances in a graph-based manner, this is done by a particular Web interface instead of an integrated view in the workflow modeling tool. The experience of having the functionality of the workflow system available at one place is not given. Furthermore, the level of monitoring needs to be widened: in a Grid environment, information about used services, underlying operating systems and hardware are required for a coherent view on the overall system [FK04, Mi09]. In business WfM, it is important *that* a service keeps the promised functional and non-functional properties. It is not important *how* a service achieved its aims in terms of used resources, software, and tools. Quite the contrary, this information can even be the business model of enterprises and hence their secret. Therefore, a business WfMS and the utilized infrastructure need to be extended for the use in the scientific domain. The used resources need to provide an interface to reveal current resource properties, such as size of the job queue, installed operating system, number and load of processors. This can be achieved with the help of WSRF. Since WSRF is only a general framework that enables querying the state of a resource, it is up to the resource which properties it is willing to reveal. That means, the incorporated resources need an interface to allow a monitor querying at least an agreed upon number of properties.

**Flexibility.** In conventional workflow technology, flexibility mechanisms, such as run time adaptation concepts, are investigated to a great extend [RD98, Ka06]. Well-known approaches are, amongst others, to insert or delete an activity, to reiterate parts of a workflow, or to inquire a second opinion to finish an activity [LR00]. Although such

features promise to support scientists in conducting experiments in a trial-and-error manner, they are currently almost unaddressed in the scientific community. An exception is the e-BioFlow system [Wa09] where an ad hoc editor enables scientists to execute and re-execute incomplete workflows (so-called workflow fragments [Eb09]). However, when adopting the concepts in the scientific area, carefully designed extensions are needed on the functional and non-functional level. For instance, modifications of workflows need to be thoroughly tracked for the sake of reproducibility. This is beyond the current scope of auditing information that mainly track events raised during workflow execution. Moreover, the application of adaptation and flexibility concepts in a Grid environment with stateful resources needs to be examined. Consider, for example, the workflow in Figure 2(a) that invokes a program A on resource A. The program creates data that resides on the resource. Later on, the workflow invokes program B that is also located on resource A and that relies on the data produced by program A. Figure 2(b) visualizes the same scenario but the workflow was adapted after the execution of activity 1. Activity 2 now invokes program B on resource B. That means the appropriate data has to be shipped from resource A to B or else program B would raise an error. One could imagine several approaches to solve these kinds of problems, e.g. a validation component of the modeling tool could prohibit such modifications, or the ESB could do the data shipping transparently for the scientist. The situation gets even more complex if the resources were bound lately. A workflow language that allows specifying data dependencies between one or more activities (or even tools) could simplify the solution of this specific problem. Unfortunately, most business WfMSs are control flow-oriented and thus need an appropriate extension.

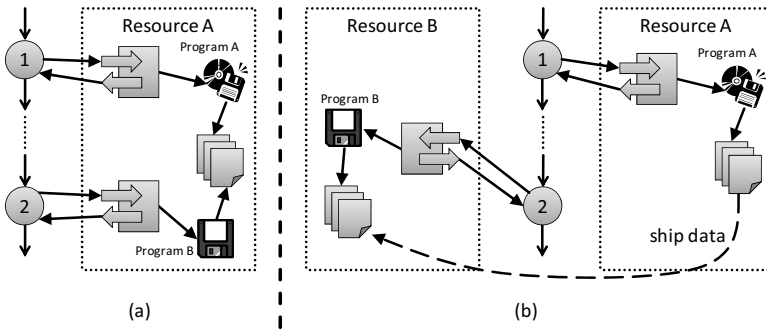


Figure 2: Workflow adaptation in a situation with stateful services

**Provenance.** For legal or analysis reasons business WfMSs track information about workflow execution on the workflow instance level in an audit trail. Scientists require provenance information to reproduce their results (i.e., obtaining similar results with the same initial data). Auditing data can contribute to a provenance record but are insufficient for an accurate reproducibility. Information beyond the workflow instance level is needed, such as concrete services found, bound, and invoked by an ESB. Provenance tracking is challenging especially in highly dynamic environments as in scientific computing (e.g. resources may come and go, workflows are modified at run time). That means, the audit trail must be at least extended by ESB events that are expressive enough to follow the search, selection, and binding of resources.

## 4 Conclusion and outlook

This work discussed the technical characteristics of business WfMSs and evaluated them for the purpose of applying them for modeling and execution of scientific applications for simulations, experiments, and computations. Based on this analysis we identified a number of missing features of conventional WfMSs:

- That fact that business WfMSs consist of several, not integrated tools impedes usability for scientists.
- An explicit deployment step requires deep knowledge of the underlying technology (e.g. BPEL's partner link concept, WSs) and the employed infrastructure (e.g. workflow engine, web application server) and is hence hard to accomplish by a non-computer expert.
- Typically, workflows are started by incoming messages or with the help of additional workflow clients. Scientists need to start workflows from within the modeling tool.
- Monitoring is focused on aggregated statistical information on a workflow level over several workflow instances. Scientists require a graph-oriented monitoring of single workflow instances as well as monitoring on additional levels, such as employed tools, operating system and hardware.
- Grid-awareness is uncommon in current ESB implementations [IBM] [ASM] (e.g. searching for idle resources, code and data shipping, job scheduling, authentication, authorization, or credential delegation).
- A seamless trial-and-error workflow design as often required by scientists is usually not supported by conventional WfMSs. Flexibility mechanisms in stateful Grid environments impose additional yet unaddressed requirements.
- Audit trails store information on a workflow instance level and thus do not provide a provenance tracking mechanism that ensures confidence in and reproducibility of scientific results. Especially, provenance tracking in flexible scientific environments is challenging.

We provided recommendations as of how the identified drawbacks can be addressed. Based on the results of this work we aim at implementing a scientific WfMS built on top of an existing open-source workflow system, a modeling tool, and a service bus. Special attention will be paid to the usability, Grid-awareness, monitoring, flexibility, and data-centrality of the system.

**Acknowledgements.** The work presented in this paper has been funded by the DFG Cluster of Excellence Simulation Technology<sup>1</sup> (EXC310).

## References

- [A104] Altintas, I. et al.: "Kepler: An Extensible System for Design and Execution of Scientific Workflows". In: *Int'l Conf. on Scientific and Statistical Database Management*, 2004.

---

<sup>1</sup> <http://www.simtech.uni-stuttgart.de>



- [Al07] Alves, A. et al.: Web Services Business Process Execution Language (BPEL) Version 2.0, OASIS standard, April 11<sup>th</sup>, 2007.
- [ASM] Apache ServiceMix. [Online]  
<http://servicemix.apache.org/home.html> [February 17<sup>th</sup>, 2010]
- [Ba08] Barga, R. et al.: “The Trident Scientific Workflow Workbench”. In: *IEEE International Conference on eScience*, 2008.
- [BG07] Barga, R.; Gannon, D.: “Scientific versus Business Workflows”. In: Taylor et al. (Eds.), *Workflows for e-Science: Scientific Workflows for Grids*, Springer, 2007.
- [Ch05] Churches, D. et al.: “Programming Scientific and Distributed Workflow with Triana Services”. In: *Concurrency and Computation: Practice and Experience*. Special Issue on Scientific Workflows, 2005.
- [Cu08] Currie-Linde, N. et al.: “SEGL: A problem solving environment for the design and execution of complex scientific Grid applications”. In: *3rd Russian-German Advanced Research Workshop*, Springer, 2008.
- [De04] Deelman, E. et al.: “Pegasus: Mapping Scientific Workflows onto the Grid”. In: *2<sup>nd</sup> European AcrossGrids Conference*, Springer, 2004, pp. 11-20.
- [Eb09] Eberle, H. et al.: “Workflow Fragments”. In: OTM Part I, 2009.
- [FK04] Foster, I.; Kesselmann, C.: “The Grid 2: Blueprint for a New Computing Infrastructure”. Morgan Kaufmann, 2004.
- [Gi07] Gil, Y. et al.: “Examining the Challenges of Scientific Workflows”. *IEEE Computer*, 40(12), 2007.
- [IBM] IBM: WebSphere Enterprise Service Bus. [Online]  
<http://servicemix.apache.org/home.html> [February 17<sup>th</sup>, 2010]
- [Ka06] Karastoyanova, D.: “Enhancing Flexibility and Reusability of Web Service Flows through Parameterization”. PhD thesis, TU Darmstadt and University of Stuttgart, 2006.
- [LR00] Leymann, F.; Roller, D.: “Production Workflow: Concepts and Techniques”. Prentice Hall, 2000.
- [Lu09] Ludaescher, B. et al.: “Scientific Workflows: Business as Usual?” In: *7<sup>th</sup> International Conference on Business Process Management (BPM)*, 2009.
- [Mi09] Mietzner, R. et al.: “Business Grid: Combining Web Services and the Grid”. In: *Transactions on Petri Nets and Other Models of Concurrency (ToPNoC)*, Special Issue on Concurrency in Process-aware Information Systems, Springer Verlag, 2009.
- [Oi06] Oinn, T. et al.: “Taverna: Lessons in Creating a Workflow Environment for the Life Sciences”. In: *Concurrency and Computation: Practice and Experience*, 2006.
- [Ora] Oracle: Oracle BPEL process manager. [Online]  
[http://www.oracle.com/appserver/bpel\\_home.html](http://www.oracle.com/appserver/bpel_home.html) [February 17<sup>th</sup>, 2010]
- [RD98] Reichert, M.; Dadam, P.: “ADEPT<sub>flex</sub> – Supporting Dynamic Changes of Workflows Without Losing Control”. In: *Journal of Intelligent Information Systems*, 10(2), 1998.
- [Sl06] Slomski, A.: “On using BPEL extensibility to implement OGSF and WSRF Grid workflows”. In: *Concurrency and Computation: Practice and Experience*, vol. 18 (10), pages 1229-1241, 2006.
- [So10] Sonntag, M. et al.: “Process Space-Based Scientific Workflow Enactment”. In: *Special Issue on Scientific Workflows in the International Journal of Business Process Integration and Management (IJBPIIM)*, Inderscience Publishers, 2010. (to appear)
- [Wa07] Wassermann, B. et al.: “Sedna: A BPEL-based Environment for Visual Scientific Workflow Modeling”. In: Taylor et al. (Eds.): *Workflows for e-Science: Scientific Workflows for Grids*, Springer, 2007.
- [Wa09] Wassink, I. et al.: “Designing workflows on the fly using e-BioFlow”. In: *7<sup>th</sup> Conference on Service Computing (ICSSOC)*, 2009.
- [We05] Weerawarana, S. et al.: “Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More”. Prentice Hall, 2005.



**2nd European Workshop on Patterns for Enterprise  
Architecture Management**

**(PEAM)**



# Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes

Helge Buckow, Hans-Jürgen Groß, Gunther Piller,  
Karl Prott, Johannes Willkomm, Alfred Zimmermann

SOA Innovation Lab e.V.  
Workstream SOA and Standard Platforms  
c/o Deutsche Post AG  
Charles-de-Gaulle-Straße 20  
53113 Bonn

info@soa-lab.de, helge\_buckow@mckinsey.com, hans-juergen.gross@daimler.com,  
gunther.piller@fh-mainz.de, karl.prott@capgemini-sdm.com,  
johannes.willkomm@capgemini-sdm.com,  
alfred.zimmermann@reutlingen-university.de

**Abstract:** The SOA Innovation Lab has investigated the use of standard software packages in a service-oriented context. As a result, we present a holistic approach for developing a service-oriented enterprise architecture with custom and standard software packages. It starts on enterprise level with the identification of domains where both the SOA paradigm and standard software are of relevance. Here SOA capabilities of products from different vendors can be evaluated within a proposed maturity framework. After pre-requisites and dependencies between distributed components are determined, a high-level architecture can be developed. On the basis of use cases and integration patterns this high level architecture can be refined and verified. Besides methods and related artifacts, we present current adoption issues for standard software packages in service-oriented contexts.

## 1 Introduction

The growing complexity of IT landscapes is a challenge for many companies. A large number of standard software packages - mostly extended and modified, individual software solutions, legacy applications, and different infrastructure components lead to high cost and limited responsiveness to new business requirements. Many companies start enterprise architecture management (EAM) initiatives to tackle this problem. In areas where flexibility and agility are important, SOA is the current paradigm to organize and utilize distributed capabilities. Here, the use of standard software is often a challenge, especially when dealing with services on a fine granular level. Although many vendors of standard software packages advertise that their product is SOA enabled, the adoption of SOA with standard software is still rare.

The use of standard application platforms emerged in many cases from the introduction of ERP systems, aiming to cover all basic business functionalities of a company and

integrating those into a more or less closed system. ERP application suites often dominate the enterprise architecture application layer and the associated automation of business processes in a rather monolithic, proprietary way [Gr07]. Disadvantages of standard application platforms include potential difficulties when fitting to individual business processes. In addition, their limited agility after first customization provides obstacles for the adaption to changed business needs and flexible product or service extensions [Cs96]. Standard platforms are often limited in scope and less performing as compared to their best of breed or custom developed counterparts. Additionally, the integration with other leading systems may be a serious issue and challenge for open system environments, which need to support end-to-end business processes.

Some of these limitations can be overcome with the help of service-oriented architecture. SOA is an IT architecture paradigm that utilizes services as fundamental, flexible, and interoperable building blocks for both, structuring the business and for developing applications. SOA is a business oriented architecture style, often based on best of breed technology for agnostic business services, delivered by applications in a business-focused granularity [Gr07]. An introduction into fundamental SOA concepts, technologies, and case studies can be found e.g. in [Er05, Kr+05]. A discussion of current architectural standards for SOA reference models and ontologies, reference architectures, maturity models, SOA modeling profiles, and open standards work related to the topic of SOA governance is provided in [KE09].

Initially SOA was burdened with hype and inflated expectations. Now it is part of an ongoing discussion about software architecture. The benefits of SOA are recognized, they comprise flexibility, process orientation, time-to-market, and innovation. The adoption of tools and methods for SOA is growing. An overview about the current status of adoption can be found in the SOA Check 2009 [ME09], an empiric study about the development of current SOA systems in Germany. Reports on the maturity of SOA technology from vendors are provided by various analysts (see e.g. [Na+08, Pe+08, KP09]).

We have developed an approach for the design of a service-oriented enterprise architecture with custom and standard software packages. Besides the overall method, we have built a SOA architecture maturity framework. It allows a comprehensive assessment of the SOA ability of standard platforms, covering all dimensions of an enterprise architecture, as e.g. structured in TOGAF [To09]: architecture strategy and management, business architecture, information architecture, application architecture, technology architecture, service & operation architecture, as well as architecture realization. Further useful artifacts of our approach are a method for mapping SOA services to domain maps, SOA use cases and integration patterns – including corresponding templates – and a capability map for SOA infrastructure components.

One of our findings is that some disadvantages of standard application platforms remain, even after the extension of classical standard system modules with fine-grained service-oriented access shells. In many cases such services are not self-sustaining and have to be used with their whole package, or need to integrate several complete system modules.

In this paper we give an overview of our method in Section 2. Section 3 and 4 provide details about the identification of domains, where a combination of SOA and standard software is suitable. Finally, in Section 5 we briefly introduce other artifacts of our approach, which will be published elsewhere.

## 2 Method Overview

In order to identify areas for the use of standard software packages in a service-oriented environment, it is necessary to establish basic EAM capabilities. This includes the definition of a domain map and a picture of the current IT architecture landscape as the core artifacts for EAM. To develop these artifacts, the role of an enterprise architect needs to be established, who is able to understand business needs and translate them into IT requirements. Finally, governance processes need to be introduced to ensure that the target architecture will be implemented.

The central artifact for developing a SOA landscape is the domain map. The SOA Innovation Lab has developed a method for the creation of domain maps, which is explained in detail in Section 3. We have developed a taxonomy and templates for describing domains and have started to collect best practice examples from members of the SOA Innovation Lab. Each domain map is highly individual, suiting a specific business context. Nevertheless, our collection of domain maps can be used as a starting point for the development of templates for specific industries.

SOA has strong benefits, like agility, flexibility, and reduction of redundancies. These benefits are not a priority in all functional areas - some require efficiency and reliability instead, e.g. HR or controlling. This means, there is a need to develop cases for SOA and standard platforms based on individual domain requirement, e.g. sales order management. Therefore, the next step is to assess and weight business needs on domain level. The SOA Innovation Lab has developed principles for the weighting of domains, which are explained in detail in Section 4. As a result, one can define for each domain whether a solution should be SOA enabled.

For a domain with a business need for the benefits that SOA might offer, it has to be assessed if a SOA enabled standard package is available as a solution. In order to identify packages that might suffice, the SOA Innovation Lab has developed a questionnaire to evaluate the SOA ability of a vendor. This questionnaire is based on a SOA architecture maturity framework, which we constructed by integrating different analysis views, using a consistent meta-model approach based on correlation analysis of intrinsic model elements. For this purpose we have transformed CMMI [Cm09], which is originally an assessment framework for software processes, into a framework to analyze systematically enterprise architectures for packaged based SOA environments. Hereby we have used assessment criteria, maturity domains, architecture capabilities, and level rankings from different SOA maturity models [Ma09]. Additionally we have selected architecture elements from state of art architecture frameworks like TOGAF [To09] and Essential [Es09], as well as from Quasar Enterprise [En+08], a collection of methods for creating enterprise architecture artifacts.

In addition to the overall SOA ability, it is necessary to map the vendor solution to the detailed domain map, to evaluate the overall functional fit. The SOA ability of the identified package then needs to be evaluated against specific SOA use cases. The SOA Innovation Lab has developed a taxonomy for SOA use case descriptions which span functional as well as non-functional characteristics, like performance, transaction volume, security.

Once a functional fit has been verified, it is necessary to identify dependencies of the standard software package regarding processes, functions and data. For most standard software packages, the deployment unit is not a small collection of tightly coupled services, but contains rather large bundles of services – often with not transparent interdependencies. In some cases the whole package is required in order to use only few services.

For standard software packages that fulfill the functional and non-functional requirements, a first draft of the to-be architecture then needs to be developed. This includes the high-level system architecture, as well as integration patterns for the physical architecture (see also [FP03, HW04]). We have developed a capability map for integration that allows, to structure corresponding requirements. In addition, a taxonomy for integration patterns has been constructed and a set of best practices for integration in a SOA environment has been compiled. These patterns can be used to assess the integration capability of a standard software package.

The final step is to define the roadmap for the implementation of the SOA enabled standard software packages. The as-is architecture needs to be evaluated against the target architecture. In many cases a number of standard software packages are already in place, but are not used in a SOA enabled way. The gap between as-is and target architecture then needs to be identified and business cases have to be developed, to prioritize the steps of the implementation.

### **3 Definition of Domain Map**

When developing enterprise architecture under the SOA paradigm, a domain map is the central artefact, which structures and organizes the needed capabilities. It uses – from a business point of view – the principles of information hiding, separation of concerns, and tight/lose coupling, and makes them transparent. Since one major goal of SOA is the increase of reusability and flexibility by respecting these principles in business and IT, domain maps provide the corresponding topographic base. Methods for the identification of domain maps are described in several publications, e.g. [En+08]. Figure 1 shows a typical result.

However, more work needs to be done, if an architect wants to identify areas, where SOA and the use of standard software packages make sense. Here we have introduced the following steps for refinement:

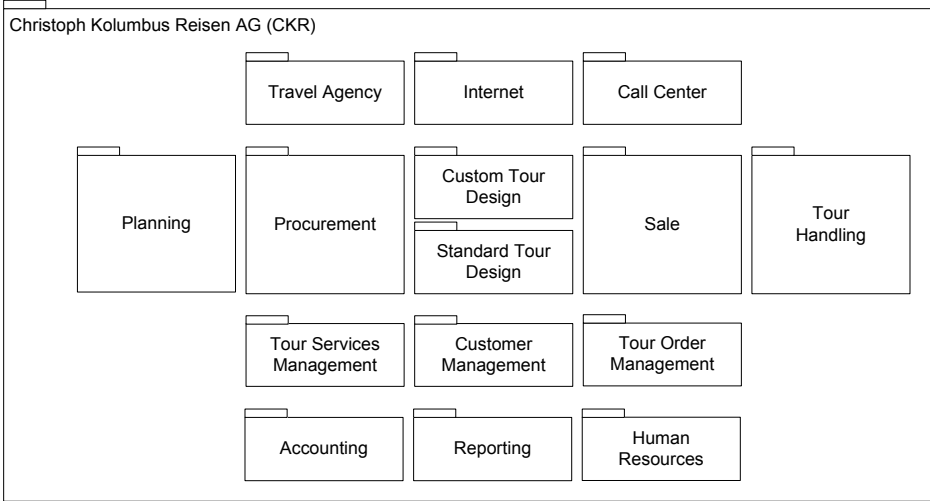


Figure 1: Illustration of a domain map from [En+08]

**1) Define degree of differentiation and sharing:** In this step an architect evaluates on domain level the need for differentiation and the ability to be shared throughout the company. This investigation will result in a categorization of the domains and the corresponding capabilities as shown in Figure 2.

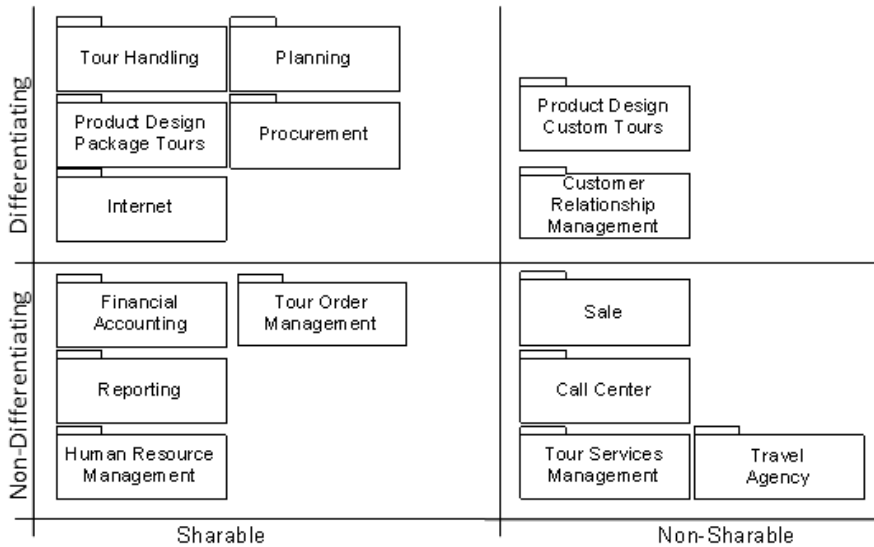


Figure 2: Illustrative example for categorizing domains and their capabilities



The architect needs this categorization in order to apply the following refinement patterns.

**2) Refine domains:** In this step the architect will apply domain refinement patterns to the classified domain map. They are based on the method for finding domains from the SOA Innovation Lab.

**Modularisation:** Here one domain is split into two. This pattern is often applied to domains where the need for differentiation cannot be clearly marked. For example, some capabilities in one domain might have high needs for differentiation, others not. In this case modularisation can make sense, because the IT systems that automate the two different service types are often subject to competing non-functional requirements, e.g. high flexibility versus cost efficiency.

**Generalisation:** This pattern is applied to domains that have some, but not all capabilities in common. The common capabilities are generalized and form a new, centralized domain. They now can be used in a shared way. This situation appears often in domains that cover the same functional area but modularisation has taken place because of a few, e.g. organization or product specific, non-sharable capabilities.

**Aggregation:** This pattern is usually applied to a number of domains that share their low need for differentiation. Here two or more domains are merged. In this case the architect reduces potentially unnecessary complications due to needless loose coupling.

**3) Finalization:** During the whole process of defining a domain map, the stakeholders from business and IT needed to be deeply involved. In addition to this, the architect will have produced different versions of domain maps. In this final step he will need to consolidate the different versions and get a final buy-in for his suggestion.

## 4 Weight Business Needs on Domain Level

Once an architect has defined a domain map as described above, he will need to characterise the domain map and provide information on a more fine-grained level. The goal is to develop a to-be architecture, which marks clearly those areas where the usage of standard software packages makes sense. In order to achieve this, the architect needs to classify each domain according to its *business needs*, in addition to the need for differentiation and ability to be shared. Figure 3 gives an example for this categorization.

In Figure 3 the columns *sales planning*, *sales processing* and *sales delivery* are the subdomains of the domain *sales*. Let us illustrate the principle behind these ratings with few examples:

Business Needs		Sales Planning	Sales Processing	Sales Delivery
①	Need for Differentiation	• H	• H	• L
②	Agility	• M	• H	• L
③	Compliance	• M	• H	• H
④	Ability to be reused	• L	• M	• L
⑤	Business Criticality	• M	• H	• M

H (High)  
M (Medium)  
L (Low)

Figure 3: Illustrative Example for categorizing subdomains according to their business needs

For the domain *sales processing*, the business need for *need for differentiation* was rated high, since it is in our example crucial for gaining advantages over competitors to have unique business processes, which assure the shortest throughput times and the best adherence to delivery dates. Because of this, processes must be changed according to changed market requirements and products as quickly as possible – the underlying IT systems must be agile.

For the domain *sales delivery*, we rated the *need for differentiation* and *agility* low. Since, in our example, it would make only little difference whether the corresponding processes were performed similar to those of the competitors or not. Furthermore, these processes have not changed much in the last years and probably will not change in the years to come.

For the latter there is a high potential to use a standard software package in a service-oriented way sustainably - if a package can be found, which automates the needed functionality appropriately.

For the former the architect will have to do some refinement in order to identify areas on more fine-grained level, where the service-oriented usage of standard software packages makes sense. He will have to identify subdomains and rate them again.

These refinement and rating steps have to be iterated until either agility and differentiation is of low relevance, or the domains represent capabilities that correspond to exactly one role and one goal. This stop criterion is necessary to control the level of granularity and to limit the complexity of an overall SOA landscape. Automating isolated fine-grained services with standard software usually will not bring additional

benefits, because dealing with built-in dependencies of standard software package is most often more expensive than automating these services in a custom-built way.

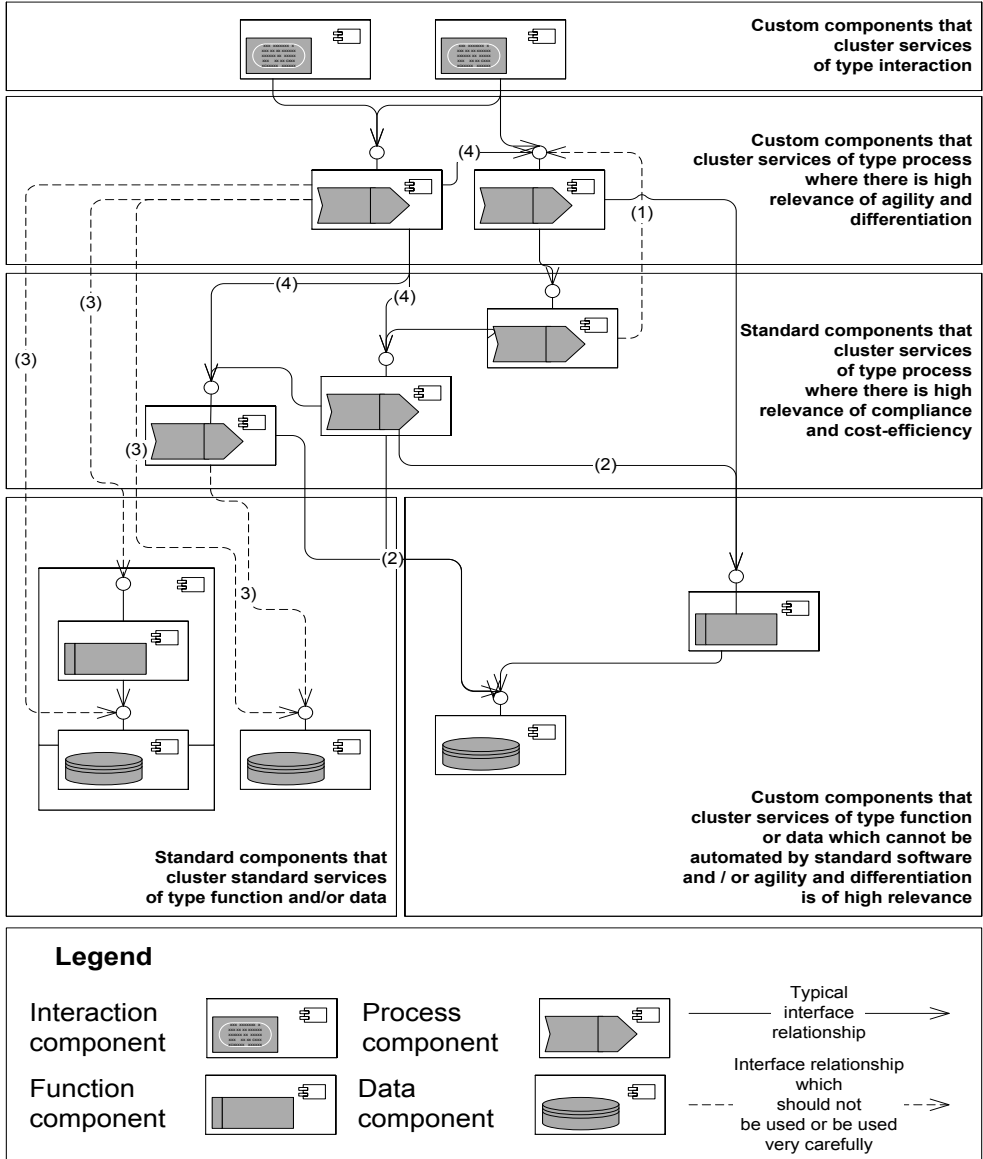


Figure 4: Typical architecture for combining custom and standard software in a service-oriented way

After the architect has iterated through these steps, he will have to identify concrete standard software packages for those domains, for which agility and differentiation are of low relevance. On this basis the architect will develop a to-be architecture, in which fine-grained services on function or data level enhance the standard software packages with needed functionality and where standard software packages offer services that can be combined to higher-level processes. A typical architecture for combining custom and standard software in a service-oriented way is shown in Figure 4.

This architecture is based on the *Application Landscape Reference Architecture* from [En+08]. It suggests that in an ideal SOA, components should only automate services of exactly one category – interaction, process, function or data – and that the latter should not depend on the former in terms of interface relationships. When dealing with standard software, additional aspects are important:

- (1) Try to avoid the integration of complex custom processes components into standard software components, because this would lead to a significant loss in flexibility, when the custom software components need to be changed.
- (2) Missing isolated functionality in standard software components should be added through a service-oriented integration of custom function and data services. Be careful when integrating functionality of higher complexity.
- (3) Try to avoid custom specific relationships to standard software packages if the latter are solely used for isolated function or data services, because the management of those standard software components and the analysis of interdependencies will usually be very time consuming and error prone.
- (4) Combine custom software and standard software processes to higher level processes on the basis of custom built software.

## 5 Further Methods and Artifacts

The identification of a company specific domain map and the assessment of business needs on domain level are starting point for an evaluation, whether SOA with standard platforms makes sense. As outlined in Section 2, next steps need to follow, like an evaluation of the SOA ability of standard software packages, an analysis of use cases, and the development of an appropriate solution architecture. In this section we list some corresponding methods and artifacts which were developed by the SOA Innovation Lab. Details will be given in forthcoming publications.

- **SOA architecture maturity framework:** Basis for an evaluation of the SOA ability of standard software packages. This framework assesses the SOA strategy of a vendor, the business architecture, technical architecture and architecture realization. It uses TOGAF and CMMI structures. The maturity framework includes also a detailed questionnaire and anticipated answer types.

- **Method for mapping SOA services:** Needed for a detailed evaluation, whether SOA can be used with a particular standard software package in domains where SOA is the favoured architecture. The proposed method includes an evaluation of functional, as well as non-functional requirements. Examples for the latter are dependencies between SOA enabled standard components, or pre-requisites on the availability of master data in vendor solutions.
- **SOA use cases:** Template for the description of SOA use cases and characteristics for their classification based on the complexities of the integration logic and underlying application landscape.
- **Integration patterns:** Building blocks for the architecture of SOA use cases. Integration patterns are triggered by business scenarios and integration needs for SOA infrastructure components from different vendors, e.g. federation of different enterprise service repositories. An example for an integration pattern is the so called *Service Façade*. It manages the access to common data for different software packages and offers one unique technical and business interface.
- **Capability map:** Helps to determine and compare the ability and completeness of SOA infrastructure components from different vendors. It provides a starting point for questions like: Which components of a vendor shall be used to realize specific use cases or implement particular integration patterns?

## 6 Outlook

Besides publications on the topics mentioned above, the SOA Innovation Lab plans further investigations on the usage of SOA within an EAM framework.

One area of future activities deals with the construction of software landscapes in the context of monolithic business applications. As a result, we will obtain solution proposals for software landscapes with standard software packages and SOA. Part of this project is the development of a requirement catalogue for vendors, aiming at more flexibility in the usage of standard software components. Furthermore we intend to detail our general results related to domain modelling and road mapping for complex application landscapes. Here the systematic development of architecture principles in a SOA world is key. Finally, we plan to investigate methods and solutions for the integration of internal and external services in mixed application landscapes, which consist of on-premise and on-demand solutions.

## Literature

- [Cm09] CMMI for Development. Version 1.2 Carnegie Mellon University, Software Engineering Institute, 2006, <http://www.sei.cmu.edu/reports/06tr008.pdf>, 2009.
- [Cs96] CSC: Standardsoftware und geschäftliche Flexibilität. Foundation Bericht 107, June 1996.
- [En+08] Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.P., Voß, M., Willkomm, J.: Quasar Enterprise. dpunkt.verlag 2008.
- [Er05] Erl, T.: Service Oriented Architecture. Prentice Hall 2005.
- [Es09] The Essential Architecture Project: <http://www.enterprise-architecture.org>, 2009.
- [FP03] Fowler, M.: Patterns of Enterprise Application Architecture. Addison Wesley 2003.
- [Gr07] Grigoriu, A.: An Enterprise Architecture Development Framework. Trafford Publishing 2007.
- [HW04] Hohpe, G., Woolf, B.: Enterprise Integration Patterns. Addison Wesley 2004.
- [KP09] Kenney, F.L., Plummer, D.C.: Magic Quadrant for integrated SOA Governance Technology Sets. Gartner Research 2009. [http://www.gartner.com/DisplayDocument?doc\\_cd=166481](http://www.gartner.com/DisplayDocument?doc_cd=166481), 2009.
- [Kr+05] Krafzig, D., Banke, K., Slama, D.: Enterprise SOA. Prentice Hall 2005.
- [KE09] Kreger, H., Estefan, J.: Navigating the SOA Open Standards Landscape around Architecture. [http://www.adobe.com/devnet/livecycle/pdfs/soa\\_standards.pdf](http://www.adobe.com/devnet/livecycle/pdfs/soa_standards.pdf), 2009.
- [Ma09] ACMM Architecture Capability Maturity Model, The Open Group 2009. Inaganti, S., Aravamudan, S.: SOA Maturity Model. BP Trends, April 2007, <http://www.bptrends.com/publicationfiles/04-07-ART-The-20SOA-20MaturityModel-Inagantifinal.pdf>, 2009. Sonic: SOA Maturity Model. [http://soa.omg.org/Uploaded-20Docs/SOA/SOA\\_Maturity.pdf](http://soa.omg.org/Uploaded-20Docs/SOA/SOA_Maturity.pdf), 2009. Oracle: SOA Maturity Model, <http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet>, 2009. Open Group: OSIMM Maturity Model for SOA, <http://www.opengroup.org/projects/soa-book/page.tpl?CALLER=faq.tpl&ggid=1319>, 2009. IBM: SIMM Services Integration Maturity Model, <http://www.ibm.com/developerworks/webservices/library/ws-soa-simm/>, 2009.
- [ME09] Martin, W., Eckert, J.: SOA Check 2009. Ergebnisse einer empirischen Studie. [www.soa-forum.de/pdf/SOA-check2009.pdf](http://www.soa-forum.de/pdf/SOA-check2009.pdf), 2009.
- [Na+08] Natis, Y.V., Pezzini, M., Thompson, J., Iijima, K., Sholler, D.: Magic Quadrant for Application Infrastructure for New Systematic SOA Application Projects. Gartner Research 2008. [http://www.gartner.com/DisplayDocument?doc\\_cd=163409](http://www.gartner.com/DisplayDocument?doc_cd=163409), 2009.
- [Pe+08] Pezzini, M., Natis, Y.V., Iijima, K., Sholler, D., Thompson, J., Vecchio, D.: Magic Quadrant for Application Infrastructure for SOA Composite Application Projects Gartner Research 2008. [http://www.gartner.com/DisplayDocument?doc\\_cd=163401](http://www.gartner.com/DisplayDocument?doc_cd=163401), 2009.
- [To09] TOGAF Version 9. The Open Group Architecture Framework 2009.



# A Method for Integrating EAM and BPM

Christoph Moser BOC AG Wipplingerstr. 1 Vienna, Austria christoph.moser@ boc-group.com	Daniel Fürstenau BOC Germany Voßstr. 11 Berlin, Germany daniel.fuerstenau@ boc-de.com	Stefan Junginger BOC AG Wipplingerstr. 1 Vienna, Austria stefan.junginger@ boc-group.com
---	--	---

**Abstract:** Business Process Management (BPM) has been a widely accepted management tool for many years. Recently, the steadily growing complexity of IT landscapes, the increasing speed of change of business models, business processes and organisational structures in many business sectors, and the high pressure for cost reductions in the IT area have lead to the deployment of Enterprise Architecture Management (EAM) in a growing number of organisations. There is an obvious overlapping between BPM and EAM but there are also some significant differences. This paper presents a method to derive a sensible integration between BPM and EAM considering in particular organisation-specific objectives and conditions.

## 1 Introduction

It is well accepted that business processes are a core element of organisations. The better an organisation manages its business processes the more successful it is [Ka96]. There are numerous application areas in which business processes need to be considered, for example re-organisation, capacity planning, process costing, quality management, specification of IT applications, risk management and compliance. Therefore, we define in this paper "business process management" (BPM) in a wide sense, denoting all activities which deal with business processes. On the other hand nearly all organisations suffer from steadily growing complexity of IT landscapes, the increasing speed of changes of business models, business processes and organisational structures, and the high pressure to reduce IT costs. To address these challenges many organisations implemented Enterprise Architecture Management (EAM). Analogous to BPM we define EAM in this paper as all activities which deal with the Enterprise Architecture (EA). For EA we use the common understanding that it covers in a holistic and integrated way strategic aspects, business aspects (such as products, organisational structure, business processes etc.), (IT) applications, interfaces, technologies and infrastructure [Ai08, TO09a, TO09b]. Typical objectives of EAM, from our experience, are usually IT-driven, such as IT master planning (blueprinting), consolidation of application landscapes, implementation and management of service-oriented architectures (SOA), technology management etc. It is obvious from the objects considered within BPM and EAM that there is overlapping, in particular the consideration of business processes is one aspect covered by both disciplines. However, an integration of BPM and EAM is not as simple as it seems. In large organisations BPM



and EAM are organised by different people, with different objectives, and usually with different tools. For example, in EAM a higher abstraction level for business processes is normally appropriate compared to the flow level as is common in BPM.

The authors of this paper are practitioners. They are with a company (BOC) providing BPM and EAM tools (ADONIS, ADOit, ADOben) as well as EAM consulting [BO10]. The authors observed that many organisations tend to chose one of the extremes: Either there is no integration between BPM and EAM at all (usually when implementing EAM) there is a very tight integration between them. In the first case both initiatives do not profit from each other and their results are usually not consistent or might even be conflicting. In the latter case it is often discovered afterwards that a tight integration is not feasible and does not deliver the planned advantages. The main reason is that expectations regarding the exchanged artifacts cannot be fulfilled. Therefore we developed the presented method which helps to define a sensible integration between BPM and EAM considering organisation-specific objectives and conditions.

The remainder of this paper is structured as follows: Chapter 2 introduces a conceptual model for the integration of BPM and EAM and discusses related work. Chapter 3 illustrates the conceptual model by presenting two industry practices for the integration of BPM and EAM. Then, chapter 4 explains our method for integrating BPM and EAM. Chapter 5 concludes the paper with a summary and an outlook to future research.

## **2 Conceptual Model of the Integration Method and Related Work**

The method possesses a process-oriented approach and is based on the terminology of TOGAF 9 [TO09a], one of the most common used EAM frameworks. Figure 1 depicts the conceptual model used to describe the integration between BPM and EAM (examples will be presented in chapter 3). It structures the integration method into three layers: metamodel, deliverable and organisational layer. On the metamodel layer the various information sources are integrated systematically. The metamodel – the language for representing the business processes, architecture artifacts, and their interdependencies - builds the frame for defining "deliverables" (deliverable layer). Deliverables are defined as work products that are contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders. For example typical EAM deliverables are as-is and to-be IT blueprints [GR07]. In BPM a typical deliverable is a work instruction described by a business process model (usually on an activity level). Artifacts are work products that describe the considered objects from a specific viewpoint. So, we see deliverables as specific types of artifacts. To characterise artifacts we use the determinants "abstraction level", "time relevance" (as-is, to-be, or even an exact point of time), "change frequency" and "organisational coverage" (parts of the organisation described by the artifact).

The way BPM and EAM are organised within an organisation can be described as processes which are executed by defined roles [De09, Ke07]. It is clear, that the integration between BPM and EAM takes place by handing over deliverables from BPM to EAM or vice versa. Hence, on the organisational layer, processes to create the deliverables and to organise hand-over's, as well as responsible BPM/EAM roles, have

to be defined. Additionally, it has to be defined when these hand-over's should take place. For this, the change frequency of the deliverables/artifacts has to be considered. If all the above aspects are clarified it usually makes sense to automate the integration by coupling the used BPM and EAM tools. In general it has to be noted that during projects the integration is relatively simple (and in the majority of cases a one-time effort). For example a BPM project might use some EAM deliverables and no sophisticated update activities are necessary. However, most EAM objectives can only be reached if EAM is implemented as a permanent activity within the organisation. The same applies to many BPM objectives.

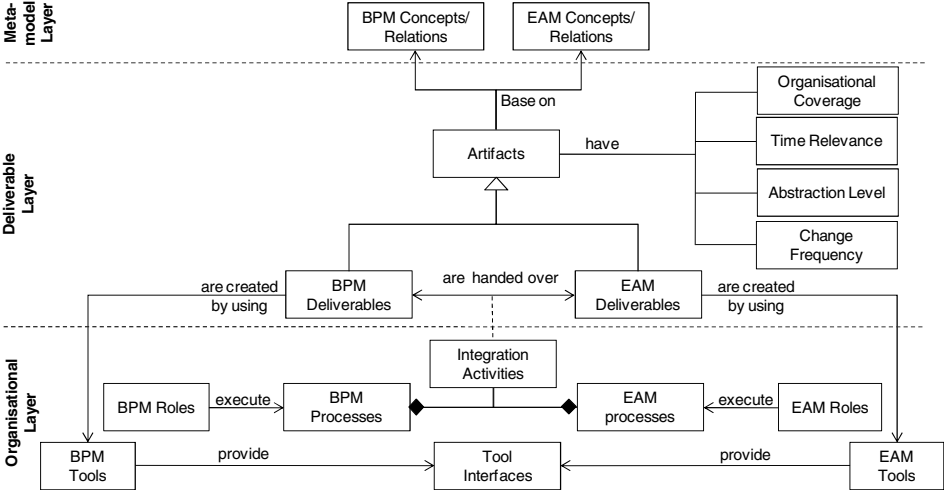


Figure 1: Conceptual Model for the Integration of BPM and EAM

In the EAM field, there are some publications about integrating EAM with other disciplines, discussing the re-use of existing artifacts from other areas within EAM. This is not surprising; EAM metamodels by definition cover many aspects of an organisation [Bu09, Ca08, En08, Fr02, TO09a, TO09b]. In general it is accepted that it makes sense to use federated approaches in EAM. TOGAF 9 [TO09a] states that knowledge of business architecture is a prerequisite for architectural work. According to TOGAF 9 a key objective is to re-use existing material, e.g. business process models (often catered in other organisational processes like enterprise planning, strategic business planning or business process reengineering) as much as possible, in order to keep efforts for architectural work to a minimum. However, there is nothing said on how to organise this in detail. [TO09b and St09] for example discuss integrated metamodels for EAM and BPM, but do not focus on procedural guidance, integration activities and obstacles in order to facilitate collaboration of (existing) BPM and EAM initiatives. In [Fi07] a generic process of how EAM data from various sources can be integrated is discussed. The abstraction level of artifacts is considered, but the other determinants of the presented method (time relevance, change frequency and organisational coverage) are not elaborated.

### 3 Some Integration Scenarios

In this chapter two sample integration scenarios, based on the practical experiences of the authors that leverage the integration method at hand, are discussed. Each scenario is described featuring the three layers of the conceptual model. As stated in chapter 1, depending on the particular objectives, specific methods are applied for EAM/BPM. The integration method brings the required measures for integrating BPM and EAM initiatives into focus. Hence, the discussed metamodels, deliverables, and organisational aspects, are not self-contained methods for BPM or EAM.

#### 3.1 Work Instructions (BPM) – Application Portfolio Management (EAM)

In this scenario, BPM focuses on the provision of work instructions (operational procedures) which is a very common application scenario of BPM [YG01]. Work instructions are depicted in the form of process landscapes (high level process architectures) and detailed activity diagrams advising employees on how to do their work. On EAM side we take the application scenario of Application Portfolio Management as an example (see [Ju08] for details). As depicted in the *metamodel* layer (see figure 2) BPM needs to assign applications to activities of the process descriptions, especially in a business area where users need to use multiple applications for performing the business processes. EAM reuses processes (the process maps) - originally created on BPM side for structuring the work instructions – forming a major building block of the business architecture. Applications are assigned to the processes, e.g. to analyse the support provided by applications for the individual business processes on a high level of abstraction (see method pattern "High Level Process Support" in [Bu09]). The *delivery layer* depicts the required deliverables exchanged between EAM and BPM in conformity with the metamodel. BPM provides process maps on an agreed abstraction level. Work instructions cover the as-is situation. Hence, EAM reuses the as-is process maps of BPM for analysing the as-is business architecture. As to-be processes are not required for work instructions the development of to-be process maps remains an EAM task. On the other side EAM usually possesses application architectures for as-is, to-be and intermediate time states. Since work instructions focus on the as-is only, only as-is applications (applications in state "productive") need to be delivered. On the *organisational layer* on the BPM side the process "Define/Update Process Map" ensures an appropriate structure for the work instructions mapped in "Define/Update Work Instructions". The third BPM process covers the provision of artifacts, like applications or other central elements of BPM (e.g. roles, organisational units), all of which can be reused multiple times in process flow models. In EAM processes for the maintenance of application data, technology products, and process maps (and certainly a lot more EA artifacts, see the metamodel of TOGAF [TO09a]) need to be in place. If process maps are reused then the effort on the EAM side is reduced significantly. For EAM, yearly synchronisation cycles are sufficient as architecture planning usually underlies a yearly cycle. However, BPM needs information on currently productive applications for keeping work instructions up to date and so synchronisation is done on a weekly basis (weekly synchronisation). The roles 'Process Owner' and 'Enterprise Architect' are accountable for the quality of the exchanged deliverables.

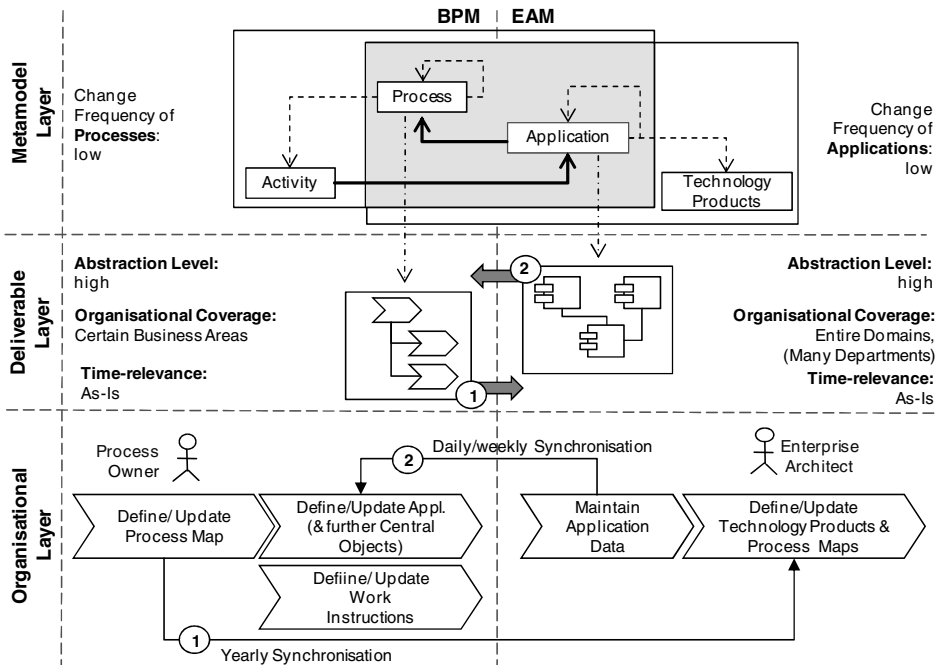


Figure 2: Work Instructions – Application Portfolio Management

### 3.2 Process-based IT Specification (BPM) – Service Enablement (EAM)

In this scenario business process models (on BPM side) represent a major building block for the specification of (IT) applications. Hence, on the BPM side of the *metamodel layer* process maps and process flows on activity level are important concepts. Towards a service-oriented architecture (SOA), the business processes need to be evaluated in order to identify required "services" – flexible software units that can be combined and reused to realise autonomous business functions. "Activities" which should be supported/automated by leveraging "services" need to be identified. On the EAM side our example includes "applications", "interfaces", and "services" as the main concepts. Services are decomposed into "service operations". In order to identify and enable reuse of existing services, the catalogue of services (see deliverable layer of figure 3) maintained by the EAM needs to be accessible for all application development projects on the BPM side. In EAM, the business architecture is usually the starting point for the service-oriented design of the IT application landscape [En08]. As process maps are one prominent way of representing the business architecture it is reasonable to reuse these on the EAM side. As this scenario is elaborated in detail in the scenario above (see chapter 3.1) it is not elaborated here.

On the *deliverable layer*, services are governed on the *instance level* (services versions, see chapter 4.3 for details) as BPM needs to know if the provided functionality of currently available services is sufficient or if a new, advanced version of the service

needs to be developed. As EAM covers cross-departmental domains or the entire organisation, all existing services need to be delivered allowing BPM to select appropriate services. BPM is usually interested not only in existing but also in upcoming services. Hence, the delivered catalogue of services must contain not only as-is services (state "implemented") but also service candidates (services in status "planned", "in development" etc.). This will allow software implementation projects (BPM side) to address requirements on planned services prematurely. This will enable EAM to increase the re-use of the services (one of the major objectives of SOA), by covering requirements of multiple projects. On the *organisational layer* BPM defines and updates process maps and activity diagrams. These business processes are analysed and services are assigned to activities. On the EAM side processes for maintenance of the EA artifacts are in place. Provision of the catalogue on services usually needs to take place in short time intervals (daily/weekly synchronisation) to avoid misguided decisions on service reuse/implementation. Of course, it is reasonable to define services (in state "planned") on the BPM side if no adequate services can be found in the service catalogue and to deliver these to the EAM in order to inform the EAM of required services. However, this is not elaborated here.

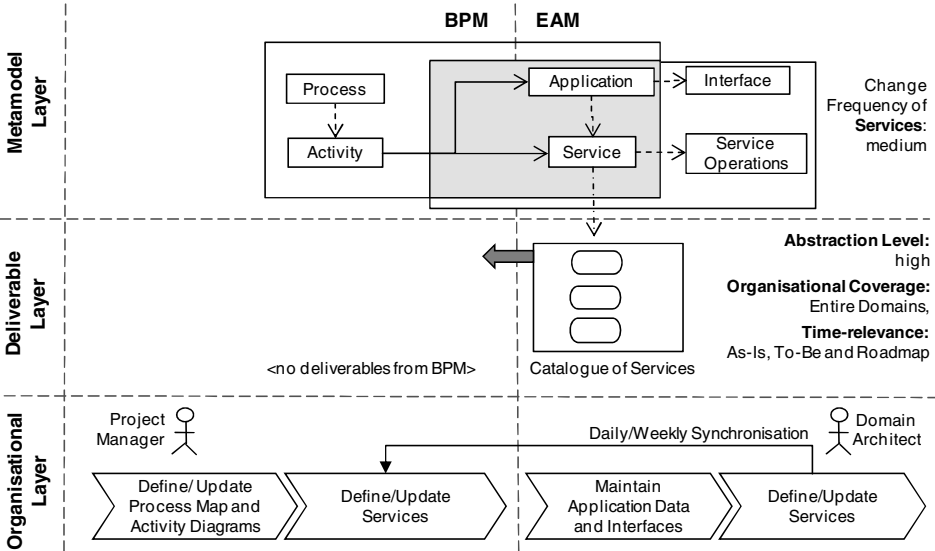


Figure 3: Process-based IT Specification – Service Enablement

#### 4 Procedural Model for Integrating EAM and BPM

This section discusses the steps of the BPM/EAM integration method (see figure 4 for an overview). Step 3 and step 4 are discussed in detail. Prerequisites are clearly defined individual objectives of BPM and EAM. Problems to be resolved in EAM and BPM need to be elaborated.

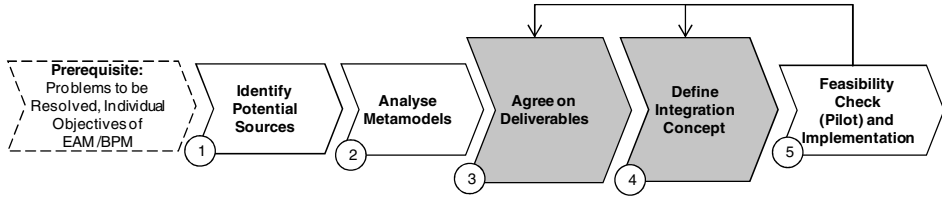


Figure 4: Procedural Model for BPM and EAM Integration

#### 4.1 Step 1: Identify Potential Sources

Based on the individual objectives, BPM and EAM need to examine the organisation in order to identify existing and suitable data sources and data suppliers. From an EAM view, organisational and/or business departments might be a good contact point for provision of BPM information, if required for achievement of individual EAM objectives. For BPM the EAM team (e.g. the enterprise architect) or members of the IT management would be a first contact point. A high-level evaluation of the existing information is performed to assess the potential benefit of the integration. Main factors are actuality, accurateness, and internal consistency of the existing information.

#### 4.2 Step 2: Analyse Metamodels

In this step the structure of the information sources taken into consideration is analysed in detail. The metamodels of EAM and BPM are compared and "shared concepts" (concepts represented in both metamodels) are identified. If no explicit metamodels are available or maturity level in terms of the structure of existing documentation is too low, the metamodels need to be derived first. Based on the results the metamodels might be adapted and/or expanded to make integration possible. For a structured approach for metamodel integration see [Zi07].

#### 4.3 Step 3: Agree on Deliverables

After metamodel integration the determinants (*organisational coverage, abstraction level, change frequency and time relevance*) need to be defined and agreed for each deliverable and its associated elements. Examples are given in the previous chapter. E.g. for the deliverable "catalogue of services" (see scenario "Process-based IT Specification – Service Enablement", chapter 3.2) the element "service" is refined according to the determinates of the integration method. The resulting impact – in terms of practicality, costs, and benefits – needs to be evaluated (see cost-benefit analysis, chapter 4.4).

**Organisational Coverage.** In order to deal with missing or inconsistent information, the organisational coverage of EAM and BPM needs to be considered. Figure 5 depicts an example. In "Business Area A" the required process maps are available while for "Business Area B" no information is available. EAM covers the entire organisation.

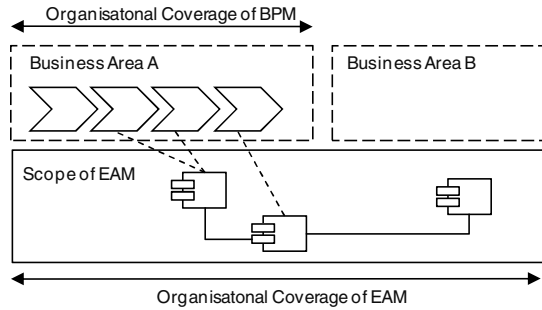


Figure 5: Organisational Coverage

Depending on the intended use, BPM will not stringently cover all business areas of the enterprise. Only some processes or business areas of the enterprise might be in scope for BPM. However, EAM needs a broad base information set that covers the whole organisation or at least entire domains to identify improvement potential [Ai09], e.g. to uncover potential reuse of web services. For this, the degree of organisational coverage has to be evaluated before deciding on scenario integration.

**Abstraction Level.** Due to the different objectives that the information is acquired for, often the level of abstraction varies between BPM and EAM. It is important to determine the appropriate abstraction level for the exchanged deliverables. The granularity of the deliverables needs to be determined according to objectives, and measures of BPM/EAM. Finer-grained deliverables permit closer management and measurement but require greater effort to govern and maintain. Figure 6 depicts different mechanisms for the definition of EAM and BPM artifacts on various abstraction levels (see [Mo08] for details).

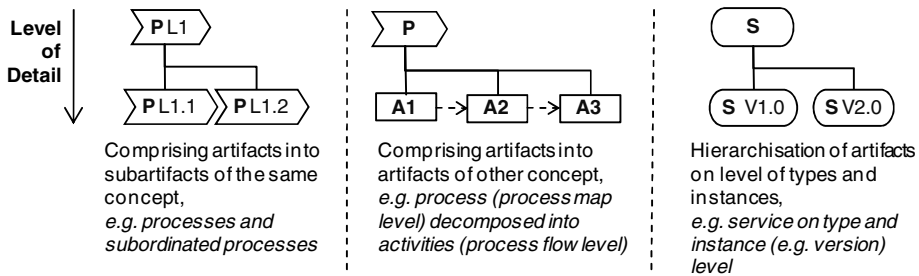


Figure 6: Different Levels of Abstraction in BPM and EAM

**Change Frequency.** In reality artifacts, like processes and applications, are subject to continuous change. However, the change frequency of the deliverables describing these artifacts highly depends on the chosen abstraction level (see above). While artifacts on a low abstraction level are rapidly changing (e.g. versions of services or activities of a certain process) artifacts on a higher abstraction level (e.g. services on type level or top level processes) tend to be more stable. As a consequence creating and exchanging deliverables on high abstraction levels reduces the need for short synchronisation cycles and efforts for integrating the "external" deliverables. Figure 7 depicts change frequency of (a) applications and (b) processes on different abstraction levels.

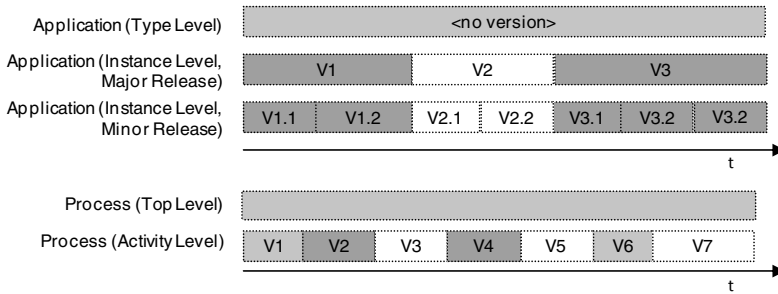


Figure 7: Change Frequency of Artifacts/Deliverables

**Time Relevance.** Usually the considered time horizons of BPM and EAM differ (see figure 8). In BPM the as-is state (e.g. mapping of work instructions, see chapter 3.1) or a mid-term time horizon (e.g. specification of IT systems, see chapter 3.2) is considered. Strategic aspects in EAM (e.g. planning for new products, business models) require the modelling of to-be architectures in the long-run. Therefore the EA is typically described by as-is and to-be states and several intermediate states (roadmap states) that vary in their level of detail. When suitable information for future architectural planning is not available on the BPM side, it needs to be created and maintained on the EAM side. An example is given in chapter 3.1 where the BPM delivery "process maps" represents the as-is but no to-be process maps can be delivered by BPM.

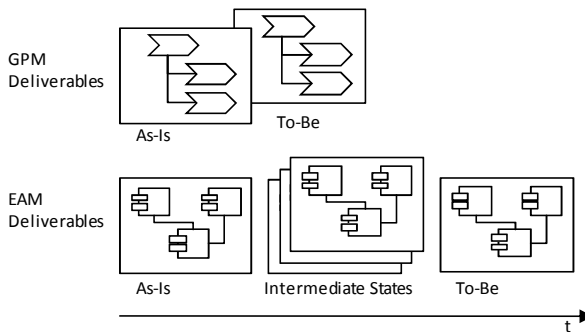


Figure 8: Time Relevance

#### 4.4 Step 4: Define Integration Concept

Depending on the complexity of the integration scenario "clarification of data ownership", "adaptation of existing processes", "provision of adequate tool support", "execution of cost-benefit analysis" and "definition of SLAs" is required.

**Define Data Ownership.** The data ownership for the exchanged artifacts is clarified to ensure data integrity and consistency. Responsible roles and accountabilities have to be determined. BPM usually has sovereignty on artifacts of the business architecture (e.g. business processes and organisational structures). EAM usually owns technical artifacts like applications, interfaces and technologies forming the IT architecture. However, this



is not always true. Chapter 3.1 brings up the example of process-based work instructions depicting the as-is process maps. It is mentioned that defining the to-be process maps might be up to EAM. Figure 9 depicts roles involved in the BPM and the EAM and ownership of artifacts/deliverables.

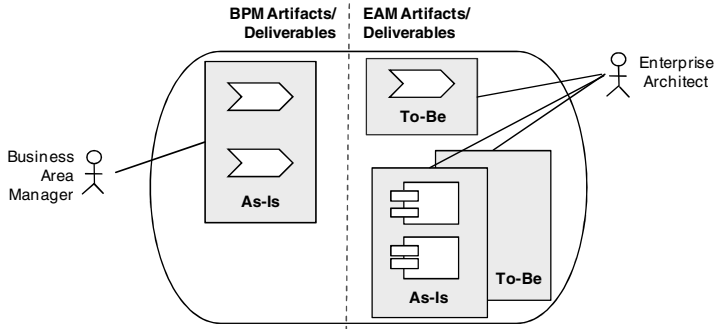


Figure 9: Agreement on Data Ownership of Artifacts and Deliverables

**Adapt Existing Processes.** To keep the conjoint information base up to date, procedures and guidelines for the creation, approval and exchange of deliverables needs to be defined and integrated into existing processes on the BPM and the EAM side (see integration activities of the conceptual model, figure 1). [Mo09] and [Fi07] discuss patterns/methods for organising federated data acquisition for EAM. These can be applied on the BPM side as well. Besides the processes, responsible and accountable roles need to be defined and staffed. If the necessary skills to create the required deliverables are not available among the staff, then these skills need to be developed.

**Tool Support.** When the criteria of chapter 4.3 have been agreed upon and on both sides structured information is available, automating the exchange of the deliverables is feasible. Required tool mechanisms for putting architecture artifacts (EAM side) under appropriate control are presented in [Mo08], discussing functionality like versioning, release workflows, metamodeling functionality and logical deletion of artifacts.

**Cost-Benefit Analysis.** The determinants *abstraction level* and *change frequency* are effort drivers for integration of EAM and BPM. It has to be considered, that update mechanisms are usually needed at the receiving side because when having received a former version of the deliverable it has been assigned to other artifacts. Hence, besides the effort for creating the deliverables on supplier side, the effort for integration of the deliverables on receiver side also needs to be considered carefully. Costs in terms of effort for creation and integration of the deliverables need to be balanced with the gained benefits. Appropriate tool support lowers efforts for delivery and integration. *Organisational coverage* and *time relevance* mainly influence the possible benefits of integration. Consider the sample scenario of chapter 3.1. As working instructions might focus on a single department, that may not cover a significant part of the organisation, and only contains as-is process maps, reuse of these process maps might be of little value to the EAM.

**Agree Service Level Agreements on Deliverables.** [Fi07] proposes to set up maintenance contracts to ensure the quality and maintenance schedule of the deliverables. This also applies for the method presented here. In practical terms, Service Level Agreements (SLAs) must be defined in sufficient detail and scope for the deliverables exchanged. The criteria discussed in chapter 4.3 and 4.4 must be defined.

#### **4.5 Step 5: Feasibility Check (Pilot) and Implementation**

In this step the integration concept is piloted. Based on this pilot the original motivation for the integration is checked. The solution is fit for the purpose of supporting subsequent work in BPM/EAM. If the solution does not bring the aspired effects, the structure and granularity of the deliverables (see step 3, chapter 4.3) and the defined integration concept (see step 4, chapter 4.4) needs to be reconsidered. If the pilot is successful the integration concept can be implemented.

## **5 Experiences and Outlook**

This paper presented a method for BPM/EAM integration. The most important advantages – business/IT alignment and reduction of efforts due to the reuse of existing documents/models – were stated. A conceptual model, covering all aspects to be considered for the scenario was developed. Then, real-world examples were briefly discussed in order to clarify the elements of the conceptual model and the problem domain being addressed. Then the main steps of the integration were introduced. The paper then focused on the criteria to be agreed upon for sustainable and effective integration, taking the cost-benefits of the scenario into account.

The discussed mechanisms evolved step by step during the various EAM and BPM projects conducted by the authors. The authors are aware that the presented method, to a great extend, can be applied to various integration scenarios in any other domain of enterprise modelling. An example is the integration of EAs with data of Configuration Management Systems.

The suitability of the presented method certainly needs to be proven in a more structured manner. Our evaluation approach will define and evaluate key performance indicators (KPIs), e.g. the "reduction in scenario integration implementation time/effort" and indicators evaluating the reduction in workload on the BPM and EAM side. Soft indicators, like improved business/IT alignment will be collected via structured interviews. Among our future research objectives is the development of concrete patterns for BPM and EAM integration based upon the presented method, which can be understood as an extension of the EAM pattern catalogue [Bu09] and the EAM process patterns of [Mo09].

**Acknowledgment:** The authors want to thank the anonymous reviewers and our colleagues for the time they spent for reading, commenting and giving feedback on the EAM/BPM integration method.

## 6 References

- [Ai08] Aier, S. et al.: Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. In: *Wirtschaftsinformatik*, 50, 4, 2008, pp. 292-304.
- [Ai09] Aier, S. et al.: Enterprise Architecture Design as an Engineering Discipline. In: *AIS Transactions on Enterprise Systems*, 1, 1, 2009, pp. 36-43.
- [BO10] BOC Homepage, [www.boc-group.com](http://www.boc-group.com) (access 2010-01-03).
- [Bu09] Buckl, S. et al.: Enterprise Architecture Management Pattern Catalog Wiki. Garching b. Muenchen, 2009, <http://www.matthes.in.tum.de/wikis/eam-pattern-catalog/home> (access: 2010-01-05).
- [Ca08] Capgemini (2008) Integrated architecture framework. [www.capgemini.com/iaf](http://www.capgemini.com/iaf) (access 2010-01-03)
- [De09] Dern, G.: *Management von IT-Architekturen*. 3rd Edition, Vieweg, Wiesbaden, 2009.
- [En08] Engels, G. et al.: *Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten*. dpunkt.verlag, Heidelberg, 2008.
- [Fi07] Fischer, R. et al.: A Federated Approach to Enterprise Architecture Model Maintenance. In: Reichert, M., et al. (Eds.): *Proceedings of the 2nd Int'l Workshop EMISA 2007*, LNI P-119, 2007, pp. 9-22.
- [Fr02] Frank, U.: Multi-Perspective Enterprise Modelling (MEMO) - Conceptual Framework and Modelling Languages. In: *Proceedings of the Hawaii International Conference on System Sciences (HICSS-35)*, Honolulu, 2002, pp. 10ff.
- [GR07] Gronau, N., Rohloff, M.: Managing Change: Business/IT alignment and Adaptability of Information Systems. In: Österle, H. et al. (Eds.): *Proceedings of the 15th European Conference on Information Systems (ECIS2007)*, St. Gallen, Switzerland. 2007.
- [Ju08] Junginger et al. (2008): Anwendungsportfoliomanagement mit ADOit im ZIVIT. In: Riempp, G., Stahringer, S. (Eds.): *Unternehmensarchitekturen. HMD – Praxis der Wirtschaftsinformatik*, 262, Heidelberg, 2008, pp. 29-38.
- [Ka96] Karagiannis, D. et al.: Introduction to Business Process Management Systems Concepts. In: Scholz-Reiter, B.; Stickel, E. (Eds.): *Business Process Modelling*. Springer, 1996, pp. 81-106.
- [Ke07] Keller, W.: *IT-Unternehmensarchitektur. Von der Geschäftsstrategie zur optimalen IT Unterstützung*. dpunkt.verlag, Heidelberg, 2007.
- [Mo08] Moser, C. et al.: Business Objectives Compliance Architecture Framework - Mechanisms for Controlling Architecture Artifacts. In: *Modellierung 2008*, 12.-14. März 2008, Berlin P-127, 2008, pp. 73-88.
- [Mo09] Moser, C. et al.: Some Process Patterns for Enterprise Architecture Management. In: *Patterns in Enterprise Architecture Management (PEAM2009)*. Software Engineering 2009 Kaiserslautern, Workshopband, 2009.
- [St09] Staehler, D. et al.: *Enterprise Architecture, BPM und SOA fuer Business-Analysten Leitfadens fuer die Praxis*. Muenchen, Hanser, 2009.
- [TO09a] The Open Group: The Open Group Architecture Framework, <http://www.opengroup.org/togaf> (access 2010-01-03).
- [TO09b] The Open Group: ArchiMate 1.0 Specification – online, [http://www.opengroup.org/archimate/doc/ts\\_archimate](http://www.opengroup.org/archimate/doc/ts_archimate) (access 2010-01-29).
- [YG01] Yahya, S., Goh, W.: The implementation of an ISO 9000 quality system. In: *International Journal of Quality & Reliability Management*, 18, 9, 2001, pp. 941-966.
- [Zi07] Zivkovic, S. et al.: Facilitate modelling using method integration: An approach using mappings and integration rules. In: Österle, H. et al. (Eds.): *Proceedings of the 15th European Conference on Information Systems (ECIS2007)*, St. Gallen, Switzerland. 2007, pp. 2038 – 2050.

# Interrelating Concerns in EA Documentation – Towards a Conceptual Framework of Relationships

Sabine Buckl, Florian Matthes, Christian M. Schweda  
{buckl,matthes,schweda}@in.tum.de

**Abstract:** Over the last years a multitude of approaches and frameworks making prescriptions on how to document an enterprise architecture (EA) have been developed. These approaches target different purposes and correspondingly different concerns (areas of interest) in the architecture. In this way, a company seeking to develop or evolve an organization-specific EA documentation technique based on different approaches most likely runs into difficulties to understand the interdependencies between the different frameworks. In this research-in-progress paper, we revisit well-known documentation approaches to devise different types of *concern relationships* that may be helpful during development and evolution of an organization-specific EA documentation technique.

## 1 Introduction and motivation

The increasing frequency of change, modern enterprises face in today's globalized and competitive environments, leads to a rising internal complexity of the socio-technical system *enterprise*. A promising and commonly accepted instrument to deal with this complexity and to foster business-IT-alignment is enterprise architecture (EA) management [HV93, Lan05, RWR06]. Originating from the field of information systems architecture (cf. [Zac92]), EA management takes a holistic perspective targeting all areas of an enterprise from business and organizational aspects via application and information to infrastructure and data aspects. EA is thereby in the sense of the ISO Standard 42020 understood as the "fundamental organization of a system [enterprise] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [Int07]. A multitude of methods to EA management has been developed by researchers, practitioners, and standardization bodies (cf. [Fra02, KW07, Lan05, The09, vdRvV08, RWR06]). Although differing in respect to the scope, reach, and focus, the proposed methods usually distinguish the following activities of the EA management function: a) document and maintain the current state of the EA, b) develop and describe planned and target states of the EA, c) enact and communicate planned EAs and architectural principles, and d) analyze and evaluate architectures.

A central challenge arising during the aforementioned typical activities of EA management is *stakeholder involvement* [The09, Sch08]. To address this challenge the architectural documentations referring to the current, planned, or future states of the EA, need to represent the corresponding concerns of the stakeholders. Thereby, concerns are defined in accordance with the ISO Standard 42010 as "those [areas of] interests which pertain to

the system's [enterprise's] development, its operation or any other aspect that are critical or otherwise important to one or more stakeholders" [Int07]. Put in other words, a concern can be understood as the area of the enterprise that the respective stakeholder is interested in. Whereas, a product manager for instance is interested in the performance of the business process and the services responsible for creating the product, an application manager is concerned with the standard conformity of the business applications.

In the holistic perspective of EA management the two aforementioned concerns are clearly interrelated. Crosspoints are the terms service and business application. The thereby denoted concepts are most likely related or may even be identical. The differences in terminology can be explained by the different language communities the two stakeholders belong to. To develop a comprehensive EA description, covering the different concerns of the stakeholders, the relations between the concerns should be made explicit. This directly yields the research questions that our paper seeks to address:

*What types of relationships between EA concerns exist? How can these relationships be utilized in developing and evolving EA models?*

In the remainder of the article above research questions are approached in a practice-driven manner. More precisely, we explore exemplary concern descriptions from the *Enterprise Architecture Management Pattern Catalog* [Cha10], a collection of best-practice solutions for recurring problems in the context of EA management (cf. Section 2). Thereby, special emphasis is put on the conceptual models that are used in the pattern catalog to present the conceptualizations underlying the concerns. From this, we identify different types of concern relationships and discuss how these relationships can be defined. Complementing the foregoing findings, we revisit the state-of-the-art of concern relationships as contained in prominent EA management approaches from literature (see Section 3). In Section 4, the paper concludes with a critical reflection of the achieved results and an outlook on future areas of research.

## 2 Concern relationships

The *Enterprise Architecture Management Pattern Catalog* [Cha10] presents a collection of best-practice solutions to recurring problems in the context of EA management. Centrally, the catalog establishes a 'trifecta' of successful solutions consisting of *management methods*, *architectural viewpoints* and *information models*<sup>1</sup>. Especially the information models reflect areas-of-interest in the overall EA, i.e. correspond to architectural concerns in the sense of the ISO 42010 [Int07]. Put in other words, it is sensible to understand an information model as representative of an underlying domain ontology targeting a specific part of the EA. Against the background of this understanding, we revisit three selected information models, whose selection is inspired by a practice case. In this case (see [Die08] for a more comprehensive description of the case) an international financial service provider

---

<sup>1</sup>In the pattern catalog, the conceptual meta-models describing the EA or parts thereof are called "information models".

was using the pattern catalog to implement compliance management in the EA management context. The financial service provider started compliance management by establishing a 'book of standards' for business applications, i.e. a collection of technologies that comply to architectural preselections. Based on this set-up, a more mature EA management function was built on *architectural solutions*<sup>2</sup>, which define a set of matching technologies on different tiers together with additional information on how to integrate these technologies into a solution to realize e.g. a four tiered architecture. In this sense the initial concern of compliance management in the example case was 'broadened'. Further, the the company experienced the pressure to standardize its application landscape. This gave rise to a different concern, in which it was determined, if a business application was custom or standard software, respectively.

From the above discussions, we can derive a set of relationship types relating architectural concerns. The most basic type represents the fact that concerns share concepts or do not. In the light of the discussion on terms and their meanings, two different ways to understand 'share' exist: *syntactic* and *semantic*. While sharing in the latter sense means that two concerns have a concept with similar meaning but possibly different names in common, syntactic sharing describes that two concerns are related via a concept with common name. To qualify these two different understandings of 'share', we denote 'syntactic sharing' by the name *intersection*. Based on the intersection, we consistently establish the notion of *compatibility*, indicating that all intersecting concepts of two concerns are also semantically shared. If in contrast one intersecting concept is assigned distinct meanings in both concerns, the concerns are called *incompatible*. Admittedly, the forstanding definition of 'semantic sharing' is fairly informal and one might argue, that *equivalent meanings* in a strictly ontological sense may need a more precise definition. In response, we rely on a rather weak form of semantic sharing here, assuming that the concepts' meanings are given as textual definitions. Based on these equivalence of meanings is interpersonally defined via stakeholder consensus in accordance to Kamlah and Lorenzen [KL96], i.e. two concepts share a common meaning, if "any informed member of the language community would [say so]".

The following two information models from the EA management pattern catalog [Cha10], namely STANDARD VS. CUSTOM SOFTWARE (I-41) and USAGE OF ARCHITECTURAL SOLUTIONS (I-6), illustrate the compatibility relationship. To facilitate considerations on the relationships between concerns, we present the central parts of the information models, but abstain from adhering to the pattern structure as utilized throughout the pattern catalog.

### **Standard vs. Custom Software (I-41)**

*Concern:* You want to know, whether a business application is custom or standard (off-the-shelf) software or not.

Important concepts in the information model of Figure 1 are:

**BusinessApplication** A software system, which is part of an information system used to support an organization's business processes. The `type` attribute, if set, indicates whether the application is custom or off-the-shelf software.

---

<sup>2</sup>Sometimes architectural solutions are also referred to as *technology stacks* or *platforms*.



Figure 1: Information model pattern I-41 (adapted from [Cha10])

**BusinessApplicationType** Enumeration of values stating that a corresponding business application is either `custom` or `standard` software.

### Usage of Architectural Solutions (I-6)

*Concern:* You want to know which business application conforms to which architectural solution, if any.



Figure 2: Information model pattern I-6 (adapted from [Cha10])

Important concepts in the information model of Figure 2 are the `BusinessApplication` concept defined above and:

**ArchitecturalSolution** An approved set of compatible technologies that can be used to build business applications. The architectural solution further determines how its constituting technologies are used in a thereby realized business application.

**conformsTo** Relationship expressing that one or more business applications are built upon the technologies described in the corresponding architectural solution, if any.

The two concerns `STANDARD VS. CUSTOM SOFTWARE` and `USAGE OF ARCHITECTURAL SOLUTIONS` represented by their corresponding information models I-6 and I-41 can be considered *intersecting* via the shared concept of the `BusinessApplication`. This concept's semantics is textually given in both models with the same description. Further, the intersection quality of the concept can be illustrated along the fact that a valid instance of the class `BusinessApplication` in I-6 is also a valid instance of the class with same name in I-47. In this light, the two concerns are not only intersecting but are also *compatible*, i.e. can be integrated with each other without causing contradictions.

Grounding on the basic notion of compatibility, *subconcern* and *superconcern* relationships can be established. One concern A is subconcern of another concern B, if the concerns are compatible and all concepts of A are intersecting concepts. Put in other words,

concern B completely covers the area of interest represented by concern A. This relationship type mirrors the notion of more abstract or more concrete concerns as outlined above. In a mathematical sense, the relationship type subconcern and its counterpart superconcern further give rise to a notion of concern *equivalence*. Two concerns A and B are equivalent, if A is both *sub-* and *superconcern* of B. To exemplify the understanding of sub- and superconcern, respectively, we introduce the information model ARCHITECTURAL SOLUTION CONFORMANCE (I-67) below.

**Architectural Solution Conformance (I-67)**

*Concern:* You want to know to which extent your custom-built business applications comply with the architectural solutions that they are allowed to be built on.

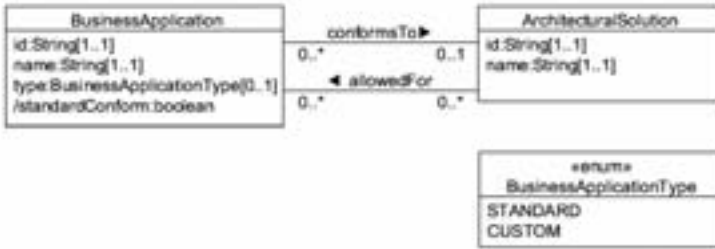


Figure 3: Information model pattern I-67 (adapted from [Cha10])

Complementing the concepts from the information model in Figure 3 the following *derive expression* using the *Object Constraint Language* (OCL) [OMG06] applies:

```

context BusinessApplication::standardConform
derive: type == STANDARD | allowed.isEmpty() |
        allowedFor->contains(conformsTo)
  
```

Important concepts in the information model (cf. Figure 3) are the ones used in I-6 and I-41, respectively. In addition, the following concepts are employed:

**allowedFor** Relationship expressing that a business application is allowed to build upon the technology set as described by zero or more architectural solutions. If no solution is specified, the business application is allowed to use any technologies.

**standardConform** Attribute indicating whether a business application conforms to the standards, i.e. is off-the-shelf software or uses an allowed architectural solution, or not.

Relating the concerns STANDARD VS. CUSTOM SOFTWARE and ARCHITECTURAL SOLUTION CONFORMANCE, i.e. the information models I-41 and I-67, we can exemplify the *sub-concern* and *super-concern* relationship, respectively. Described in more detail,



we see that I-67 shares all modeling concepts of I-41 and further also the textually described semantics for the corresponding classes. In consequence, any valid instantiation of classes from the latter information model also forms a valid instantiation of concepts from I-67, which nevertheless adds some more concepts, i.e. classes, attributes, and associations. The aforementioned arguments also apply on the relationship between the concerns USAGE OF ARCHITECTURAL SOLUTIONS and ARCHITECTURAL SOLUTION CONFORMANCE, in which the former can be considered a sub-concern of the latter.

Reframing the practice case introduced at the beginning of the section, a key benefit of understanding the relationship becomes obvious. Having started with architectural descriptions satisfying the concerns I-6 and I-41, our financial service provider can derive further concerns that not only relate to the already addressed concerns, but are also *compatible*, i.e. employ a similar understanding of the relevant concepts. Additionally, the *sub-super*-relationship between concerns facilitates consideration on possible next stages of refinement, when it comes to architectural descriptions. In the practice case, the financial service provider did not find the concerns as covered by I-6 and I-41 to be sufficient for a comprehensive compliance management. Knowing, that I-67 not only subsumes I-6 and I-41, but also provides additional concepts, the company was able to adopt the information model for architectural descriptions as devised by I-67.

In the latter sense, the relationships between the concerns can provide valuable insights into the different ways to conceptualize an EA, but also to evolve the corresponding description technique. To facilitate considerations on the relationships between concerns, we further propose to use a graphical visualization thereof, called *concern map*. An exemplary map is shown as part of this paper’s example section in Figure 4.

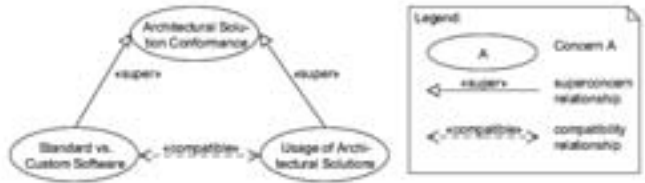


Figure 4: Concern map for the three exemplary concerns

### 3 Concerns in the state-of-the-art in EA descriptions

EA management is a research topic with an increasing number of publications [LW04], in which especially the fields of EA modeling and EA analysis are heavily researched. Central to both activities is the notion of the architectural description reflecting the corresponding architectural concerns. Subsequently, we explore the state-of-the-art in EA modeling and analysis with a special emphasis on the EA descriptions used thereby as well as their underlying concerns.

The approach of *multi-perspective enterprise modeling (MEMO)* was initially presented by Frank in 2002 [Fra02]. Therein, Frank outlines a modeling framework, which is based on an extendable set of special purpose modeling languages. The special purpose modeling languages correspond to the different language communities, which typically exist in an enterprise, e.g. salespersons or project managers. By further providing a common meta-language, the *MEMO meta modelling language (MML)* [Fra09], which the special purpose languages rely on, the integration of the different languages can be facilitated. Examples for special purpose languages are e.g. the *strategy modelling language (MEMO-SML)* [Fra99] and the *organization modelling language (MEMO-OrgML)* [FJ01]. Concepts of two different languages can be associated, i.e. are common to both languages. Therefore, the different concerns addressed by the MEMO special purpose languages can be regarded as being *intersected* and *compatible*.

The communication challenge already mentioned in the approach presented by Frank, is further discussed by Buckl et al. in [BEL<sup>+</sup>09], Schelp and Winter in [SW09], and Schönherr in [Sch08]. While the first publication refers to the communication challenge within an enterprise, the latter two publications discuss the different language communities mirroring the academic groups conducting research in the area of EA management. Due to the absence of a standardized terminology and a commonly-accepted description language for EA management, the different approaches and communities have developed their own terms, leading to *incompatibility* issues, when different approaches are combined.

A systemic perspective on EA modeling is presented by Wegmann et al. in [WBL<sup>+</sup>05]. They provide a method and a tool to formalize the alignment of the multiple levels that constitute an EA. In particular, they propose to organize the different concepts that constitute the EA in *organizational* and *functional levels*. Thereby, the functional levels represent behavioral and the organizational levels the constructional hierarchy. Within each organizational level, two different viewpoints are available, the *information viewpoint* – a black box view on the respective concern – and the *computational viewpoint* – a white box specification of the concern. These viewpoints can be refined, which results in a hierarchy of viewpoints and the underlying information, respectively. Based on this understanding, a *super- and sub-relationship* between different viewpoints and concerns can be identified. Furthermore, Wegman et al. point out that a "vocabulary mapping" [WBL<sup>+</sup>05] between related concepts on different organizational levels has to be performed, which can be ascribed to the already mentioned *compatibility* question.

In [JE07] Johnson and Ekstedt discuss an approach to EA decision making based on EA models and analyses. Based on the current documentation of the EA, future EA scenarios are derived and assessed in respect to selected quality attributes, as e.g. performance, interoperability, availability, security, or usability. For each of these quality attributes, an *influence diagram* is presented by the authors, which details on causal dependencies between architectural properties. If for instance availability is considered, the reliability or recoverability of the system under consideration influences the overall availability. Thus, the concerns included in an influence diagram have a *subconcern/superconcern* relationship to each other, while concerns of different influence diagrams may be *intersected*.

Central ideas for a language for describing EAs are outlined by Jonkers et al. in [JvBA<sup>+</sup>03] and further elaborated in [JGBvB05]. They discuss two requirements that relate to the no-

tion of the concern, namely *meta model flexibility* and *integration of heterogeneous models*. Meta model flexibility is thereby meant to demand that general EA description concepts can be refined to organization-specific concepts and standards. Conversely, a language for describing EAs must facilitate the integration of heterogeneous models, as special purpose languages for specific parts of the EA exist, whose concepts can be 'translated' into or 'associated' with concepts for EA description. Jonkers et al. further discuss in [JvBA<sup>+</sup>03] the distinct levels of detail, on which the corresponding descriptions may act. In this sense, one or more concerns from special purpose modeling languages might be 'aggregated' into a single concern at the EA description level. In [JGBvB05] Jonkers et al. provide a core meta-model for architectural descriptions, more precisely behavioral aspects thereof, based on a dichotomy of *service-* and *implementation-aspects*. The corresponding basic concepts *service* and *behavior element* are refined to specific *business*, *application* and *infrastructure-layer* concepts, respectively. Nevertheless, the basic modeling language allows to omit concepts on some of these layers, if they are not of interest for the specific modeling purpose. Put in the terminology of our concern-discussion, the language implicitly allows for the derivation of *subconcerns*.

## 4 Conclusion and Outlook

In this paper we discussed the concept *concern* in the context of EA modeling and EA management, respectively. Concerns thereby represent distinct information demands in respect to the described architecture and can hence be reflected by information models that provide a 'schema' for the information to be modeled and collected during the corresponding EA management process. Based on this model-centric understanding of concern, we revisited the state-of-the-art in EA modeling and analysis to derive a set of relationship types that may exist between concerns. Taking also into account the semantic aspects of the information models, we further refined the understanding of the relationship types while abstaining from mathematical formalization thereof. Complementing the forestanding considerations, we exemplified the relationship types with three selected concerns from the EA Management Pattern Catalog of [Cha10].

This paper presents itself as a research-in-progress paper, that outlines the idea of concern relationships. From this point different directions for further development and research are open. At first, the informal presentation of the relationship types in this paper should be complemented with a more formal understanding thereof. In this respect, the semantics of the different relationship types could be formally defined based on a formal understanding of architectural descriptions. This would then allow for more intricate considerations on the relationships between information models that reflect the corresponding concerns. Secondly, concern relationships might not only be an issue of theoretical interest, but may prove to be interesting for supporting maturity considerations for EA management functions. In this respect, the relationships could also be beneficial for evolving an already existing EA management function towards a 'broader' and 'deeper' coverage of the overall EA. This subject has yet not been investigated in practice, although the notion of the "related" concerns as outlined by Ernst in [Ern10] and in [Cha10] have shown to be bene-

ficial during the process of establishing and evolving an EA management function.

## References

- [BEL<sup>+</sup>09] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Florian Matthes, and Christian M. Schweda. State of the Art in Enterprise Architecture Management 2009. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2009.
- [Cha10] Chair for Informatics 19 (sebis), Technische Universität München. EAM Pattern Catalog Wiki. <http://eampc-wiki.systemcartography.info> (cited 2010-02-25), 2010.
- [Die08] Thomas Dierl. *Models, Methods, and Visualizations for Complicance Management*. Bachelor's thesis, Fakultät für Informatik, Technische Universität München, 2008.
- [Ern10] Alexander M. Ernst. *A Pattern-Based Approach to Enterprise Architecture Management*. PhD thesis, Technische Universität München, München, Germany, 2010. (in submission).
- [FJ01] Ulrich Frank and J. Jung. The MEMO Organisation Modelling Language (MEMO-OrgML). Technical Report 26, Arbeitsberichte des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau, 2001.
- [Fra99] Ulrich Frank. Memo: Visual Languages for Enterprise Modelling. Technical Report 18, Arbeitsberichte des Instituts für Wirtschaftsinformatik Koblenz, 1999.
- [Fra02] Ulrich Frank. Multi-perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In *Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on System Sciences (HICSS 2002)*, pages 1258–1267, Washington, DC, USA, 2002.
- [Fra09] Ulrich Frank. The MEMO Meta Modelling Language (MML) and Language Architecture (ICB-Research Report). Technical report, Institut für Informatik und Wirtschaftsinformatik, Duisburg-Essen, Germany, 2009.
- [HV93] John Charles Henderson and N. Venkatraman. Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):472–484, 1993.
- [Int07] International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.
- [JE07] Pontus Johnson and Mathias Ekstedt. *Enterprise Architecture – Models and Analyses for Information Systems Decision Making*. Studentlitteratur, Pozkal, Poland, 2007.
- [JGBvB05] Henk Jonkers, L. Goenewegen, Marcello Bonsangue, and René van Buuren. A language for enterprise modelling. In Marc Lankhorst, editor, *Enterprise Architecture at Work*. Springer, Berlin, Heidelberg, New York, 2005.
- [JvBA<sup>+</sup>03] H. Jonkers, R. van Burren, F. Arbab, F. de Boer, M. Bonsangue, H. Bosma, H. ter Doest, L. Groenewegen, J.G. Scholten, S. Hoppenbrouwers, M.-E. Iacob, W. Janssen, M. Lankhorst, D. van Leeuwen, E. Proper, A. Stam, L. van der Torre, and G.V. van Zanten. Towards a language for coherent enterprise architecture descriptions. In *7<sup>th</sup> International Enterprise Distributed Object Computing Conference (EDOC 2003)*, Brisbane, Australia, 2003. IEEE Computer Society.

- [KL96] Wilhelm Kamlah and Paul Lorenzen. *Logische Propädeutik: Vorschule des vernünftigen Redens*. Metzler, Stuttgart, Germany, 3<sup>rd</sup> edition, 1996.
- [KW07] Stephan Kurpjuweit and Robert Winter. Viewpoint-based Meta Model Engineering. In Manfred Reichert, Stefan Strecker, and Klaus Turowski, editors, *Enterprise Modelling and Information Systems Architectures – Concepts and Applications*, *Proceedings of the 2<sup>nd</sup> International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07)*, St. Goar, Germany, October 8-9, 2007, LNI, pages 143–161, Bonn, Germany, 2007. Gesellschaft für Informatik.
- [Lan05] Marc Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Heidelberg, Germany, 2005.
- [LW04] Kerstin Langenberg and Alain Wegmann. Enterprise Architecture: What Aspect is Current Research Targeting? Technical report, Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2004.
- [OMG06] OMG. Object Constraint Language (OCL) Available Specification, version 2.0 (formal/06-05-01), 2006.
- [RWR06] W. Ross, Jeanne, Peter Weill, and C. Robertson, David. *Enterprise Architecture as Strategy*. Harvard Business School Press, Boston, MA, USA, 2006.
- [Sch08] Marten Schönherr. Towards a common terminology in the discipline of Enterprise Architecture. In Stephan Aier, Pontus Johnson, and Joachim Schelp, editors, *Pre-Proceedings of the 3<sup>rd</sup> Workshop on Trends in Enterprise Architecture Research*, pages 107–123, Sydney, Australia, 2008.
- [SW09] Joachim Schelp and Robert Winter. Language communities in enterprise architecture research. In *DESIRIST '09: Proceedings of the 4<sup>th</sup> International Conference on Design Science Research in Information Systems and Technology*, pages 1–10, New York, NY, USA, 2009. ACM.
- [The09] The Open Group. TOGAF "Enterprise Edition" Version 9. <http://www.togaf.org> (cited 2010-02-25), 2009.
- [vdRvV08] Bas van der Raadt and Hans van Vliet. Designing the Enterprise Architecture Function. In *4<sup>th</sup> International Conference on the Quality of Software Architectures (QoSA2008)*, pages 103–118, Karlsruhe, Germany, 2008.
- [WBL<sup>+</sup>05] Alain Wegmann, Pavel Balabko, Lê, Gil Regev, and Irina Rychkova. A Method and Tool for Business-IT Alignment in Enterprise Architecture. In *Proceedings of the CAiSE'05 Forum*, pages 113–118, Porto, Portugal, 2005.
- [Zac92] John Zachman. Extending and Formalising the Framework for Information Systems Architecture. *IBM Systems Journal*, 31(3), 1992.

# Roadmaps for Enterprise Architecture Evolution

Alexander M. Ernst, Alexander W. Schneider

Chair for Informatics 19

Technische Universität München

eMail: {ernst, schneial}@in.tum.de

**Abstract:** Managing the enterprise architecture (EA) is a major challenge of every larger modern enterprise. Aligning business and IT in order to optimize their interaction is the main goal of EA management. The first step of an EA management approach typically is to analyze and document the as-is-situation. Subsequently, the second step has to be the management of the EA evolution in order to achieve the desired target EA. Although IT projects guide this evolution by implementing the transformations from a current to an envisioned architecture, they are considered sparsely by EA approaches. Therefore, this article presents a methodology pattern (M-Pattern) describing a process including all necessary tasks and associated roles, which help to cope with the challenge of EA evolution management.

## 1 Introduction

Modern enterprises face the challenge to survive in an ever changing environment. In order to support the transformation of the enterprise according to new business demands, legal regulations, or changing market situations a holistic perspective on the enterprise architecture (EA) has to be taken. Thereby, EA is understood as the *fundamental conception of the system [enterprise] in its environment embodied in its elements, their relationships to each other and to its environment, and the principles guiding its design and evolution* [Int07]. Managing the EA is one of the major challenges of modern enterprises. It aims at aligning business and IT in order to optimize their interaction.

Documenting and managing the EA is an advanced topic, as the application landscape, which is part of the EA, often includes a few hundreds up to a few thousand business applications and their interconnections. Thereby, managing the EA is a task, that has to be executed as the need for a flexible IT is an integral concern of most companies. Another reason for the importance of EA management are regulations like e.g. the *Sarbanes Oxley Act (SOX)* [tCO2], which determine the information a company has to have available about its EA.

This article includes patterns which are part of the *EAM Pattern Catalog*, a *pattern language for enterprise architecture management* [BEL<sup>+</sup>07, BELM08, BEL<sup>+</sup>08, Ern08, BEMS09b, Ern09], which uses a pattern based approach to EA management. The complete *EAM Pattern Catalog* is available online at <http://eampc->

wiki.systemcartography.info/ [Cha09] and currently includes 164 EAM patterns<sup>1</sup>. The intention behind this article is to further extend the existing *EAM Pattern Catalog* by a methodology pattern (M-Pattern), documenting best practices for managing the evolution of an EA, in order to advance the EAM pattern language.

The *EAM Pattern Catalog* introduces four different types of patterns:

**M-Patterns** specify a methodology to address management problems in a stepwise manner. The procedures defined by the M-Pattern can be very different, ranging from visualizations and group discussions to more formal techniques as e.g. metrics calculations [LS08]. M-Patterns explicate the methodologies in order to complement activities carried out in an ad-hoc manner or relying on implicit knowledge with activities carried out more systematically.

**V-Patterns** provide visualizations like diagrams, reports, etc., which are practically proven to be adequate to address problems in EA management. The data required to produce the visualization is documented in one or more I-Patterns.

**I-Patterns** supply best-practice information model fragments, including definitions and descriptions of the used concepts, which can be used to collect information to address a certain problem in EA management.

**Anti patterns** document typical mistakes made in the context of EA management, and provide revised solutions in order to support the pattern user to prevent these pitfalls.

The rest of this section gives a short overview about the intended audience, the nature of patterns, and an overview about included EAM patterns.

## 1.1 Intended Audience

This article and the herein included patterns are intended for people concerned with planning the enterprise architecture evolution, people who have to manage a project portfolio, and people responsible for application life cycles.

## 1.2 Pattern-based approach to EA management

Patterns in general, and so do EAM Patterns, provide well-known and proven practice solutions for recurring problems in a given context. At first glance it might appear that there is no benefit in documenting such solutions. But the opposite is true because patterns are a valuable source of information for novice practitioners as well as a source for inspiration for experienced ones. This is one of the reasons for the adoption of the pattern concept in various disciplines. Ernst [Ern09] presents an in-depth introduction to the pattern-based approach to EA management.

---

<sup>1</sup>For a detailed explanation of the concept of EAM patterns refer to [Ern08, Ern09].

Typically, a pattern has to be found in practice at least three times to prove that it addresses a recurring problem and provides a proven practice solution. The EAM Pattern described in this article could, at the time of publication, be found in one company. Nevertheless, it can be expected that it holds for other contexts as well. Running and managing projects is a common task for companies. If there is more than one project accomplished at once, the described pattern can be used because it does not depend on the projects' topics.

### 1.3 Overview of included EAM Patterns

This article includes the M-Pattern EA DEPENDENCY GUIDED PROJECT MANAGEMENT. Figure 1 shows EA DEPENDENCY GUIDED PROJECT MANAGEMENT and its relationships to other EAM Patterns of the *EAM Pattern Catalog*. The figure also shows the concern addressed by EA DEPENDENCY GUIDED PROJECT MANAGEMENT and the topic the concern belongs to.

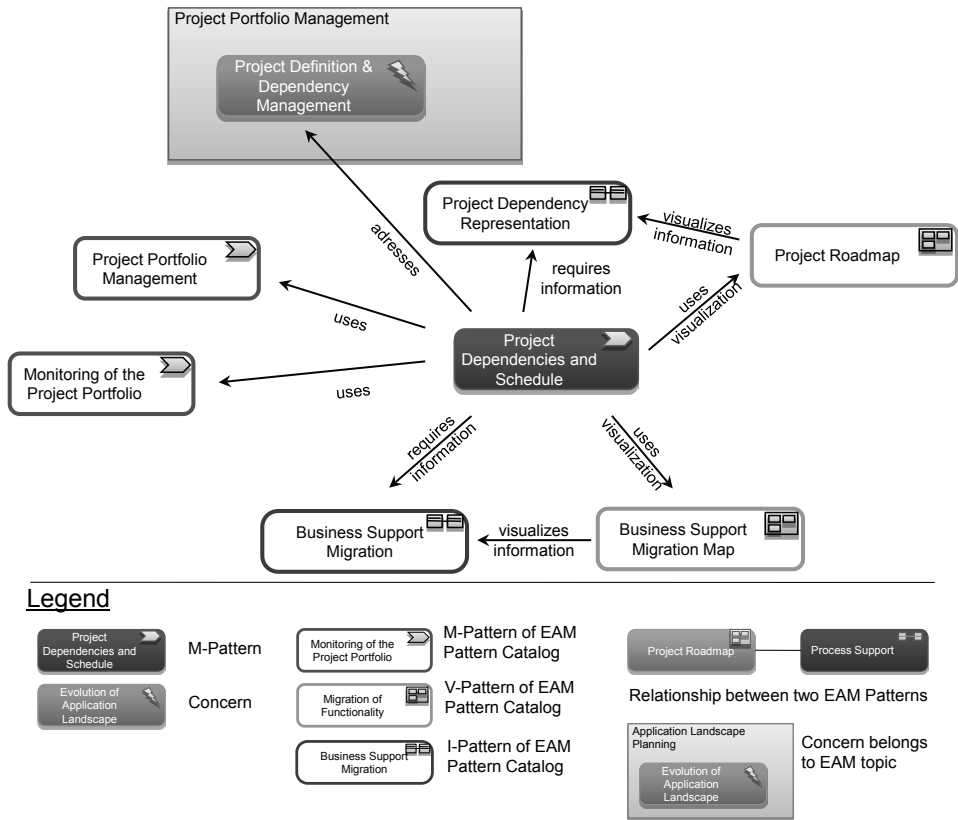


Figure 1: EAM Pattern Map



The following description gives an overview about the not included EAM Patterns:

**Project Dependency Representation** provides a structure for organizing information about projects and their interrelationships.

**Project Roadmap** provides a way to visualize information about projects, their progress, and their relations to each other. This includes project phases, milestones, and dependencies between projects.

**Business Support Migration Map** provides a way to visualize which application will take over functionality of other applications.

**Business Support Migration** provides a structure for organizing information about the EA evolution including the introduction and retirement of applications and the migration of business supports.

**Monitoring of the Project Portfolio** is concerned with the monitoring of the project portfolio.

**Project Portfolio Management** is used to analyze the project portfolio concerning the defined strategies.

Additional detail about those EAM patterns can be found in [Sch09].

## **2 EA dependency guided Project Management**

The objective of EA DEPENDENCY GUIDED PROJECT MANAGEMENT is identify all important dependencies between current and planned projects from an EA management point of view, in order to schedule projects and react on delays timely.

### **2.1 Example**

The MRAM project was considered to roll out three newly developed or adapted business applications at a specific organizational unit. The custom-built systems used at this location should be replaced by global homogeneous applications. The first application was called FAB and was responsible for contract initiations with customers. The second was called RM-NL and was responsible for contract administration. The third was called FS-RI and was responsible for the generation of the annual report of the company. In this order data included in contracts had to pass these systems. Because of the high complexity of the supported business processes, each application was developed in a separate project which consisted of several sub-projects. Of course, project managers were aware that their application has to import data from another application. But initially, they did not exactly define the data because they did not regard this as a major issue and postponed it. After applying EA DEPENDENCY GUIDED PROJECT MANAGEMENT, all project managers described their dependencies in detail and discovered that the contracts to be administrated by these applications are very individual and cannot be standardized. Therefore, the refinement of the application's interfaces caused a huge delay in all three projects.

### **2.2 Context**

You work in a company running multiple software development projects in parallel and you have to manage their dependencies from an EA management point of view. A multi project management approach might already be established in your company focusing typical tasks like the effective planning, evaluation, and control of strategic projects [Kun05]. The individual project schedules should not be determined by EA management beforehand. All projects to be regarded are already prioritized and have a fixed budget. For each project a project plan, e.g. according to the project management body of knowledge [Dun96], is in use.

### **2.3 Problem**

**Which tasks have to be performed to ensure an effective inter-project dependency identification to foster EA evolution management?**

The following *forces* influence the solution:

**Strict dependency management versus laissez-fair** Who is responsible for identified dependencies? Are projects allowed to handle their dependencies on their own or is there a need for a steering committee?

**Completeness of dependencies versus effort** How can you ensure the completeness of identified dependencies but keep the needed effort slight?

**One-time versus continuous approach** Should you manage changing project requirements and schedules continuously or by a one-time approach?

**All-embracing versus selective approach** How many projects can be regarded and how much information is appropriate to ensure manageability? Is it necessary to manage all projects or is it sufficient to consider critical projects?

**High versus low transparency** Does the constraint of public status reporting result in information withholding? Are project managers willing to provide the needed information or is there certain governance needed?

## 2.4 Solution

EA DEPENDENCY GUIDED PROJECT MANAGEMENT follows a four-step approach, which is shown in Figure 2. The steps **Dependency identification** and **Business support migration** form the analysis phase, which regards the projects in the portfolio context and their impacts to the EA. The second phase, covering the steps **Joint scheduling** and **CPM recomputing**, is an implementation phase which applies insights of previously executed phases.

There are at least three conceivable events that are considered to occur likely and result in a new iteration of EA DEPENDENCY GUIDED PROJECT MANAGEMENT:

- *New requirements from business emerge* due to changing processes or new strategies demanding for new projects to be set up.
- *Technical reasons* can require additional projects for new applications or infrastructure like operating systems, e.g. if the vendor's time of support is over.
- *EA management decisions* can result in new projects, for example to consolidate the application landscape.
- *Mergers and acquisitions* result in a program of integration projects which also have to be regarded from an EA point of view.

If one of the previously mentioned events occurs and a new iteration should begin all four steps have to be performed. In the following the respective steps are presented in detail:

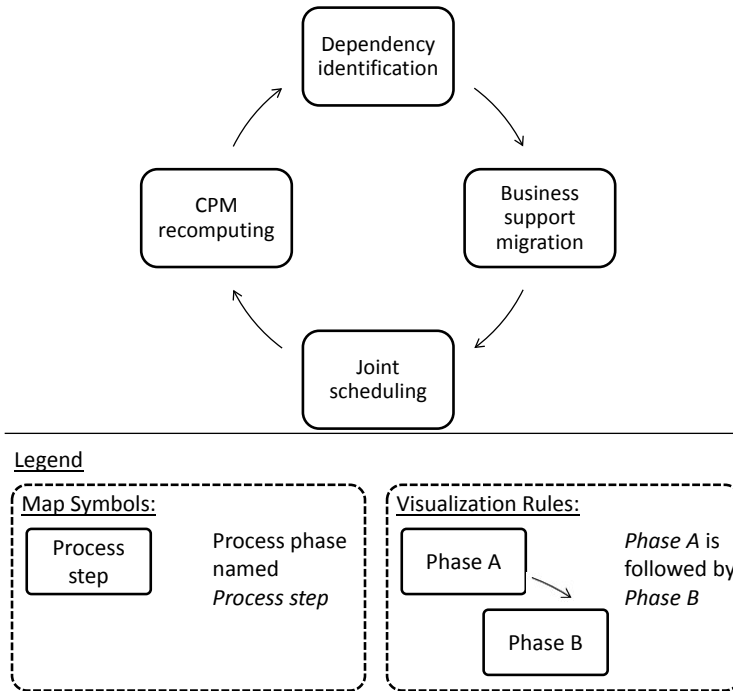


Figure 2: EA DEPENDENCY GUIDED PROJECT MANAGEMENT process

**Dependency identification** This step is as essential as challenging. The responsible project manager will find some dependencies during his planning activity, for example, that test data is needed from another project. Some dependencies like resources required at the same time might be identified by the multi project management. But there are also dependencies which are not that easy to identify. Therefore, this activity has to be performed separately by *project managers* and *IT architects* together. At least, they should regard three types of dependencies:

- **Organizational dependencies:** They result from the organizational structure and context of the company. For example, two projects are going to roll out their application at the same organizational unit at the same time and both estimate that the rollout will take the whole weekend. The two IT employees at that organizational unit might not be able to handle this. As a result, the two projects depend on one another because only one can rollout at this time. Another reason might be that the application which should deliver data to the project is not able to do this because it computes the annual accounts.
- **Interfaces between applications:** If a project uses data or functionality of another application currently developed, it depends on that projects ability to deliver.
- **Technical dependencies:** Usually, an application uses other software consid-

ered as infrastructure like, for example, databases or enterprise resource planning systems. A project depends on all other projects introducing or changing one of the used infrastructure systems.

- **Rollout dependencies:** In complex business processes applications often have to interact with each other. If more than one of them has to be changed, dependencies like *"has to be rolled out before X"* or *"has to be rolled out together with X"* can occur.
- **Retirement dependencies:** More often than not, new applications replace one or more old applications. If more than one new application is considered to retire such a legacy system, these new applications depend on each other, because they all need to be implemented before the retirement can begin.

In general, it is important, that all projects affected by a dependency formally commit to it. In addition, it is not enough to identify a dependency. Every dependency has to be described in detail so everybody is aware of all resulting impacts.

**Business support migration** This step might reveal additional dependencies and ensure the right handling of legacy systems, because new applications are not built on a greenfield, it is necessary to determine all applications or parts of them which are retired by the currently developed application. If a legacy system can be retired immediately, the retirement process might also require another project. If only a part of the functionality is adopted, it has to be decided which application adopts the other parts if they are still needed. If some functionality is not needed any more, this fact also has to be documented.

**Joint schedule** Based on the initial project prioritization the new projects can now be included in the overall schedule (roadmap). If the new projects have any impact to other projects, these impacts have to be communicated.

**Critical Path Method (CPM) recomputing** After a roadmap update, the critical path through all managed projects needs to be recomputed. That is the sequence of project tasks, which do not have any slacktime so a delay will cause the entire project or depending projects to be late [KW59]. The results might then have additional impacts to projects, which have to be applied and communicated. A method for critical path computing in a project portfolio environment is described in [MC10].

## 2.5 Implementation

In order to implement EA DEPENDENCY GUIDED PROJECT MANAGEMENT in your company, the process steps described in the solution section need to be assigned to different roles. Table 1 shows a RACI chart [Val10] which clarifies which role is involved in each activity. Overall there are five roles needed:

**CIO** The Chief Information Officer is a member of the executive board or reports directly to it and is in charge for the information technology (IT) of the company.

**Project portfolio management board (PPM board)** The project portfolio management board is responsible for tailoring and approving projects. They also prioritize projects according to the business or IT strategy and communicate rescheduling impacts.

**Project manager** A project manager is in charge for one or more projects of the project portfolio. He is responsible for the initial project data like phases and milestone dates.

**IT architect** The IT architect has the overview about the whole application landscape and is responsible for the consistent retirement of legacy systems and the adherence to the application landscape vision.

**Business unit** The business unit includes all people of a department. The business unit is involved in the business process supported by the application developed during the specific project. It can also be the project proposer.

	PPM board	IT architect	Project manager	Business units	CIO
Dependency identification	I	R/A	R		
Business support migration		R/A	C	C/I	
Joint scheduling	R/A			C/I	I
CPM recomputing	R/A				I

Table 1: RACI chart for EA DEPENDENCY GUIDED PROJECT MANAGEMENT

The different roles assigned in the RACI chart above are defined as follows [Val10]:

**(R) = Responsible** - owns the problem / project

**(A) = to whom "R" is Accountable** - who must sign off (Approve) on work before it is effective

**(C) = to be Consulted** - has information and/or capability necessary to complete the work

**(I) = to be Informed** - must be notified of results, but need not be consulted

### 2.5.1 Known Uses

EA DEPENDENCY GUIDED PROJECT MANAGEMENT is in use at Munich Re.

## 2.6 Consequences

**Strict dependency management versus laissez-fair** A strict dependency management will clarify accountabilities for dependencies but also raises costs. In order to achieve a holistic overview of dependencies, a relatively strict dependency management including a formal process of commitment should be used. The steering committee is accountable to ensure the commitment of project managers to identified dependencies. Of course, this approach causes some additional effort, but it is necessary to ensure that project managers address dependencies within their planning.

**Completeness of dependencies versus effort** Achieving a complete list of dependencies will minimize the risk of undiscovered dependencies but raises costs at the beginning of the project. Undiscovered dependencies can cause greater cost and delays at the end. Achieving a complete list of dependencies among projects requires an intensive communication between IT architects, project managers, and business units.

**One-time versus continuous approach** From an EA management point of view, which regards long-term perspectives, an instant notification of schedule changes is not needed. An interval of about one month seems appropriate for the consolidation of project changes to the joint schedule.

**All-embracing versus selective approach** Because even non-critical or very small projects can be affected by the critical projects, they also have to be included in a joint schedule. But if a project does not have any dependencies to other projects, including the rollout, it can be removed from the overall schedule in order to reduce its complexity.

**High versus low transparency** The transparency achieved by this solution is high, but also raises costs. Thus, the company gets a better overview about its projects. Because the process depends on data about individual project schedules, it is important to convince project managers about their benefits of delivering this data and so prevent information withholding.

## 2.7 See Also

When using EA DEPENDENCY GUIDED PROJECT MANAGEMENT also the results of PROJECT PORTFOLIO MANAGEMENT and MONITORING OF THE PROJECT PORTFOLIO can be of interest and should be considered, because they are also concerned with projects and their relations. The implementation of this pattern can be supported using PROJECT PORTFOLIO ROADMAP, BUSINESS SUPPORT MIGRATION MAP, PROJECT DEPENDENCY REPRESENTATION, and BUSINESS SUPPORT MIGRATION [Sch09].

### **3 Acknowledgment and Outlook**

This section includes acknowledgments to the people who supported the creation of this article and gives an outlook to the next steps in the development of the EAM pattern approach.

#### **3.1 Acknowledgments**

We want to thank all participants of the writer's workshop of PEAM 2010 and especially our shepherd Wolfgang Keller for the time they spent reading, commenting, and discussing this article. In addition we want to thank Ms. Dr. Walter and Mr. Perkens-Golomb for their support during the EAM Pattern documentation.

#### **3.2 Next Steps in EAM Pattern Approach Development**

The *EAM Pattern Catalog* is available at <http://eampc-wiki.systemcartography.info/>, based on the results of an extensive online survey, some articles on EAM patterns, see e.g. [Ern08, BEK<sup>+</sup>09, LFB<sup>+</sup>09, MJBS09, BEMS09a, BEMS09b], and a Ph.D. thesis [Ern09]. In order to improve the current version and to further exploit the advantages of patterns in EA management, an excerpt of the *EAM Pattern Catalog* had been included in this document to be discussed in the pattern community.



## References

- [BEK<sup>+</sup>09] S. Buckl, A. Ernst, H. Kopper, R. Marliani, F. Matthes, P. Petschownik, and C. M. Schweda. EAM Pattern for Consolidations after Mergers. In *SE 2009 - Workshopband*, pages 67–78, Kaiserslautern, 2009.
- [BEL<sup>+</sup>07] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Kathrin Schneider, and Christian M. Schweda. A pattern based Approach for constructing Enterprise Architecture Management Information Models. In *Wirtschaftsinformatik 2007*, pages 145 – 162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.
- [BEL<sup>+</sup>08] Sabine Buckl, Alexander Ernst, Josef Lankes, Florian Matthes, and Christian M. Schweda. Enterprise Architecture Management Patterns – Exemplifying the Approach. In *The 12th IEEE International EDOC Conference (EDOC 2008)*, Munich, 2008. IEEE Computer Society.
- [BELM08] Sabine Buckl, Alexander M. Ernst, Josef Lankes, and Florian Matthes. Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.
- [BEMS09a] Sabine Buckl, Alexander Ernst, Florian Matthes, and Christian Schweda. Enterprise Architecture Management Pattern for EA Visioning. In *Proceedings of EuroPloP 2009*, 2009.
- [BEMS09b] Sabine Buckl, Alexander M. Ernst, Florian Matthes, and Christian M. Schweda. How to make your enterprise architecture management endeavor fail! In *Pattern Languages of Programs 2009 (PloP 2009)*, Chicago, 2009.
- [Cha09] Chair for Informatics 19 (sebis), Technische Universität München. EAM Pattern Catalog Wiki. <http://eampc-wiki.systemcartography.info> (cited 2009-11-06), 2009.
- [Dun96] W.R. Duncan. A Guide to the Project Management Body of Knowledge: PMBOK Guide, 1996.
- [Ern08] Alexander Ernst. Enterprise Architecture Management Patterns. In *PloP 08: Proceedings of the Pattern Languages of Programs Conference 2008*, Nashville, USA, 2008.
- [Ern09] Alexander M. Ernst. *A Pattern-Based Approach to Enterprise Architecture Management*. PhD thesis, Technische Universität München, München, Germany, 2009. (in submission).
- [Int07] International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.
- [Kun05] Christian Kunz. *Strategisches Multiprojektmanagement Konzeption, Methoden und Strukturen*. PhD thesis, Universität Bamberg, Wiesbaden, Germany, 2005.
- [KW59] James E. Kelley, Jr and Morgan R. Walker. Critical-path planning and scheduling. In *IRE-AIEE-ACM '59 (Eastern): Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173, New York, NY, USA, 1959. ACM.

- [LFB<sup>+</sup>09] Armin Lau, Thomas Fischer, Sabine Buckl, Alexander M. Ernst, Florian Matthes, and Christian M. Schweda. EA Management Patterns for Smart Networks. In *SE 2009 - Workshopband*, pages 79–90, 2009.
- [LS08] Josef Lankes and Christian M. Schweda. Using Metrics to Evaluate Failure Propagation and Failure Impacts in Application Landscapes. In *Multikonferenz Wirtschaftsinformatik*, Berlin, Germany, 2008. GITO-Verlag.
- [MC10] Wolfram Müller and Jens Coldewey. Critical Chain Project Management. *Object Spectrum*, 2010.
- [MJBS09] Christoph Moser, Stefan Junginger, Matthias Brückmann, and Klaus-Manfred Schöne. Some Process Patterns for Enterprise Architecture Management. In *Software Engineering 2009 – Workshopband*, pages 19–30. Lecture Notes in Informatics (LNI), 2009.
- [Sch09] Alexander Schneider. Roadmaps for Enterprise Architecture Evolution. Bachelor’s Thesis, Technische Universität München, München, Germany, 2009.
- [tC02] 107th Congress. Sarbanes-Oxley Act of 2002 (Public Law 107204). <http://www.sec.gov/about/laws/soa2002.pdf>, 2002.
- [Val10] Value Based Management.net. RACI Model. [http://www.valuebasedmanagement.net/methods\\_raci.html](http://www.valuebasedmanagement.net/methods_raci.html), 2010.



# EA Management Patterns for Future State Design

Chris Aitken

Queensland Investment Corporation  
c.aitken@qic.com

**Abstract:** Developing a future state design for any enterprise component is a crucial step in the enterprise management lifecycle. Traditional approaches rely heavily on business stakeholder participation and input. Several problems are identified with this approach which may contribute to poor design decisions. This paper introduces two EA Management patterns which aim to incorporate both stakeholder requirements and design principles to promote elegant future state designs. The paper includes a means for both documenting and assessing the application of design patterns, a description of the patterns of good design, and a method for applying these to a future state.

## 1 Introduction

At some point in most enterprise architecture endeavours there is a need to develop a description or model of a future state solution. The need to develop such a design can arise at both the level of the enterprise as a whole or within the context of individual business solutions. A future state solution needs to meet the requirements of the enterprise at both strategic and operational levels and enable the enterprise to meet its strategic goals and objectives. The Open Group Architecture Framework (TOGAF) Version 9.0 [TO09a] is an example of a commonly used EA methodology. TOGAF relies heavily on the use of Business Scenarios as a means to identify business requirements and as a mechanism to identify aspects of a future state design. Conventional wisdom regarding the development of future state solutions usually includes a process of eliciting requirements and ideas via architect-hosted interviews or workshops with senior business stakeholders. In these sessions business stakeholders are encouraged to envisage or brainstorm ideal future scenarios for the enterprise and its clients. Options are then listed and prioritised. The architects then model a solution, based on their understanding of current technology, and which satisfies the ideal scenarios and requirements identified through the interviews and workshops.

However, there are at least two problems with this approach and its considerable reliance on business stakeholder input. The first issue is that each business stakeholder is likely to approach the exercise of „future state envisaging“ through the filter of their own set of priorities. As a result, it can be difficult to achieve a whole-of-enterprise perspective amongst business stakeholders who all have their own priorities and requirements. Furthermore, it should be remembered that some of the business stakeholders may have

been directly responsible for the current state with all its problems and shortcomings. Therefore, a future state solution based solely on stakeholder input is likely to be influenced by stakeholder competing priorities, biases and pre-conceptions about what is possible.

The second problem concerns the potential conflict between stakeholder requirements and good design. This potential conflict arises because business stakeholders are typically not designers or engineers but rather are responsible to manage the enterprise. For the purposes of this paper, good design is defined as the combination of form and function to elegantly and completely solve a given problem. The concept of good design encompasses design efficiency, effectiveness and simplicity or „economy of design. There are well understood engineering principles which promote good design and design elegance. However, these principles will act to constrain the set of all possible solutions



Figure 1. The ‘Winchester House’

to a given problem. This constrained set of solutions may not include those identified through a stakeholder consultation process like the one outlined above. An extreme example of this potential conflict is the often cited “Winchester House” [WI10] in which user requirements were followed blindly and without regard for principles of good design. The original owner, supposedly guided by spirits over a period of 38 years, oversaw the construction of the 160 room sprawling mansion. The resulting structure contains many examples of incoherent design.

Clearly, there is a need to identify an enterprise management pattern which accommodates the requirements and objectives of business stakeholders, and which also ensures that the future state design reflects the application of the principles of good design. To address this need, the current paper sets out two enterprise architecture management patterns. The first pattern (Design Principle Definition and Assessment) addresses the issues of how to describe design principles generally, and how to assess the extent to which a given design reflects the application of a given principle. The second pattern (Future State Solution Design) introduces the principles of good design, and outlines how these are used together with stakeholder requirements to identify elegant future state designs.

## 2 Pattern 1 - Design Principle Definition and Assessment

The concept „design“ is a logical level concept in that a design does not need to include implementation detail [AI09]. Indeed, a design description may actually be more useful if implementation level detail is purposely excluded. By this it is meant that a single design may be physically implemented in multiple ways.

For the purposes of this pattern, design principles are defined as statements which when followed or applied influence and constrain a design. It is recommended that design principles be used to constrain and guide the development of designs. Whereas, standards be used to constrain the implementation options of a given design. In the context of enterprise architecture, design principles should be applied to enterprise level and individual solution designs. Architectural standards on-the-other-hand should be applied to the specified implementations of these designs.

This methodology pattern describes a means to document design principles that supports the assessment of their application in any given design. The pattern is based on and extends the pattern described in [TO09b].

### Intended Audience

This EA management pattern is intended to assist with the definition and application of design principles. Therefore, the pattern will be of interest to those involved in enterprise level Information Communication Technology (ICT) planning and portfolio management, as well as those involved in the development of specific business solutions (i.e. Solution Architects).

### Known Uses

The following are examples of the use of this pattern.

- Queensland Investment Corporation (QIC) EA Principles document [QI09a]. This documents sets out the principles to guide the development of QIC future state designs, and the implementation of enterprise architecture within the organisation.
- QIC Conceptual Solution Design (CSD) document [QI09b]. A CSD is developed for each IT solution within QIC. The CSD lists those principles which were applied to arrive at the given solution design.
- Queensland Health Department EA Principles document [QU08].
- TOGAF V9.0 [TOG09b]. This widely adopted EA methodology framework uses elements of this pattern.

### Context

There exists an enterprise which requires a design. The design needs to be developed to reflect specific design principles and evaluated in terms of its ability to demonstrate the application of these principles.

## Problem

In order to evaluate whether a given design represents the application of a given principle it is necessary to be able to identify the principle, and assess the extent to which the given principle has been applied to the design. How can the application of a given design principle be assessed?

The solution will be influenced by the following forces.

## Forces

*Consistency of interpretation:* To ensure that designers identify and pre-scribe principles in a consistent fashion it is necessary that principles are clearly and succinctly defined. Design principles usually incorporate a number of concepts. To ensure consistency it is necessary to ensure that the concepts relevant to a given principle outlined in a standard fashion.

*Measurement:* The ability to quantify the requirements of each principle is necessary to enable designers to be able to assess the application of a given principle to a particular design.

*Governance:* Principles require some position of accountability to ensure appropriate application within an enterprise.

## Solution

Each principle is documented using the following format:

Name	The name of the principle. The name should be able to be prefaced with “The principle of...”
Statement	A concise statement of the principle.
Rationale	A statement of the rationale behind the principles, and why it is important.
Requirements	A description of the characteristics of a design to which the principle has been applied.
Assertions	A set of quantifiable and testable statements which can be answered true or false when applied to a design, and which relate to one or more requirements of the principle.
Sponsor	The role or position within the enterprise that is responsible for the maintenance of the principle definition and oversight of its application within the enterprise (e.g. Chief Enterprise Architect or Chief Information Officer).

References	One or more references to either examples, supporting research or other authoritative sources.
------------	--

Table 1. Principle documentation template

The relevant design principles are identified by the architect. The assessment of the design is undertaken by determining how many principle assertions are satisfied by the given design. The design may then be rated as „Complete Implemented“, „Partially Implemented“, or „Not Evident“ for each principle.

**Consequences**

**Benefits:**

- Comparison of designs. This pattern provides a mechanism to evaluate and compare designs in terms of their ability to demonstrate the application of specific design principles.
- Consistency. The pattern promotes consistency of interpretation by standardising the specification of design principles.
- Assessment and governance. The pattern also enables the assessment of the degree to which a particular principle has been applied to a design, and promotes specific governance of the application of individual design principles through the use of a sponsor role.

**Drawbacks:**

- Resources. The pattern requires that significant time and resources be allocated to the development and refinement of the definition of principles
- Assessment. The ability to assess the application of a given principle rests on the objectivity of the assertion statements within the definition. This requires that these statements be carefully constructed and reviewed to ensure that they will be able to be understood and applied in a wide variety of contexts.
- Assessment granularity. This pattern currently only provides for three levels of application. This may well be too coarse grained to provide useful guidance. For example, there may be little in the way of useful guidance provided to the practitioner if all considered designs are assessed as “Partially Implementing” a given principle.
- Competing requirements. Conflicting requirements may arise in situations where multiple design principles are applied. This situation requires that the architect in consultation with business stakeholders make some assessment as to the relative importance of the respective principles.
- Inconsistent terminology. The wording of principles needs to undergo thorough review and evaluation to ensure that the same principle is not being described in a different way or using different terminology. Design principles need to be



worded in a manner that promotes easy identification and applicability across a wide range of contexts.

### **See Also**

EA Management Pattern for Future State Design

EA Management Pattern - M-4 Standard Conformity Management [EA10a]

## **3 Pattern 2 - Future State Solution Design**

This methodology pattern describes an approach to developing a future state design. The pattern incorporates business stakeholder requirements and the principles of good design. It should be noted that this pattern does not specifically address the gathering, documentation and analysis of business stakeholder requirements. It is recommended that these issues be addressed in a separate EA Management Pattern.

### **Intended Audience**

This EA management pattern is intended to assist with the development of future state designs. Therefore, the pattern will be of interest to those involved in enterprise level ICT planning and portfolio management, as well as those involved in the development of specific business solutions (i.e. Solution Architects).

### **Known Uses**

- QIC Conceptual Solution Design (CSD) document [QI09b]. A CSD is developed for each IT solution within QIC. The CSD lists those principles which were applied to arrive at the given solution design.
- QIC EA Methodology and Governance Framework [QI10]. This document outlines the methods used to implement and govern EA within QIC.
- Methodologies and paradigms which strongly reflect aspects of this pattern are:
  - Object Orient Programming
  - Service Oriented Architecture
  - Lean Business Process Management
  - Coherency Management [DO08]

### **Context**

There exists an enterprise in which ICT or enterprise planning is undertaken for which a future state design is required. The design may be for the enterprise landscape as a whole or a component of this. This pattern is one of a number of methodology patterns that together outline a methodology for enterprise planning. The processes of ICT or enterprise planning, and business stakeholder requirements analysis are not described within this pattern. 6

## **Problem**

A future state design needs to address both function and form requirements. How does an enterprise architect identify a future state design which reflects the principles of good design from amongst the range of all possible designs?

The solution is influenced by the forces listed below.

## **Forces**

*Addressing business stakeholder requirements:* Stakeholder requirements will be reflected in the solution functionality. Stakeholder requirements can also affect the form of the solution. Traditionally, these have been described as functional and non-functional requirements.

*Applying principles of good design:* There are many design principles which might be selected to guide the development of a given design. However, there are certain principles which are evident in all good designs. These principles are reflected in features of a solution design that address effectiveness, efficiency and form. Good designs have longevity, and are functionally effective and efficient. Design longevity is achieved through maintainability, flexibility and completeness. Moreover, the principles of good design need to apply to the design as a whole as well as its components. It is these principles of good design that this pattern seeks to identify.

*The role of enterprise specific design principles:* In addition to generic principles of good design there are also likely to be design principles that are specific to an individual enterprise or practice domain (e.g. software development, building construction). The principles of good design listed in this pattern should be used to supplement any enterprise specific principles. Enterprise specific design principles should be documented using the EA Management Pattern - Design Principle Definition and Assessment.

*The role of architectural standards:* Such standards will act to limit the possible implementations of a given the future state design. The architectural standards will identify which technologies or methodologies will be used.

## **Solution**

The solution to this problem follows an iterative five phase methodology. The first phase identifies business stakeholder requirements. The second phase develops one or more possible future state designs. The third phase is concerned with the application of principles of good design, and any enterprise specific design principles to identify a preferred future state design. The fourth phase provides a means for stakeholders to provide feedback on the preferred design. The fifth phase involves the assessment of proposed design against architectural standards to identify the preferred implementation.

The scope of iterations will vary depending on the underlying development paradigm. Agile approaches will typically use this pattern with shorter iterations and with a strictly

constrained scope. „Waterfall“ methodologies are likely to make use of the pattern but with iterations of much longer duration and broader scope.

**Phase 1 - Stakeholder requirements:** Requirements are gathered from an analysis of relevant strategic documentation. Further requirements are added to this initial set through direct engagement with stakeholders via interviews or workshops. Through this process stakeholders should be encouraged to envisage their version of the ideal solution. Where this pattern is applied at the level of individual business solutions, stakeholders should also be encouraged to „walk through“ the solution in terms of the actors involved and the outcomes and behaviour they could expect.

A key output of this phase is the identification of the purpose or objective of the future state design. The stated purpose, together with the requirements gathered fully specifies the problem space to be addressed by the future state design.

**Phase 2 – Generate future state design alternatives:** In this phase the architect is responsible to develop several future state designs that meet the stakeholder requirements. The extent to which a given stakeholder requirement is addressed may vary between the proposed designs. Any design that meets stakeholder requirements should be considered for inclusion at this point. The architect is not as concerned with the relative merits of each design during this phase as with the need to identify as many designs as possible within the time available. Within an agile approach consideration should be given to selecting alternatives that address the immediate problem only and specifically do not address issues that might be thought to be likely to arise as development is further progressed. The principles of good design and any enterprise specific design principles should be used as guides to narrow the number of potential designs considered during this phase.

**Phase 3 – Application of the principles of good design:** The application of the principles of good design is an iterative process in which greater compliance with the requirements of the principles is achieved in successive iterations as the initially proposed future state designs are further refined.

Each of the principles identified below is applied to address specific characteristics of good design. In summary, the Principle of the Separation of Concerns [RE89] is used to identify „logical“ components within the design that each address discrete issues, features or characteristics within the problem space. Consideration of component autonomy will determine the logical boundaries between components, or groupings of components within the design. The Principle of Optimisation [PA00] is applied to the logical components to determine the configuration of maximal efficiency and effectiveness. Note that the efficiency of a design can be considered separately to the efficiency of a particular implementation of that design. The Principle of Coherency (see [DO08] for enterprise-wide application, and [CO68] for software application) is applied to identify unnecessary redundancy or complexity of the design configuration or logical components. This principle ensures that all components contribute to the accomplishment of the design’s overall purpose.

The principles of good design are listed below using the EA Management Pattern - Design Principle Definition and Assessment.

Name	Principle of the Separation of Concerns
Statement	A design should be comprised of a set of independent components each of which addresses a discrete function within the problem space.
Rationale	This principle addresses the issues of component delineation and modularity. This principle promotes a designs that are resilient, responsive to change and easily maintainable.
Requirements	<ul style="list-style-type: none"> <li>• Components within a solution should be designed to be autonomous and include sufficient redundancy and management of error conditions to minimise the impact of failure on other components.</li> <li>• Interfaces between components should be designed to promote loose coupling between components to: <ul style="list-style-type: none"> <li>○ reduce dependencies on other components or entities external to the solution.</li> <li>○ promote compose-ability and facilitate re-orchestration of components.</li> <li>○ promote ease of maintenance and longevity by enabling straightforward exchange or replacement of components.</li> </ul> </li> </ul>
Assertions	<ul style="list-style-type: none"> <li>• There is a one-to-one matching between design components and functions within the problem space.</li> <li>• Replacement of any one component does not compromise the design as a whole.</li> <li>• Interfaces between components comply with documented standards.</li> </ul>
Sponsor	Enterprise Architect
References	[RE89]

Table 2. Principle of the Separation of Concerns

Name	Principle of Optimisation
Statement	A design should provide a complete solution for the problem space and achieve economy of both components and resource usage.
Rationale	This principle promotes designs that are both effective and efficient.

Requirements	<ul style="list-style-type: none"> <li>• The solution design should address as much of the problem space as possible.</li> <li>• Each component within a design should encapsulate a single discrete function within the complete problem space.</li> <li>• The solution designs should minimise consumption of resources and execution time.</li> <li>• Components should be designed to promote appropriate cost effective re-use.</li> <li>• Components should be re-used where possible.</li> </ul>
Assertions	<ul style="list-style-type: none"> <li>• The problem functional space is completely supported by the solution design.</li> <li>• There are no unnecessary or redundant components within the design.</li> <li>• All opportunities to re-use components are realized within the design.</li> <li>• Given the scope of the problems functional space addressed by the design, the design is evidently the most efficient in terms of resource consumption and execution time.</li> </ul>
Sponsor	Enterprise Architect
References	[PA00]

Table 3. Principle of Optimisation

Name	Principle of Coherency
Statement	The design components should be harmonised with the design purpose.
Rationale	This principle addresses the issue of alignment of components with the purpose of an entity as a whole. This principle acts to balance the design characteristics of the previous two principles. This principle also addresses the issue of sufficient complexity. That is, designs maybe complex, but only to the degree to which is required to achieve the solution design's purpose. Unnecessary complexity or functionality does not represent good design.
Requirements	<ul style="list-style-type: none"> <li>• The components of the solution design should work together as a whole to achieve the design purpose.</li> <li>• A design should include only components that support the design purpose.</li> <li>• Design complexity and functionality is driven by requirements of the solution design.</li> <li>• Each component should be able to demonstrate a relationship</li> </ul>

	(i.e. .line of sight.) with one or more requirements of the solution design.
Assertions	<ul style="list-style-type: none"> <li>• The solution design has a documented and well understood purpose.</li> <li>• The solution design requirements derive directly from the design purpose.</li> <li>• Each component has a ‘line-of-sight’ with at least one requirement of the solution design.</li> </ul>
Sponsor	Enterprise Architect
References	[DO08], [CO68]

Table 4. Principle of Coherency

The application of each of these principles is assessed in terms of the degree to which its requirements are evident in the particular future state design under consideration. Assessment is conducted as described in the EA Management Pattern - Design Principle Definition and Assessment.

This phase should also see the application of any enterprise specific design principles. These principles too should be applied using the EA Management Pattern - Design Principle Definition and Assessment.

The key output from this stage will be the preferred future state design. This design will reflect the application of the principles of good design, and any enterprise specific design principles to the greatest extent.

**Phase 4 – Stakeholder review:** This phase provides the business stakeholders with an opportunity to provide feedback on the preferred design. The review may be a formal review or conducted informally. Stakeholders should be encouraged to „test the design to ensure that their particular requirements are satisfactorily addressed.

Where stakeholder feedback indicates that there are concerns about the preferred design the architect will need to review the concerns and ensure they are addressed by returning to undertake a further iteration of Phase 3. In this instance there should also be an additional stakeholder review. This review may be limited to only those stakeholders who initially raised concerns.

**Phase 5 – Design implementation:** In order to determine how the preferred design is to be implemented within the enterprise it is necessary to assess it against the relevant enterprise architecture standards. This assessment will identify which technologies and/or methodologies will be used to implement the design. The key output from this phase is a design implementation specification.

At the conclusion of the methodology a preferred design and its implementation should have been identified and documented. In addition, the preferred design will have been assessed in terms of its completeness, and longevity through the application of the principles of good design.

## **Consequences**

### **Benefits:**

- This pattern promotes the development of elegant future state solution designs by identifying the principles of good design, and providing a means to evaluate their application.
- The pattern provides a means to compare a number of future state designs and identify a preferred design.
- The pattern provides a means to balance stakeholder requirements with the requirements of the principles of good design.
- The pattern conceptually distinguishes between the role that principles and standards play in the development of future state designs and their implementation.

### **Drawbacks:**

- Phase sequencing. The sequence of phases described in this pattern is idealised and presented as a guide. In real world use the sequence is likely to be iterative and non-linear. The precise sequencing and the number of iterations will need to be determined by the architect and the underlying development paradigm.
- Resources: The application of this pattern is likely to require significant time, resources, and expertise on the part of the architect.
- Comprehensiveness of the principles: This pattern is likely to require further refinement over time as there are possibly aspects of good design that are not adequately addressed by the principles as currently identified or worded. Similarly, the listed requirements or assertions within an individual definition may be incomplete. However, the current principles have been developed after review of a number of sources.
- Generation of alternatives: This is a crucial phase in the methodology, and is entirely reliant on the expertise and experience of the architect. Preference needs to be given to alternatives that do not „lock in“ solution implementations, but rather leave a number of alternative viable pathways open.
- Generation of equal alternatives: The pattern may result in the identification of two equally meritorious future state designs. In this situation it is recommended that the architect consult with business stakeholders to determine the preferred design.
- Addressing stakeholder requirements: There is a need to develop an EA Management Pattern for the eliciting, collating and analysis of business stakeholder requirements.
- The role of architectural standards: There is likely to be a tension between the preferred design and architectural standards. However, this can be minimised

where there is a clear distinction made between standards for implementation (i.e. architectural standards), and principles of good design. Ideally, architectural standards should not constrain design, but rather provide strict guidance on the implementation requirements of a given design. Even so, it is possible that a preferred design may become untenable in light of the implementation requirements of the relevant architecture standards.

### **See Also**

EA Management Pattern - M-4 Standard Conformity Management [EA10a]

EA Management Pattern - M-14 Development of Plan and Target Landscape [EA10b]

EA Management Pattern for Design Principle Definition and Assessment

## **4 Conclusion**

The methodology patterns presented in this paper have been developed to provide guidance regarding the principles of good design and to formalise the use of design principles. The patterns provide both a means to apply the principles to future state design and a means to evaluate their application. This is an important step within the overall EA lifecycle, but one on which there is often limited available guidance. Traditional approaches place an understandable emphasis on the use of stakeholder requirements to define future state design. However, several issues have been identified in this paper where stakeholder requirements provide the sole input to the design process. Importantly, it should be noted that the patterns do not negate the need to gather and identify user requirements.

The Future State Solution Design pattern is extensible, and there are undoubtedly other design principles that might be considered for inclusion. However, in the author's experience, a number of other design principles (e.g. loosely coupled components, component autonomy, component redundancy etc.) are ultimately extensions or requirements of the principles currently included in the pattern. To this end a further refinement of the pattern might be to identify the current principles as a core primary set, and then to identify additional secondary principles that are derived from the primary set.

Although the current patterns and principles address a number of design „themes“ within a number of well established design paradigms, they have had limited use as a formalised pattern. Furthermore, the patterns described in this paper are not a replacement for architectural design expertise. Therefore, the patterns as described within this paper will benefit from further research, testing and refinement. However, given the significance of the process of future state design within the field of enterprise architecture the importance of these patterns within an overall EA Management framework is clear.



## References

- [AI09] Aitken, C. J., (2009), Design Integrity and EA Governance. In Pallab Saha (Ed), Advances in Government Enterprise Architecture, Hershey, PA: Idea Group.
- [CO68] Larry Constantine, Segmentation and Design Strategies for Modular Programming.” In T. O. Barnett and L. L. Constantine, eds., Modular Programming: Proceedings of a National Symposium. Cambridge, Mass.: Information & Systems Press, 1968.
- [DO08] Gary Doucet, John Götze, Pallab Saha, Scott Bernard, Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance, Journal of Enterprise Architecture, May 2008, Volume 4, Number 2.
- [EA10a] <http://www.matthes.in.tum.de/wikis/eam-pattern-catalog/m-4>, retrieved Jan 19, 2010.
- [EA10b] <http://www.matthes.in.tum.de/wikis/eam-pattern-catalog/m-14>, retrieved Jan 19, 2010.
- [PA00] Principles of Optimal Design: Modeling and Computation 2nd Ed. by Panos Y. Papalambros and Douglass J. Wilde, Cambridge University Press, New York, 1988, 2000.
- [QI09a] QIC EA Principles, unpublished document, Queensland Investment Corporation, 2009.
- [QI09b] QIC Conceptual Solution Design (CSD), unpublished document, Queensland Investment Corporation, 2009.
- [QI10] QIC EA Methodology and Governance Framework, unpublished document, Queensland Investment Corporation, 2010.
- [QU08] Queensland Health Department EA Principles, unpublished document, Queensland Health 2008.
- [RE89] Reade, Chris (1989). Elements of Functional Programming. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- [TO09a] The Open Group Architecture Framework <http://www.opengroup.org/architecture/togaf9-doc/arch/> , retrieved Jan 8, 2010.
- [TO09b] Architecture Principles [http://www.opengroup.org/architecture/togaf9-doc/arch/chap23.html#tag\\_24\\_01](http://www.opengroup.org/architecture/togaf9-doc/arch/chap23.html#tag_24_01), retrieved Mar 9, 2010.
- [WI10] <http://www.winchestermysteryhouse.com/> , retrieved Jan 8, 2010.

# I-Pattern for Gap Analysis

Philipp Gringel, Matthias Postina

Software Engineering for Business Information Systems  
OFFIS - Institute for Information Technology  
Escherweg 2  
26121 Oldenburg, Germany  
{philipp.gringel, matthias.postina}@offis.de

**Abstract:** Patterns are an effective way to express solutions to constantly occurring problems in software engineering. Books like Erich Gamma's *Design Patterns* ([Gam94]) or Martin Fowler's *Analysis Pattern* ([Fow96]) helped to develop a common understanding of problem solving in this field. The pattern catalog developed by members of the chair Software Engineering for Business Information Systems (se-bis) of TU-Munich ([BELM08]) addresses patterns related to reoccurring problems in enterprise architecture management (EAM) and the authors are still working on this catalog. [BELM08] mentions several concerns related to SOA but is missing patterns concerning the initiation of SOA. This contribution will elaborate on an I-Pattern related to SOA-initiation on the basis of [EHH<sup>+</sup>08]. The focus lies on the problem of gap analysis on application landscapes as stated in [PSS09] and will further add metrics for gap analysis to this I-Pattern.

## 1 Introduction

The paradigm of service-oriented architecture (SOA) has undergone a remarkable change. Considered as a technical solution for a long time, SOA is today regarded as a conceptual approach able to align enterprise IT systems with the business strategies and processes they are supposed to support. SOA success stories in large enterprises indicate that service orientation is an effective way to integrate legacy systems as reusable building blocks into a flexible enterprise architecture. However, especially for small and medium sized enterprises, SOA is a far cry from being already implemented as mainstream. So strategies for initiating SOA in enterprises (like [EHH<sup>+</sup>08], [AGA<sup>+</sup>08]) are still subject of interest. The pattern catalog ([BELM08]) addresses concerns related to SOA (C-51, C-61, C-62, C-64, C-65, C-66, C-67, C-71 for example) but is missing patterns concerning the initiation of SOA. Our contribution will elaborate on an I-Pattern supporting a specific aspect of the process of SOA-initiation on the basis of Quasar Enterprise ([EHH<sup>+</sup>08]). We focus on the problem of gap analysis on application landscapes as stated in [PSS09] and will further add metrics for gap analysis to the I-Pattern.

### 1.1 Intended Audience

This article about I-Pattern for Gap Analysis is intended for people concerned with planning the enterprise evolution - like an enterprise architect for instance. The described pattern focuses on the problem of performing a gap analysis between two states of the application landscape, where the current state has a pre-SOA status and the envisioned state should be designed according to the principles of Quasar Enterprise.

## 2 Gap Analysis Concern

The term gap analysis is used in context of enterprise architecture as a name for the comparison between two architectures or strictly speaking two states of the same architecture. The Open Group Architecture Framework (TOGAF) uses the terms of baseline- and target-architecture for these two states ([The09]).

In our case we follow the methodology of [EHH+08] and focus on SOA initiation, so we compare the current (non-SOA) landscape of an organization to an envisioned landscape (Ideal-SOA). The concern of gap analysis in this context means to understand the architectural differences between current and ideal landscape in order to get an idea of the characteristics which have to be improved on the way towards an ideal service-oriented landscape. When we use the term ideal, we mean ideal in terms of being designed corresponding to the principles of Quasar Enterprise.

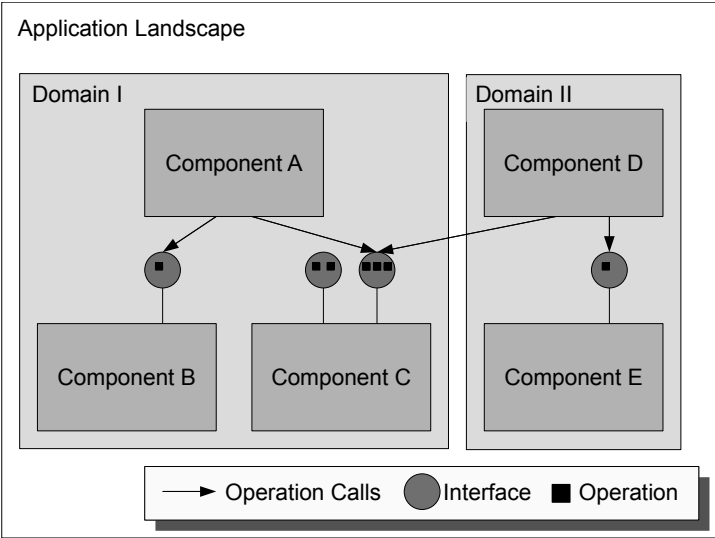


Figure 1: Ideal Application Landscape - Schematic Overview

Figure 1 provides a schematic overview of an ideal application landscape designed in ac-

cordance to [EHH<sup>+</sup>08]. Thereafter, service capabilities of an organization are provided by application components as operations encapsulated by services and described by interface descriptions. Application components are grouped by domains and domains are designed by functional decomposition of the enterprise. This means that [EHH<sup>+</sup>08] consider SOA initiation as a business driven process. Concerning gap analysis, we are observing the structural deviation between the current and ideal state of the application landscape.

To establish this understanding, we define in section 3 the data-structures describing the relationship between current and envisioned landscape as an I-Pattern. Based on this pattern, we present structural metrics which are used to quantify the differences between landscape states. Basically, we follow the definitions of [EHH<sup>+</sup>08] and partially extend the metrics defined there.

## 2.1 Forces

The problem of gap analysis on application landscapes described in this pattern could be summarized by the following question:

**How do I need to restructure the current application landscape to converge towards an ideal application landscape designed according to the principles of Quasar Enterprise?**

The following forces influence gap analysis on application landscapes:

1. **Qualitative assessment and quantitative assessment** The outcome of the gap analysis is an action list for possible measures to overcome the structural deviation between current and ideal application landscape. It deserves mentioning that this list reflects the quantitative assessment only and qualitative assessment could possibly complement this.
2. **Considering entire application landscape or focus on selective components** The gap analysis focuses on operations as atomic building blocks to detect structural deviations between current and ideal application landscapes. This is a detailed level of description when it comes to modeling both landscapes. The amount of needed data has to be considered carefully.
3. **Regarding state snapshots for gap analysis or monitor the distance over time** When performing gap analysis, two states - current and ideal application landscapes - are compared. Measures of the action list correspond to the outcome of this comparison. The architect needs to consider the circumstances of architectural work. Is the gap analysis part of a strategic enterprise architecture initiative or is it executed for a single comparison only?

### 3 I-Pattern

”An I-Pattern contains an information model fragment including the definitions and descriptions of the used information objects” ([BELM08]). According to this definition we wrote down an I-Pattern to support the gap analysis concern.

Following TOGAF and Quasar Enterprise, the **ideal application landscape** (ideal AL) should be designed according to the future / ideal business architecture. So the top level of an ideal AL is organized by domains grouping business functionality. Domains are nestable and are hosting application landscape components. These components can be regarded as - possibly only logically existing - application systems. A component provides public interfaces bundling a number of operations in order to implement a number of application services which are used by business services inside a business process dealing with business objects. Furthermore, operations are assigned to categories, which are ordered as follows:

1. *Interaction* services dealing with user interaction or interaction with other ALs.
2. *Process* services supporting business processes.
3. *Function* services supporting business processes and having an algorithmic character.
4. *Data* services accessing stored data.

This categorization is a typical means for structuring service-oriented landscapes and can also be found in other service-oriented approaches (see e.g. [KBS06]). Over the application service interfaces, the categorization can also be propagated down to the operations’ level and is, by definition of the term ideal AL in [EHH<sup>+</sup>08], also applicable for components. The ideal AL includes only components which are constructed in a way that they solely bundle services of a certain category. Quasar Enterprise postulates that interfaces of a specific category are only callable by components of the same category-level or by components of higher categories, with ”interaction” being the highest and ”data” being the lowest category. So, a component categorized as function component, hosting only application services of the function category, can legally call function services or data services for instance.

The **current application landscape** (current AL) is also divided into application components which in practice usually means that these are tantamount to existing physical applications. In contrast to the ideal AL, current ALs are usually not structured in the sense of business domains. Instead, they are either structured by organizational units (e.g. corresponding to departments) or are not structured at all. Due to this absence of a domain structure, we refrain from modeling domains in current ALs. Just like ideal components, current components bundle functionality organized as operations, which are again the atomic functional building blocks. Since service is a term used in the context of the ideal AL only, we will further focus on operations as least common comparable units when developing metrics to perform the gap analysis.

At a certain point in planning the evolution of the current AL, the architect needs to define intermediate states of the AL as steps of the migration road map. Being in between the current AL and the ideal AL, such an intermediate state is referred to as **target application landscape** (target AL).

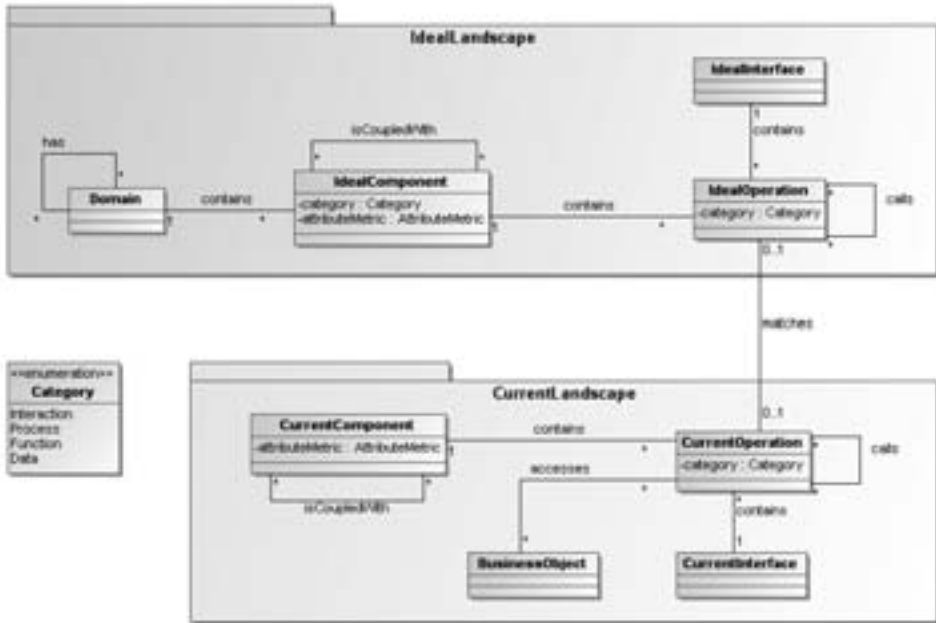


Figure 2: I-Pattern for Gap Analysis (according to [Sec09]).

The corresponding I-Pattern to perform gap analysis consists of the following entities (see also Figure 2):

### Ideal AL:

1. **Domain** - Domains are grouping high level business functionality of an ideal AL. They describe the top level work performed by an organization or an organizational unit and - when nested - also high level functionality. Typically enterprises organizational structures are organized by domains. Examples for domains include: Customer Relationship Management, Partner Relationship Management, Sales, Marketing, Product Management. The set of all domains builds the ideal application landscape.
2. **Ideal Component** - Ideal components belong to a specific domain and are hosting the atomic building blocks of an AL - operations. Hence, ideal components do also group business functionality but on a more fine grained basis than domains. Components are often considered as logical applications which reflects their granularity quite well. Ideal components only bundle operations of a specific category (*interaction, process, function, data*) and are classified corresponding to the category of

their contained operations.

3. **Ideal Interface** - Ideal interfaces describe services offered by ideal components on a logical level. Synonymously to web-services on the physical level, interfaces offer a public description of one or multiple operations.
4. **Ideal Operation** - An ideal operation is the atomic entity of an ideal AL. It is the fine grained logical description of business functionality on a method or function level (CRUD operations for instance). An operation is classified according to a specific category (*interaction, process, function, data*).

#### Current AL:

1. **Current Component** - Current components describe the business functionality of the current AL. In practice, this means they describe the functionality provided by existing applications. In contrast to the ideal AL, current components are not grouped by domains, since domain is a term to express the future (ideal) high level functionality of an enterprise.
2. **Current Interface** - Current interfaces describe services offered by current components on a logical level.
3. **Current Operation** - Like ideal operations, a current operation is the atomic entity of the current AL and is also the fine grained logical description of business functionality on a method or function level. A current operation is classified according to a specific category (*interaction, process, function, data*).
4. **Business Object** - Business objects represent the entities relevant in the business domain. Examples for business objects include: customer, contract, order, invoice. Only current operations of category *data* should have writable access to business objects - ideally exactly one operation per business object (see metric *Data Sovereignty* for details).

## 4 Metrics

We identified metrics to measure the distance between the current application landscape and the ideal landscape. We subdivide metrics into quantitative and qualitative metrics. Quantitative metrics consider structural differences and lead to concrete recommendations for restructuring the current application landscape towards the ideal landscape. Qualitative metrics complement quantitative metrics and highlight critical components of the current application landscape. In a nutshell, quantitative metrics show what needs to be restructured, whereas qualitative metrics focus on what needs to be given special attention to by the architect when the restructuring takes place. Qualitative metrics are important but they are predominantly expressed by attributes (see [PSS09] for more details) and are not directly connected to the I-Pattern, so we discard them here. The following quantitative metrics are based on [EHH<sup>+</sup>08] and directly applicable to the I-Pattern.

- *Completeness* means, that the functionality provided by the current application landscape is measured and compared to the functionality provided by the ideal application landscape. Such comparison is performed regarding operations of both, current and ideal landscape. Thus, the ratio is used as a key indicator for completeness calculated as follows:

$$\frac{\#OP_{tALnotImpl}}{\#OP_{tAL}},$$

in which  $\#OP_{tALnotImpl}$  describes the number of ideal landscape operations not implemented in the current landscape, and  $\#OP_{tAL}$  describes the overall number of ideal AL operations

- *Purity of the Domain* enunciates the degree of unambiguousness of the mapping of current AL components to ideal AL domains. The performance indicator is:

$$\frac{\#cComp_{gtOneDom}}{\#cComp},$$

in which  $\#cComp_{gtOneDom}$  describes the number of current AL components hosting AL operations mapped to more than one ideal domain. The overall number of current AL components is summarized by  $\#cComp$ .

- *Purity of Categories* describes the degree of unambiguousness of the mapping of current AL operations to a certain service category. The corresponding performance indicator is:

$$\frac{\#cComp_{OPdiffCat}}{\#cComp},$$

in which the number of current AL components hosting AL operations of different categories is described by  $\#cComp_{OPdiffCat}$  and the overall number of current AL components is described by  $\#cComp$  respectively.

- *Data Sovereignty* measures the degree of unambiguousness of the mapping of business objects to current AL components having writable access to their data representation. The performance indicator is:

$$\frac{\#BO_{nw} + \#BO_w}{\#cComp},$$

in which the number of business objects being writable by non data components is described by  $\#BO_{nw}$ . The number of business objects being writable by more than one component is summarized by  $\#BO_w$ .  $\#cComp$  represents the overall number of current AL components.

- *Correct Category Dependencies* measures the relative amount of illegal service calls between components of different categories according to the definition of callable as stated in [PSS09]. A metric is indicated by:

$$\frac{\#ISC_{cComp}}{\#SC_{cComp}},$$



in which the number of illegal service calls between current AL components is represented by  $\#ISC_{cComp}$ . The overall number of service calls between current AL components is described by  $\#SC_{cComp}$ .

Quantitative metrics can be used to quantify violations described by violation patterns of structural deviation between current and ideal AL. Each violation pattern implies measures to solve the deviation. To give an example, we refer to Figure 3.

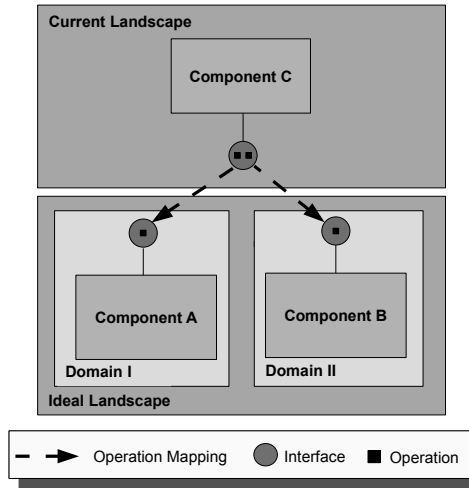


Figure 3: Purity of the Domain Violation.

It shows the violation of domain-purity quantified by the the metric *Purity of the Domain*. As part of the design phase of the ideal AL, the functional decomposition of operations from Component C of the current AL leads to a situation, where existing service functionality should be split in order to separate functional concerns for instance. In the planning phase of the ideal AL, both operations of Component C are assigned to new components of different domains. The I-Pattern allows a mapping of operations between two states of the application landscape. So this structural deviation between current and ideal AL can automatically be detected by algorithms on data having the structure of the I-Pattern, when gap analysis is performed. The metric *Purity of the Domain* is able to quantify the domain purity violations and can be used as indicator for this aspect of structural deviation. The suggested measure to overcome the deviation in this case would lead to the suggestion to initiate a project to split Component C.

## 5 Example

To clarify the abstract I-Pattern of gap analysis, we would like to give brief inside into the case study this pattern stems from. We would like to introduce our prototype for

gap analysis and the underlying data structure. For more information about the entire methodology and the actual case study we refer to [PGS09].

The case study is taken from the utility domain, where the current AL needed to be compared to an ideal AL. To manage such complex task, a prototype was developed to support the architect. This prototype is able to read a formalized representation of both, ideal AL and corresponding current AL (see Figure 4) to allow a manual semantic mapping of operations and to perform a gap analysis on these data structures.



Figure 4: Prototype to support gap analysis.

After importing the current and the ideal AL, the enterprise architect can see a graphical representation of the ALs inside the tool, and he is able to start the gap analysis by weighting the quantitative metrics according to enterprise specific preferences (Figure 4 shows the visualization of both ALs inside our prototype). These preferences can be stored by the tool and hence be reused in future analysis. Based on these metric weights, the tool calculates the structural deviation between current and ideal AL and highlights gaps inside the graphical representation of the landscapes if required. The prototype has the ability to calculate measures to resolve the structural deviation. Such measures are summarized as an action list inside the tool and a detailed description is available. Each measure is bound to a certain deviation observed by a specific metric and the position in the action list is determined by the weight of the metrics which assures an adequate prioritization of measures. Considering the metric *purity of the domain* a suggested action might be to split or to merge components of the current landscape to achieve a higher *purity of the domain*.

We use XML to store the elements of our ALs. The left side of figure 5 depicts a representation of the ideal AL whereas the current AL is described by the XML-file on the right side of the figure. The partitioning of ALs as introduced in section 3 is considered by the XML-file's structure. Component-elements can be used as child-elements of domain-elements for instance.

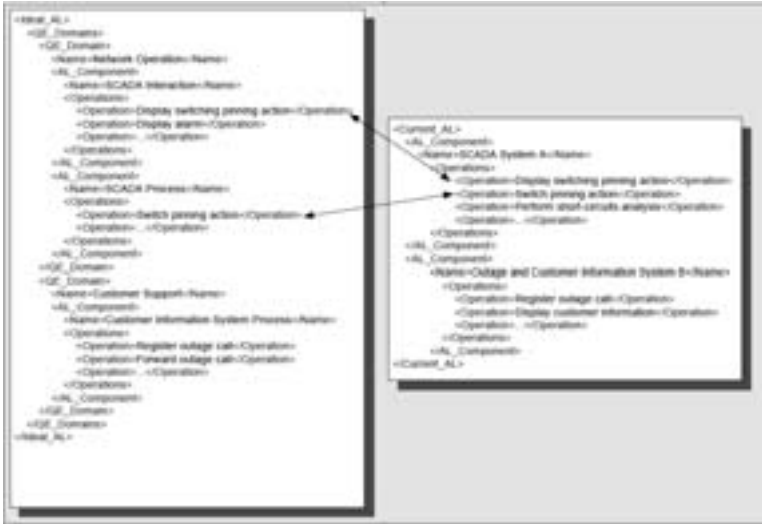


Figure 5: Rough data structure of ALs as used for structural mapping and gap analysis inside the prototype.

All quantitative metrics are calculating indicators for structural deviation between two data structures. Figure 6 illustrates the metric *Completeness*. The ideal AL-operations not implemented in the current AL are highlighted in red (*Display alarm, Register outage call*). The number of such current AL-operations divided by the number of the overall Ideal operations lead to the indicator of completeness.

According to [PSS09] all quantitative metrics form the overall distance between current and ideal AL. The tool is able to calculate the overall distance value and can record distance values over time for long term analysis (for example to indicate progress when the current AL evolves and is compared to the ideal AL again).

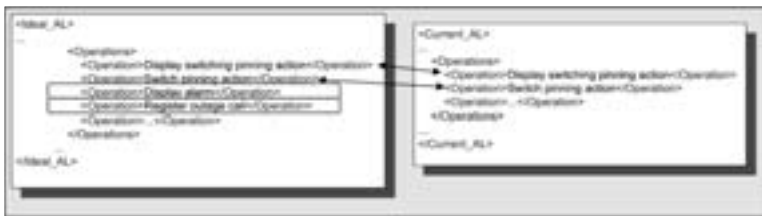


Figure 6: Violation of the completeness metric.

## 6 Consequences

We would like to consider some consequences of applying the gap analysis pattern on the basis of the forces described in section 2.1.

### 1. **Qualitative assessment and quantitative assessment**

A quantitative assessment of the differences between the as-is and ideal state produces manageable key figures and directly leads to a first list of possible needs for actions to be taken in order to converge towards the ideal state. Such assessment should be complemented by a qualitative assessment comprising more general IT strategic considerations, for example with regard to strengths, weaknesses, chances or risks of the current landscape. The results of the quantitative and qualitative assessment are weighed against the short-term operative goals to finally come to a workable list of actions to be taken. When performing the gap analysis, all these considerations need to be taken into account by an architect as part of an overall EA development process.

### 2. **Considering entire application landscape or focus on selective components**

The amount of data necessary to perform gap analysis on the entire application landscape on a detailed level considering operations is overwhelming. Unless data is automatically collectible - which has never been the case for pre-SOA landscapes in our daily work observations- it is not realistic to model the entire application landscape. However, this pattern is beneficial for partial consideration of the application landscape, when single systems need to be restructured, for instance.

### 3. **Regarding state snapshots for gap analysis or monitor the distance over time**

Gap analysis is a data intensive analysis and the manual detection of structural deviations is a time consuming task. Whenever gap analysis is a reoccurring task for an enterprise architect, tool support is advisable. This is especially true, when progress of architectural work should be measured continually. In this case, single comparison of two states is insufficient.

## 7 Conclusion and Outlook

In our contribution, we introduced the task of gap analysis and elaborated on an I-Pattern for gap analysis on application landscapes. We described how a gap analysis between a current non SOA AL and service oriented ideal AL can be performed based on a formal I-Pattern and supported by a software prototype in order to derive a suitable target application landscape.

In this contribution we focused on the shift from non SOA toward SOA. However, a growing number of enterprises is going to introduce or has already established SOAs. Gap analysis is also of relevance at evolution on established SOAs. Ongoing research is aiming in this direction and we are currently working on evolution patterns for SOA.

## References

- [AGA<sup>+</sup>08] Ali Arsanjani, Shuvanker Ghosh, Abdul Allam, Tina Abdollah, Sella Ganapathy, and Kerrie Holley. SOMA: A method for developing service-oriented solutions. *IBM Systems Journal*, 47(3):377–396, 2008.
- [BELM08] Sabine Buckl, Alexander M. Ernst, Josef Lankes, and Florian Matthes. Enterprise Architecture Management Pattern Catalog. Technical Report TB 0801, Software Engineering for Business Information Systems (sebis), Technische Universität, February 2008.
- [EHH<sup>+</sup>08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, and Johannes Willkomm. *Quasar enterprise: Anwendungslandschaften serviceorientiert gestalten*. dpunkt.verlag GmbH, Heidelberg, 1. edition, 2008.
- [Fow96] Martin Fowler. *Analysis Patterns Reusable Object Models*. Pearson Education (US), 1996.
- [Gam94] Erich Gamma. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley, Reading Mass. [u.a.], 2. print.. edition, 1994.
- [KBS06] Dirk Krafzig, Karl Banke, and Dirk Slama. *Enterprise SOA: Service-oriented architecture best practices*. The Coad series. Prentice-Hall, Upper Saddle River, NJ, 6. print edition, 2006.
- [PGS09] Matthias Postina, José Gonzalez, and Igor Sechyn. On the Architecture Development of Utility Enterprises with Special Respect to the Gap Analysis of Application Landscapes. In Ulrike Steffens, Jan Stefan Addicks, Matthias Postina, and Niels Streekmann, editors, *Proceedings of the 3rd Workshop "MDD, SOA, and IT Management"*, Oldenburg, Germany, September 2009. GITO.
- [PSS09] Matthias Postina, Igor Sechyn, and Ulrike Steffens. Gap analysis of application landscapes. In Vladimir Tomic, editor, *Enterprise Distributed Object Computing Conference*, pages 274–281. IEEEExplore Digital Library, 2009.
- [Sec09] Igor Sechyn. *Komponentenentwurf für Anwendungslandschaften am Beispiel eines Energiekonzerns. Ein werkzeuggestützter Abgleich zwischen Ist-Landschaft und Ideal*. Diplomarbeit, Carl von Ossietzky Universität, Oldenburg, 2009.
- [The09] The Open Group. *TOGAF Version 9*. van Haren Publishing, 2009.

# Business Driven SOA-Service Candidate Identification

Martin Weber, Markus Krieger  
SIS D PRO C SA1  
SIS D MC TS  
weber.m@siemens.com  
markus.krieger@siemens.com  
2010/03/10

**Abstract:** Service Oriented Architecture (SOA) is nothing new. Using SOA in the company's IT-Landscape gives the ability to drive IT-Evolution through business and to enable better business operations through a more flexible, agile and low maintenance IT. Hence it's amazing that an SOA landscape is quite rare in globally operating companies. IT-Landscapes of global companies are often historically grown and therefore complex, not agile and need a lot of maintenance. So the question is how a company can adapt an SOA within their existent IT-Landscape realizing good benefit in an effective way. The approach described in this Methodology-Pattern deals exactly with that question and helps planning an SOA-Transformation with good benefit, which is consistently aligned to the business. Therefore, the newly developed SOA-Landscape has to be business driven. Hence SOA-Services will only be implemented where SOA-Service makes sense and SOA-Services are designed in a business-need-oriented way to guarantee good business support.

## 1 Introduction

The method BUSINESS DRIVEN SOA-SERVICE CANDIDATE IDENTIFICATION (BUSINESS DRIVEN SOA SCI) deals with an effective usage of Service Oriented Architecture (SOA) for architecture transformation. This approach addresses the strategic planning of an SOA-Transformation, which is consistently aligned to the business targets and needs. SOA-Transformation in this context means the initial transformation of an application oriented architecture to SOA. Additionally, some parts of this method can also be used for the evaluation of an already implemented SOA. One result document of this method comprises an overview about the most relevant business areas of the company with regard to the business strategy. Furthermore, these relevant business areas are evaluated concerning their SOA-Potential. Therefore, using the results of this method may help Chief Information Officers (CIO) to set a new strategic direction concerning the IT-Architecture and will ensure a transparent IT-Value-Add. Additionally, IT-Business-Alignment will be achieved, because all tasks mentioned in this method are performed with close contact to the business. Hence SOA-Services will only be implemented in

areas, which have a positive strategy impact and gain advantage to the business. Beyond this, the information of the analysis can be used as base for the future-oriented SOA-Roadmap.

## **2 Business Driven SOA-Service Candidate Identification**

The principal attraction of SOA is its ability to drive IT-Evolution through business and to enable better business operations through a more flexible, agile and low maintenance IT [De05]. An approach which helps to transform an existent IT-Landscape to an SOA-Landscape with strict regard to the business strategy and business demands is documented in the method BUSINESS DRIVEN SOA SCI. The focus of this method is the transparent business-driven identification of SOA-Services without regard to the manner of their implementation. Thus the result of this method is an SOA-Service candidate portfolio. In consequence the implementation of these identified SOA-Services will lead to a newly business-driven SOA-Landscape. Due to the consistent business-orientation this SOA-Landscape ensures good business support. Remark the implementation process of SOA-Services is not described in this method.

### **2.1 Example**

Global enterprises have big IT-Landscapes, which are characterized by a lot of complex and heterogeneous systems. Normally, such an IT-Landscape is not agile or flexible, because it is often difficult and extensive to adapt complex and heterogeneous systems to new business requirements. But characteristics like agility and flexibility are requested by the business. Implementing an SOA within your IT-Landscape will help to fulfil these requested demands [He07, p.124]. In consequence some parts of an enterprise IT are already captured by SOA-Services following the process, the company's IT-department uses to identify a new requirement and starts designing and implementing the new service. Additional new requirements are announced by the business and the IT generates new SOA-Services. Thus a lot of SOA-Services exist unmanaged and are not evaluated concerning their benefits and business support. Additionally, the business value added by SOA-Services is not transparent.

### **2.2 Context**

A large enterprise is facing the challenge of an SOA-Transformation, which is driven by bottom-up initiatives and not necessarily linked to the overall business targets.

## 2.3 Problem

You feel the risk of an unmanaged SOA-Landscape without being aligned consistently to the business. In consequence your SOA-Landscape is driven bottom-up by a few experts, who know the need for action. This landscape is composed of a large number of SOA-Services, however, it is not assured that all services of the enterprise IT are identified in the right business area and implemented in the right way. Identification in the right business area implies that SOA-Services should be implemented within areas where SOA will achieve good benefit and implementation in the right way denotes that SOA-Services should be exactly designed according to business needs. Concluding, a top-down enterprise view is missing and therefore probably strategic goal relevant topics are not covered by SOA-Services.

### **How can you assure business-oriented SOA-Service identification and implementation in the right business areas within a global SOA-Transformation?**

The following forces influence this situation:

- **Top-Down vs. Bottom-Up:** On the one hand a top-down approach assures that SOA-Services will only be implemented in business areas, which have significant impact on the strategic goals. In consequence possibly lower important topics (which may have high SOA-Potential) won't be cared about. On the other hand using a bottom-up approach will lead to SOA arrangements in areas which possibly don't have significant impact to strategic goals.
- **Business-driven vs. IT-driven: Who is the driver of an SOA-Transformation?**
- **SOA vs. non-SOA:** SOA exhibits positive characteristics like e.g. flexibility, reusability, loose coupling and agility. Nevertheless it has to be checked if some of the announced positive SOA characteristics are needed by the business functionality which is discovered in the various business areas.
- **Established roles vs. new roles:** Are the established roles sufficient for a business-driven SOA-Transformation? Do you need additional roles to ensure a business-oriented SOA-Transformation?



## 2.4 Solution

Using a top-down approach in an SOA-Transformation with regard to effectiveness is necessary for realizing significant benefit for the company. Hence the SOA-Transformation process has to assure that SOA-Services will be identified in the right business area (with regard to business strategy and SOA-Potential) using a top-down approach and will be implemented appropriately to the business needs (business-oriented). Consequently, these SOA-Services, which are in detail specified by people working in the relevant business unit, guarantee good business support for this area [cf. Of08, p. 463ff.], [cf. De05].

A global business-oriented SOA-Transformation can be structured in the following four parts. First part **Strategic Business Alignment** discovers top-down business areas and highlights those, which have strong impact on the strategic goals. This discovery is followed by an SOA-Potential evaluation of selective business areas which have high strategy impact. The second part **Operationalization** discovers functional and non-functional business requirements in the selected business area and gives an implementation recommendation concerning its target architecture (SOA/non-SOA). In the **Implementation** part a person with good business understanding, good IT knowledge, especially in the area of SOA and modelling experience (in the following this role is called Business Integration Engineer) specifies in detail these requirements and implements the SOA-Service. This part is only mentioned for completeness but will not be discussed in detail, because the intention of this method is only the business-oriented identification of SOA-Services free of their implementation modality. Part four **Governance** influences all above mentioned parts and deals with the planning and controlling of the SOA-Transformation. The results of this method support the planning part in governance. Therefore these results should be used as input while planning the SOA-Transformation. Figure 1 illustrates the interrelationship of these four parts within an SOA-Transformation.

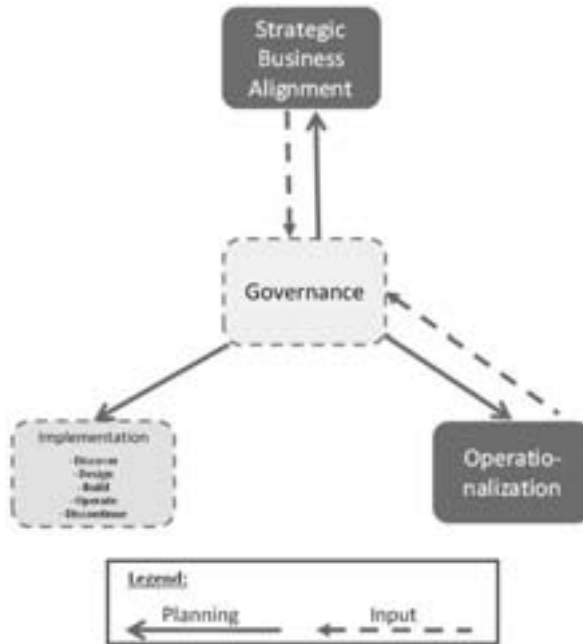


Figure 1: Interrelationship of the four parts within an SOA-Transformation

The execution of all parts is planned by the IT Strategy department which is owner of the governance part. Strategy changes influence the part *Strategic Business Alignment*. Therefore the IT Strategy should link the planned execution of this part to the time, where a business strategy update will be released. Typically the periodicity of business strategy updates is annually. For light business strategy changes, the IT Strategy should plan a light execution of part *Strategic Business Alignment*. This means that the result document of this part (generated in the initial complete execution) has to be checked against changes and has to be updated selectively in cooperation with the business strategy responsible. The execution of this part is also a trigger for (re-)planning activities of part *Operationalization* and *Implementation*.

In the following, the main parts of Business-Driven SOA SCI, *Strategic Business Alignment* and *Operationalization* will further be detailed.

### 2.4.1 Strategic Business Alignment

The base for the method Business Driven SOA SCI is the *Functional Domain Model* (Figure 2). This model provides a common entry point to IT-Architecture for business and IT. It structures areas of activity of IT-Architecture using the language of the business and consists of business domains like ‘Supply Chain Plan’, ‘Procurement’, ‘Monitoring and Controlling’ etc.

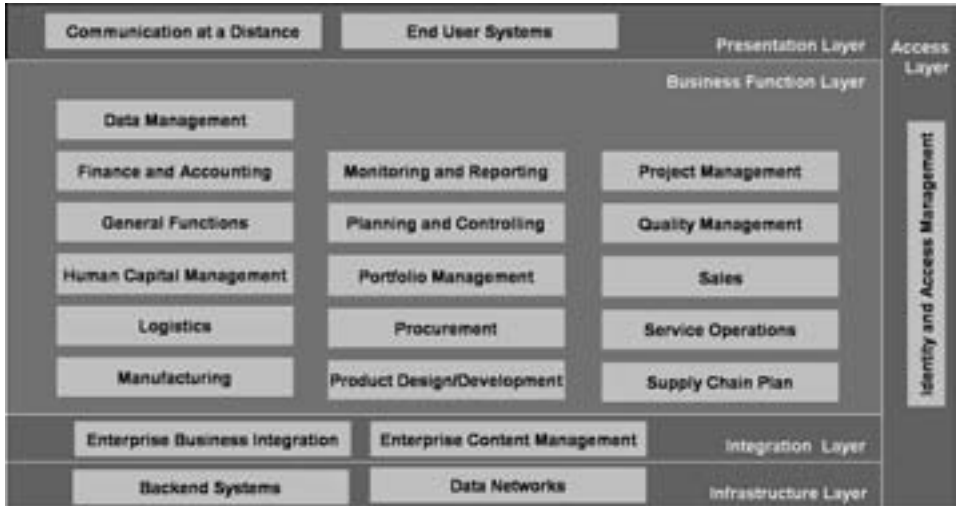


Figure 2: Siemens Functional Domain Model (Source Siemens AG : CIT)

By using this Functional Domain Structure you can enable IT-Business-Alignment. Every business function can be allocated exactly to one Functional Domain. A business function is a specific functionality, which is needed by the business and is exactly comprised within a Functional Domain (e.g. Personnel Administration inside the Functional Domain Human Capital Management). In addition to the *Functional Domain Model* the business strategy is required. *Strategic Business Alignment* correlates each Functional Domain of the Functional Domain Model with the business strategy. This part is divided into two steps: Domain ranking and SOA-Potential ranking (Figure 3).



Figure 3: Steps of part Strategic Business Alignment

Step one ranks every Functional Domain concerning its importance to reach the strategic goal. During this ranking activity, only business functionality covered by this Functional Domain is important. IT aspects and systems which support this area are not discussed. To support traceability every importance value holds a statement which describes why this value was determined by the business strategy people with regard to the corresponding strategic target.

A session with members of the business strategy department is used for ranking the Functional Domains. As result of part one you will get a ranking between all functional areas of the Functional Domain Model (Figure 4).

Strategy	Data Management	Finance and Accounting	General Functions	...
<b>Business Unit 1 Strategy</b>				
Strategic target 1	●	●	○	○
Strategic target 2	●	●	○	○
Strategic target 3	○	●	●	○
Strategic target 4	●	○	○	○
<b>Business Unit 2 Strategy</b>				
Strategic target 1	○	○	○	○
Strategic target 2	○	○	○	○
Strategic target 3	○	○	○	○
Strategic target 4	○	○	○	○

Figure 4: Excerpt of result document – Step1: Domain ranking

Every functional domain is ranked concerning its importance for reaching the strategic target. Every ranking will be performed using the question: “Which value proposition have the activities/functionalities covered by this Functional Domain for reaching the strategic goal?”. Consequently, the most important areas for the company are identified. E.g. in figure 4 the important Functional Domains are “Data Management” and “Finance and Accounting”. All Functional Domains with significant impact on the strategy targets will be used as input for the second step of *Strategic Business Alignment*.

Step two evaluates the SOA-Potential of these important Functional Domains. Aspects concerning the benefit and effort are discussed to identify areas with maximum SOA-Benefit. The evaluation will be performed by hypotheses. These hypotheses give a hint to the SOA-Benefit and deal with topics about process-efficiency, flexibility and IT-efficiency. Topics like project costs and IT-running-costs will be used for evaluating the SOA-Effort. Figure 5 shows an excerpt of the hypothesis document.

		Which hypothesis is most applicable?				
		A	B	C	D	E
Process-efficiency	<b>Number of (avoidable) media disruptions</b>					
	Media disruptions are quite seldom and are easy to address					
	Media disruptions are quite frequent but are hard to address					
	Media disruptions are quite seldom but can be mostly automated					
	Media disruptions are quite frequent and can be mostly automated					
	<b>Execution frequency and level of automation</b>					
	The level of automation concerning the process steps is high and can not be improved any more					
	The level of automation concerning the process steps is low but can not be improved					
	The level of automation concerning the process steps is high but improvement potential still exists					
	The level of automation concerning the process steps is low and can be highly improved					
<b>Number of functional redundancy (in terms of productivity)</b>						
Redundant functionality does not exist and the user productivity is already high						

Figure 5: Excerpt of the hypotheses document

Business people (Functional Domain Responsible, Enterprise Domain Architect) and Enterprise IT architects, have to choose together the most applicable hypotheses.

As result of the step *Strategic Business Alignment* all strategically important Functional Domains are evaluated concerning their SOA-Potential and will be categorized as *SOA sphere of activity*. All SOA spheres of activity are positioned in an SOA-Portfolio matrix with the two dimensions SOA-Potential and SOA-Effort. Concluding figure 6 summarizes the interaction of the two steps Domain ranking and SOA-Potential ranking.

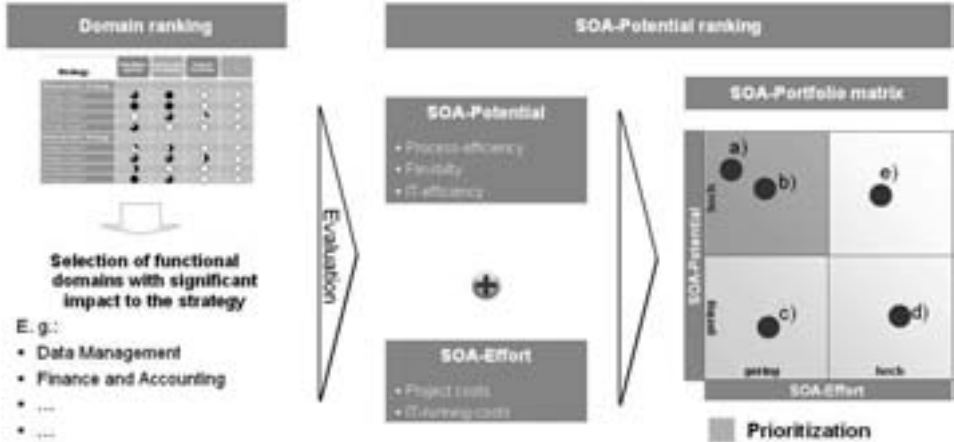


Figure 6: Interaction of step Domain ranking and step SOA-Potential ranking

In consequence, the information comprised in this SOA-Portfolio matrix can be used by CIO, IT Architects, IT strategy or other IT members as input for defining an SOA-Roadmap. This roadmap prioritizes the exploration of the SOA spheres of activity. Normally, such a roadmap is formulated by the IT strategy in cooperation with IT Architects. The exploration task is covered by the second step *Operationalization*.

**2.4.2 Operationalization**

Every area which is defined in the SOA-Roadmap will be explored. That exploration duty comprises three steps which are worked through consecutively (Figure 7).



Figure 7: Three steps of part Operationalization

All steps will be performed with members of the business department, who are responsible for the chosen business area and with members of the IT-Department. In the first step, all *business function trees* of that functional area have to be discovered using e.g. an existent process landscape. This process landscape comprises various functionalities, which are linked through workflow logic. A *business function tree* is a tree, which consists of one extensive main function (e.g. travel management) and its sub-functions (e.g. car booking, flight booking, hotel booking, train booking ...) [cf. Mü06, p. 130].

Furthermore, functional and non-functional business requirements (raised by the business people) are collected and checked against the functions listed in the trees to identify the target architecture with modified old or discovered new functionality. *Function Cards* (excerpt shown in figure 8) help to document all raised information.

Siemens IT Solution and Services		Functionscard		Date
<b>Block 1: General Information</b>				
	<b>Name</b>			
	<b>Parent Function</b>			
<b>Block 2: Requirements</b>				
<b>No.</b>	<b>non-functional</b>			
I				
II				
III				
IV				
V				
VI				
<b>No.</b>	<b>functional</b>			
I				
II				
III				
IV				
V				
VI				
<b>Result of the IT-Decision Template</b>				
<b>Implementation manner</b>		<b>SOA</b>	<b>non-SOA</b>	<b>Implementation manner is SOA. Next block must be filled out.</b>
<b>Services</b>				
<b>Block 3: SOA Business Process Service (BPS) Candidate Document</b>				
<b>No.</b>	<b>Attribute</b>	<b>Description</b>		
1	Trigger			
2	Function	<b>No.</b>	<b>Short function description</b>	
		I		
		II		
		III		
3	Contract (business)			

Figure 8: Function Card excerpt

This *Function Card* is a document, which is structured in three blocks. The first block provides general information about the business function, the second block lists requirements for the business function and the third block comprises the *SOA-Service Candidate Document*, which includes first specification details of the identified SOA-Service. Afterwards, all business functions, which need modifications or which are newly discovered will be evaluated concerning its implementation (SOA vs. non-SOA). This evaluation process is guided by an *IT-Decision template* (excerpt is shown in figure 9), which contains expressions concerning topics like individuality, external connections, transactional functions, agility and complexity.



Criteria & Expressions (Which expression is most applicable?)	
<b>Individuality</b> The individuality of a Business Function gives information about the frequency and consequence of functional/technical differing requirements	
<b>Expression 1:</b> Functional/technical different requirements are frequent and produce many (often redundant) variants of that Business Function →High functional/technical individuality	
<b>Expression 2:</b> Functional/technical different requirements are seldom respective can be planned and do not produce variants of that Business Function →Low functional/technical individuality	
<b>SOA-Characteristic:</b> An SOA has the ability to react flexible concerning new functional/technical requirements due to individual coupling and reuse of SOA-Services.	
<b>External Connections</b>	

Figure 9: IT-Decision Template excerpt

Evaluating every requirement using the previous named IT decision topics will assure, that only SOA-Services will be designed, if SOA is suggestive. Therefore, good SOA-Benefit is realized. Requirements which are not advisable to be implemented as SOA-Service are not rejected. These requirements are also business needs and have to be handled. Hence these needs, which are already collected on the *Function Card*, will be handed over to e.g. a Demand Manager. All other pro-SOA evaluated requirements are picked and associated with newly discovered SOA-Services. Every new SOA-Service will be described more detailed in the third block (*SOA Business Process Service Candidate Document*) of the *Function Card*. This document will be further used by the *Business Integration Engineer*, who will create, with the help of business people, a more detailed specification of the discovered SOA-Service and is responsible for the SOA-Service implementation. Hence such a role must be established within the organisation.

The acceptance of this identified SOA-Service by a Business Integration Engineer changes the SOA-Service state from “Candidate” to “Discovered” and consequently initiate the SOA-Service implementation process (see Figure 1).

## 2.5 Implementation

In order to use this method, it is very important that the organisation has the willingness to plan an SOA-Transformation using the parts mentioned in BUSINESS DRIVEN SOA SCI (see Figure 1). Therefore, the organisation on the one hand must be qualified to accomplish every part and on the other hand must use the results of each part correctly. Additional required governance structures have to be created and adequate people have to be involved in every part.



Consequently the following roles are requested by this approach: *Business Strategy Expert, IT Strategy Expert, Enterprise Functional Domain Architect, Enterprise IT Architect, Functional Domain Responsible, Functional Domain IT Architect, Business Integration Engineer*. Additionally, this enumeration can be expanded with other people of your company, if you think their knowledge is also needed.

Step one in part *Strategic Business Alignment* requires the participation of people owning the role *Business Strategy Expert* and *Enterprise Functional Domain Architect*. Participants in step two of this part are *Enterprise Functional Domain Architects, Enterprise IT Architects and the Functional Domain Responsible*. People owning the role *Enterprise Functional Domain Architect* are characterized to have a global business view about the Functional Domains; the *Enterprise IT Architects* are characterized to have a global view about the IT supporting the Functional Domains; the *Functional Domain Responsible* has an excellent knowledge about the business activities (business functions) which are comprised in a specific Functional Domain.

The next step *Operationalization* is performed by people owning the role *Functional Domain Responsible, Business Integration Engineer, Functional Domain IT Architect*. To identify, whether the specific functional or non-functional requirement should be implemented traditionally or with SOA-Technology input of the *Functional Domain IT Architect* is needed. Accountable for this step is the *Functional Domain Responsible*. The *IT Strategy Expert* moderates both parts *Strategic Business Alignment* and *Operationalization*. Furthermore the *IT Strategy Expert* is accountable for part *Strategic Business Alignment*. Additionally the IT Strategy department is responsible for the execution of *Strategic Business Alignment* and will be informed about the results of part *Operationalization* (SOA Service Candidates) to plan the implementation of the SOA-Service Candidates / Business Process Service Candidates [Ma09].

Further Top-Management support of this approach is needed and an escalation board has to be defined, which will solve problems concerning offense against the method or defined SOA-Roadmap. Figure 10 summarizes the above mentioned interaction between the roles using a RACI-Model.

Role \ Part	Strategic Business Alignment		Operationalization
	Step 1	Step 2	
Business Strategy Expert	C	I	I
IT Strategy Expert	R A	R A	R
Enterprise Functional Domain Architect	C	C	A
Enterprise IT Architect	I	C	I
Functional Domain Responsible	I	C	C
Functional Domain IT Architect	I	I	C
Business Integration Engineer		I	C

C = Consulted, R = Responsible, A = Accountable, C = Consulted, I = Informed

Figure 10: RACI-Model of method BUSINESS DRIVEN SOA SCI

## 2.6 Know Uses

The approach documented in method BUSINESS DRIVEN SOA SCI is in use in the following company:

- Siemens AG - Siemens IT Solutions and Services

## 2.7 Consequences

This approach deals with a lot of different people of business and IT. So it has to be considered that all people have to take activities requested by this method additionally to their main line activities. Hence it is very important that everyone, who is needed for this approach is well-informed about the benefits of using this approach and ideally the benefit for himself. In conclusion the availability of needed business and IT experts has to be checked and their attendance in the sessions has to be assured.

**Top-Down vs. Bottom-Up:** Traditionally a lot of IT activities are triggered by bottom-up initiatives, because IT Architects have great knowledge about their IT-Landscape and therefore seem to know where architectural changes (e.g. SOA) are needed. As already mentioned for realizing a good benefit for the company top-down initiatives are necessary. In conclusion a combination of both initiatives is needed. Consequently, it is suggestive to use the top-down approach and connect it to the bottom-up initiatives, which are already running. Thereby, transparency between bottom-up and top-down activities is established. For this reason important business areas for the company, which are not covered by bottom-up initiatives, are identified. Vice versa bottom-up worked areas, which do not have significant business impact, are also identified. To ensure good business benefit within an SOA-Transformation it is advisable to use a top-down approach. Thereby you get the ability to deploy your limited resources at the best concerning the business benefit and the waste of money in strategically insignificant areas is avoided.

**Business-driven vs. IT-driven:** An architectural SOA-Transformation only guided IT-driven is not accepted by the business. Thus business needs are not supported ideally and so business people are not willing to use the newly offered SOA-Services. In addition, an important target of a future-oriented IT-Architecture, IT-Business-Alignment, is failed. Therefore the driver of an SOA-Transformation should be the business and IT should consult the business with IT's technical expertise. An additional success factor for a business-oriented SOA-Transformation is that the SOA-Services specified by the business have to be exactly implemented according to the business specification to assure business support at the best. To assure this it is recommended to use documents which hold all information raised by the business. It is very important that the development of this SOA-Service document is performed by IT and business people, because both should have a mutual agreement concerning the content of this document. Furthermore this is the initial document for the SOA-Service, which is handed over to the IT. Due to the use of this document transparency between Service-Discovery by business people and Service-Implementation by people of the IT department is achieved. Accessorily this document ensures transparency between business and service view. In consequence traceability between SOA-Services and business needs is ensured.

Therefore, it is suggestive to plan such a new strategic IT direction with close contact to business people. On the one hand this yields to transparency between business strategy targets and the IT-Architecture and on the other hand business gets IT support at the best. Additional IT-Business-Alignment will be achieved.

**SOA vs. non-SOA:** Furthermore, SOA should not be an end in itself and should only be used in places, where usage of SOA is suggestive. Therefore *Functional Domain Architect*, *IT Architect* and *Business Integration Engineer* together have to check exactly, whether SOA-Technology is the right way to excellently support the business needs. If typical SOA characteristics are not requested by the business for the specific functionality it is suggestive to consider alternative implementation techniques than SOA.

**Established roles vs. new roles:** For a business-oriented SOA-Transformation special roles are needed. Roles requested by BUSINESS DRIVEN SOA SCI are often not established in large enterprises. Therefore it is advisable to check, whether all required roles are already established in your enterprise before starting with the approach described in this method.

## 2.8 See Also

In order to support the implementation of the method BUSINESS DRIVEN SOA SCI the following documents may help:

- Ranking template: This document is used for documenting the domain ranking and will lead to the portfolio matrix
- Function Card template: This document is used for documenting the functional and non-functional requirements
- IT-Decision template: This document comprises hypothesis which help to evaluate if SOA-Implementation is suggestive
- SOA-Service Candidate document template: This document comprises additional information, which is needed (in a first step) for implementing the SOA-Service

## 3 Acknowledgement and Outlook

This section includes acknowledgements to the people who supported the creation of this article and gives an outlook to the next steps in the development of BUSINESS DRIVEN SOA SCI.

### 3.1 Acknowledgements

I want to thank Markus Krieger and Christian Schweda, the reviewers of the initial version, all participants of the “2<sup>nd</sup> European Workshop on Patterns for Enterprise Architecture Management (PEAM2010)”, and especially my shepherd Alexander M. Ernst for the time they spent for reading, commenting and discussing this article.

Additionally, I want to thank my sponsor Dr. Andreas Intemann (Director SIS IT Strategy), who retained Markus Krieger and me to develop a method, which will help to plan the SIS-wide SOA-Transformation and finally allowed the documentation of this method in form of a pattern for the conference.

### 3.2 Next Steps in BUSINESS DRIVEN SOA SCI

The method behind this Methodology-Pattern (M-Pattern) was developed within SIS for planning the global SOA-Transformation with consistent alignment to the business. In conclusion SIS is able to manage the SOA-Transformation in an effective way realizing good benefit. The first step of part *Strategic Business Alignment* has already been successfully applied. The first business reactions were completely positive. Due to confidentiality the results cannot be presented in this article.

Furthermore, IT-Business-Alignment was achieved, because business people realized that the IT department tries to supply IT-Services (SOA-Services), which are defined and specified by the business and therefore will guarantee business process support at the best. Further steps of this method will be performed in the actual calendar year within the SIS.

In order to improve the readability of this pattern a future revision will split the pattern in a composite pattern [Di97] describing the overall method and in two patterns specifically targeting the two documented parts.

### Bibliography

- [De05] Deb, Manas et.al: Bringing SOA to Life: The Art and Science of Service Discovery and Design, <http://soa.sys-con.com/node/164560>
- [Di97] Riehle, Dirk "Bureaucracy": In Pattern languages of program design 3, Boston, MA, USA, 1997, pages 163-185, Addison-Wesley Longman Publishing Co., Inc.
- [HE07] Heutschi, Roger: Service Orientierte Architektur, Springer, Berlin, 2007
- [Ma09] Maier, Berthold et.al: Die SOA-Service-Kategorienmatrix: SOA aus dem wahren Leben - Teil 4, Java Magazin, Volume 09, Number 2, pages 76-84, 2009
- [Mü06] Müller, Sandra: Methodisches Erfinden im Personalmanagement, DUV, 2006
- [Of08] Offermann, Philipp: SOAM - Eine Methode zur Konzeption betrieblicher Software mit einer Serviceorientierten Architektur, Wirtschaftsinformatik Vol. 50, Number 6 / Dezember 2008, pages 461-471, Gabler Verlag

# Teaching Enterprise Architecture Management with Student Mini-Projects

Sabine Buckl, Alexander M. Ernst, Florian Matthes, Christian M. Schweda  
Chair for Informatics 19  
Technische Universität München  
eMail: {buckls, ernst, matthes, schweda}@in.tum.de

**Abstract:** Enterprise architecture (EA) management is one of the major challenges of modern enterprises. It aims at aligning business and IT in order to optimize their interaction. Experts in this context are scarce and only limited proven practice to educate students in EA management is yet available. This paper presents a pattern on how to teach EA management based on experiences and best practices gained in a lecture at the Technische Universität München.

## 1 Introduction and Overview

While an increasing demand for experts in enterprise architecture (EA) management in practice exists, approaches and practical experience in teaching EA management are scarce. On the one hand the experienced gap may be explained by the existing plurality of EA management approaches and on the other hand by the diversity of practical problem statements, which are specific for the involved enterprise, its context and culture. Whereas several consultancies offer workshops for interested practitioners, which last for several days, universities have not yet adapted their curriculum to this demand of practice.

The challenge of teaching EA management is to find the right balance between theoretical foundations and practical experience. The difficulties arising in this context do not only result from the complexity of the management subject – the EA – but from the involvement of various stakeholders<sup>1</sup>. These stakeholders with business or IT background have different and often conflicting interests in the system, which need to be appropriately accounted for during EA management. Similarly, 'political' circumstances in an enterprise may negatively affect the prospects of the EA management endeavor.

A first attempt to address the aforementioned challenges and equip graduates with theoretical and practical experience in EA management was performed by the computer science department of the Utrecht University [KBB08]. Koning et al. use a defined method with a given set of problems to be addressed, complemented by predefined viewpoints used to describe the EA of an industry partner. While the approach can be applied in many different contexts, due to its general procedure, it does nevertheless not well account for

---

<sup>1</sup>A *stakeholder* is an individual, team, or organization with interests in, or concerns relative to a system (cf. [Int07]).

the aforementioned typical EA management challenges and the specificity of EA management concerns in the participating enterprises. Similarly, the need to adapt the method to the enterprises context and culture is neglected. Hence, we propose a different teaching approach, which is based on EAM patterns suitable to address the specific concerns of each participating industry partner. Furthermore, the patterns the method builds on are adaptable to the context and culture of the industry partner under consideration.

This article includes patterns, which are part of or related to the *EAM Pattern Catalog*, a *pattern language for enterprise architecture management* [BEL<sup>+</sup>07, BELM08, BEL<sup>+</sup>08, Ern08, BEMS09, Ern10]. The *EAM Pattern Catalog* uses a pattern based approach to EA management. It is available online at <http://eampc-wiki.systemcartography.info/> [Cha09] and currently includes 164 EAM patterns<sup>2</sup>. The intention behind this article is to further extend the existing *EAM Pattern Catalog* by a M-Pattern documenting how to teach EA management.

## 1.1 Intended Audience

This article and the herein included patterns are intended for people concerned with teaching EA management. Thus, the article may not only be relevant for academics but may as a side effect also provide useful hints for in-house trainings on EA management in large companies. Put in other words, a large company could decide to let its new enterprise architects run through a mini-project likely EA management training.

## 1.2 EAM Pattern Map

This article includes the pattern TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT. Figure 1 shows TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT and its relationships to other patterns. MINI PROJECT EVALUATION is not included but may be helpful for evaluating student projects.

Related patterns about supervising thesis projects can be found in an article by Schümmer and Schmolitzky [SS08].

## 2 Teaching Enterprise Architecture Management

TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT describes how to plan and conduct a lecture disseminating knowledge on EA management. Thereby, the approach emphasizes on the importance of giving the students the possibility to not only listen to theoretical lectures but to also gain practical experience in a real-world setting.

---

<sup>2</sup>For a detailed explanation of the concept of EAM patterns refer to [Ern08, Ern10].

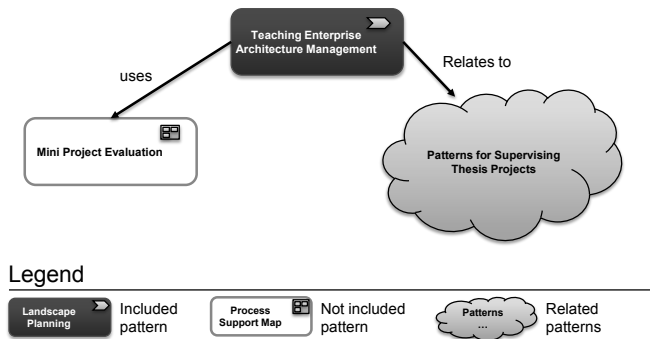


Abbildung 1: EAM Pattern Map

## 2.1 Example

The topic of EA management is an emerging one. Modern enterprises face the challenges of an ever changing environment they need to adapt to. At the same time, experts with practical experience in EA management are rare. In order to react to this increasing demand in practice the Technische Universität München has decided to include a lecture on the topic of EA management in her curriculum for the Wirtschaftsinformatik<sup>3</sup>-master. This lecture planned for 5 ECTS consisting of both a theoretical lecture and an accompanying lab course. The lecture should provide foundations on EA management and also include a practical part where students conduct so called mini projects in practical settings at an enterprise. In these mini projects the students work on problems, like *document visualizations and information models used for capturing the application landscape*, in real-world case studies at cooperating companies.

## 2.2 Context

An university wants to extend its Wirtschaftsinformatik-master curriculum with a lecture on the topic of EA management. The lecture should follow a twofold approach imparting theoretical knowledge in the area of EA management as well as providing the possibility for students to gain practical experience with real-world teaching cases in practical settings. The intended audience for the lecture are students in Wirtschaftsinformatik-master. The students can be expected to have an in-depth understanding of modeling with different languages, as e.g. UML or EPC, as well as an understanding for the economic environment, in which global acting companies have to compete nowadays.

<sup>3</sup>The term *Wirtschaftsinformatik* can be translated to business informatics or information systems.



## 2.3 Problem

You experience an increasing demand for EA management experts in industry. You as a university member want to equip students with the required theoretical background as well as practical experience to solve real world EA management problems.

**How do you organize a lecture on the topic EA management, which on the one hand provides students with a theoretical background on the subject and on the other hand imparts practical experience in solving EA management problems in real world settings?**

The following *forces* influence the solution:

- **Real-world teaching case versus artificial teaching case** What kind of teaching cases do you want to use? Real world teaching cases in cooperation with real companies or an artificial academic example? While real-world cases with a participating industry partner on the one hand enhance the motivation of the students, they on the other hand require a defined project goal limited to a certain part of the enterprise. In addition, confidentiality issues need to be discussed with the industry partner to allow anonymized publication of the project results.
- **One versus multiple teaching cases** Do you want to focus on one teaching case with better comparability or multiple teaching cases giving a broader introduction to EA management?
- **Maximal versus minimal termtime** What is a good duration for the accompanying practical project?
- **Size of student teams versus number of student teams** What is a good size for the student teams performing the teaching cases?

## 2.4 Solution

Solving the aforementioned problem, requires the involvement of different stakeholders, i.e. the university staff, who conducts the lecture, organizes, and evaluates the projects conducted at the industry partner, the practitioners from industry providing the problem to be solved, and students, who want to gain theoretical knowledge and practical experience in EA management. Figure 2 illustrates the single steps of the solution, a possible time frame of the single steps, and the main stakeholders involved.

### 2.4.1 Identify industry partners

As mentioned before, the objective to equip students with theoretical and practical knowledge in the field of EA management, can best be achieved if practitioners from industry are involved. This requires to contact potential industry partners during the planning phase,

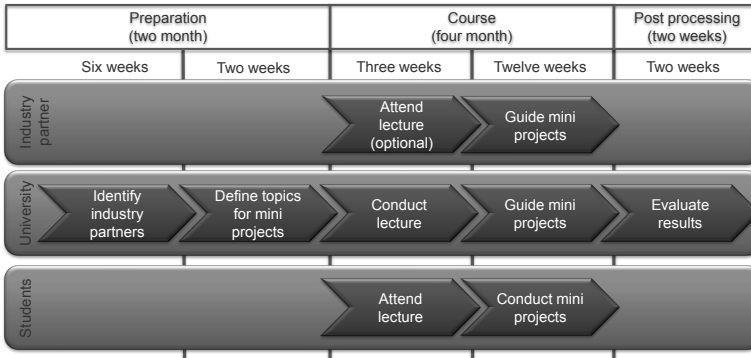


Abbildung 2: The solution process and the involved stakeholders

e.g. from previous research projects. Thereby, the parameters of the project to be conducted by the students, e.g. the tasks of the industry partner, the start, end date, and expected expenditure of time of the mini project as well as the benefits for the industry partner should be clearly outlined. Potential benefits for the industry partner are:

- Evaluating the EAM pattern approach at the industry partners enterprise.
- Early contact to graduates, who are interested in the topic.
- Meet other EA management practitioners with similar everyday problems.
- Possibility to attend lecture, e.g. for new employees.

To minimize travel times for the students, the industry partners should be located nearby. Nevertheless, the size of the company should be considered, as mainly global acting enterprises are concerned with the topic of EA management. Further, a considerable number of industry partners is required to keep the student mini project teams small in size, such that each student can be embedded well into the activities.

#### 2.4.2 Define topics for mini projects

Together with the practitioners from industry possible topics for the student projects, the so-called mini projects, have to be identified. Due to the limited time frame of the project, the problem statement has to be narrow regarding the students' activity e.g. documenting, analyzing, planning, etc. and regarding the scope and reach, e.g. the application and information layer or the business and organization layer. Possible examples for mini project topics are *cost reduction analysis for an application landscape in a corporate environment* or *analysis of a risk management system with its components and interfaces at a financial service provider*. The diversity of EA management problems reflects the maturity of the different enterprises in terms of the established EA management function and emphasizes on the enterprise specificity of the topic.

Each mini project consists of the following activities:

- Analysis of the existing EA management practices and status of the current EA
- Project planning
- Meetings with stakeholders
- Information collection
- Creation of EA deliverables (meta models, models, process descriptions, recommendations)

### 2.4.3 Conduct and attend lecture

In the lecture, the students and attending industry partners are given an overview about the state-of-the-art in EA management, including prominent frameworks like TOGAF [The09] and Zachmann [SZ92]. In addition, theoretical foundations like models, metamodels, architectures in general and IT governance are presented. Based on these foundations, the EA management function and their interplay with other enterprise-level management functions, e.g. enterprise-wide demand management, project portfolio management, and strategies & goals management, is introduced [WMFH07], followed by an in-depth discussion of the main activities of the EA management function: envisioning, documenting, planning, analyzing, and enforcing (cf. [HW08, Nie06, Sch08, WvdBLvS05]). Completing the theoretical foundation, the EAM Pattern Catalog (EAMPC)<sup>4</sup> a collection of proven best-practice solutions is presented.

### 2.4.4 Conduct and guide mini projects

In order to apply the theoretical knowledge gained in the lecture, the students are given the chance to solve a real-world EA management problem at an industry partner during a mini project. The time frame of a mini project should be chosen carefully as a too short project might on the one hand prevent the students from getting deep experience in the topic. On the other hand, a too long mini project might discourage students from selecting the lecture. The time frame for the mini projects should at least be three months to give the students enough time to get involved. Figure 3 shows an exemplary schedule with possible milestones of a mini project.

In an initial presentation each student team introduces its specific EA management problem to the fellow students and industry partners. In order to solve the specific problems, existing EA management patterns contained in the EAMPC [Cha09] can be utilized or adapted. During the solution phase, the industry partner and the organizing university members provide guidance for the students. While the university members guide the execution of the project by providing theoretical background and information on other approaches,

---

<sup>4</sup>See <http://eampc-wiki.systemcartography.info> for more information.

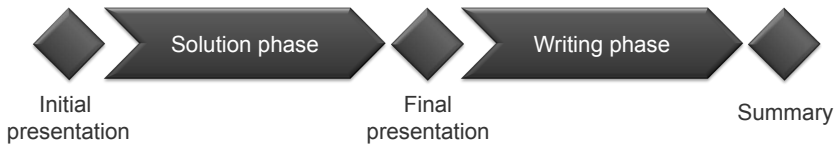


Abbildung 3: Exemplary time schedule for the mini projects

e.g. from scientific literature, the industry partner supports the student team in enterprise-specific contexts, as e.g. in making interview arrangements or providing access to internal knowledge sources like CMDBs.

To successfully complete the mini projects, the student teams have to provide three deliverables:

**Initial presentation:** Two weeks after the start of the project, the student teams should give an initial presentation, which introduces the respective problem to the other student teams and the participating industry partners. The objective of the presentation is to illustrate the idea of the chosen solution and discuss it with other student teams with possibly related problem statements. Furthermore, the industry partners may take part in the discussion to provide feedback to the proposed solution from a practitioners point of view and to get to know the students and other practitioners. The overall number of participants for such workshop should be carefully considered to foster discussion between the participants.

**Final presentation:** Two weeks prior to the project end, the final presentations should take place. The objective of each team's presentation is not only to present the developed solutions but also to discuss experienced pitfalls, surprises, and lessons learned from the student team perspective as well as the industry partner perspective.

**Summary:** Finally, the students have to submit a written summary of their mini project. These summaries can be formulated as EAM patterns, providing reusable solutions, or as case studies describing the used approach. These two forms for presenting the results are selected as they are simple to use for the students, while being further reusable for subsequent academic research.

In this phase related patterns, as e.g. *patterns for supervising thesis projects* [SS08] can provide guidance for supporting the student team in conducting their mini project.

#### 2.4.5 Evaluate results

Finally, a grading for the lecture has to take place. Therefore, the deliverables of the mini projects as well as the results of an supplementary exam can be taken into account. The diversity of problem statements and the enterprise context of each project also have to be considered. A good starting point for evaluating the work of the students in the mini project are the two presentations as well as the final report. Furthermore, the industry partners can

be asked regarding their subjective impression of the single students and the student team as a whole. This can be supported by a compact questionnaire for the industry partners containing questions as the following:

- Did the students provide EA-related input and knowledge during the project?
- Was the mini-project outcome useful for you or your organization?

To make the grading process more explicit and provide feedback to the students positive and negative bullet points should be collected, which can then be used to explain the received results.

## **2.5 Implementation**

TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT can be implemented in all universities or other higher education institutions wanting to strengthen their offering for EA management. Depending on the selected teaching case type, you will either need cooperating companies or employees with in depth knowledge about EA management to prepare the teaching cases and to guide the students during their mini projects. One practical topic for each student team is advisable, while nevertheless one company can host more than one student mini-project. A research assistant from university can oversee up to five mini-projects, although the inception phase of each project may cause some workload peaks at the beginning. On the side of the industry partner, an employee supervising each project is needed. He or she should schedule meetings with the student team at least every two weeks to keep in touch with the progress. Especially, if the project is (partially) concerned with gathering architectural information, additional employees may be involved in executing the mini-project.

The presentation of the mini project results should preferably be conducted in a larger workshop, where all participating students and companies are attending. This will support the exchange between the different mini projects and will result in extended feedback from the participants.

## **2.6 Known Uses**

The approach documented in TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT is in use at the Chair for Software Engineering for business Information Systems (sebis) at the Technische Universität München. For more details on the lecture take a look on the website [www.matthes.in.tum.de/wikis/sebis/vorlesung-eam](http://www.matthes.in.tum.de/wikis/sebis/vorlesung-eam) or in the project report from 2009 [BMSS09].

## 2.7 Consequences

The solution for teaching EA management to Wirtschaftsinformatik-master students requires a good communication between the industry partners, the students, and the teaching staff from university. This is especially true for setting up the individual teaching cases. These cases can mostly not be reused in a subsequent term, as the industry partners advance in their EA management endeavors and are likely to rise different cases in the next iteration, which they would like to see addressed by a new student team.

### **Real-world teaching case versus artificial teaching case**

Using a real-world teaching case for the mini projects results in a more realistic introduction to EA management, but it may be more difficult to prepare the teaching case to make it usable in the lecture. In addition, it requires that you have companies at hand, which participate in the mini projects and are willing to guide the students. An artificial teaching case can more easily be controlled and better be adapted to the students' demands. A drawback of an artificial teaching case is that it requires in-depth knowledge to create and prepare it.

### **One versus multiple teaching cases**

Using multiple teaching cases offers a broader overview on the topic EA management. This may be an important fact for students, which are typically novices in this topic, because they get to know its diversity, e.g. during the presentations of the other mini projects, and also get to know that the enterprise-specific context plays an important role in developing the right solution. A drawback is that you initially have to prepare the teaching cases, possibly resulting in a high effort. In contrast, using only one teaching case has the benefit that the results of the different mini projects, which all work on the same teaching case, are far more comparable and require less effort to be prepared.

### **Maximal versus minimal termtime**

Selecting the right termtime has an impact on the motivation of the participating students. In addition, it may also influence the intensity of their participation. Our experiences at Technische Universität München showed that a good term time for such mini projects is three months.

### **Size of student teams versus number of student teams**

The selection of the right size for the student teams heavily impacts the motivation of the participating students. In too large teams, the individual student tends to invest himself less, while too small teams have difficulties to attain the expected results. A team size of four students seemed appropriate for the mini projects.

## 2.8 See Also

MINI PROJECT EVALUATION documents a form, which can be used to evaluate the students work in the mini projects.

A related approach was developed by Koning et al. [KBB08], who present a defined method with a given set of problems to be addressed, complemented by predefined view-

points used to describe the EA of an industry partner. Although the approach of Koning et al. turned out to be suitable for the case study, it would not be suitable in the above described context. Due to the different industry partners participating in the practical part of the lecture and the therefore different maturity levels of the mini projects, a standardized method like the one developed by Koning et al [KBB08] would not be applicable. This is especially visible, if the different goals of the mini projects are considered, which range from basic documentation and modeling initiatives to sophisticated risk management analysis [BMSS09]. Nevertheless, in different contexts, i.e. if only one industry partner is concerned or if multiple industry partners with similar goals and levels of EA management maturity take part, this might be different.

In addition, PATTERNS FOR SUPERVISING THESIS PROJECTS can be considered for additional support on supervising students (cf. [SS08]).

### 3 Acknowledgment, Critical Reflection and Outlook

This section includes acknowledgments to the people who supported the creation of this article and provides a critical reflection. Further an outlook on future topics in teaching EA management is given.

#### 3.1 Acknowledgements

We want to thank all participants of the writer's workshop of the 2<sup>nd</sup> European Workshop on Patterns for Enterprise Architecture Management 2010 (PEAM 2010) in Paderborn, Germany and especially our shepherd Andreas Krupinski for the time they spent for reading, commenting, and discussing this article. In addition, we also want to thank the students and cooperating companies, which participated in the lecture at the Technische Universität München and provided valuable feedback on the lecture.

#### 3.2 Critical reflection

The TEACHING ENTERPRISE ARCHITECTURE MANAGEMENT} pattern presented in this article is a first step to the development of best practices for universities, which want to extend their curriculum with the EA management topic. However, the approach has only been applied once in practice at the Technische Universität München and therefore can on the one hand only be seen as a first step in the validation of the method. On the other hand the approach was used at nine different enterprises (cf. [BMSS09]) and can therefore be seen as nine different cases in which the method has proven to work. According to the often quoted *rule of three* [Cop96], a pattern describes an abstraction of at least three occurrences. When it comes to the evaluation of the method, the students' feedback as well as the practitioners' feedback provides valuable input. The majority of the students empha-

sized that the lecture prepared them well for the mini-projects, although minor remarks were made in the missing preparation for modeling tasks in the enterprise environment. An optional step of "practitioner bidding" may be helpful in selecting the "right" students for the right industry partner. In this step, the students would have to apply for an industry project giving a short curriculum vitae and some lines of motivation. Thereby, the university staff as well as the participating industry partners would be able to select the best suited students. This nevertheless rises the additional effort to create fall-back-solutions that can be performed by students that do not succeed in applying for an industry-partner.

### 3.3 Outlook

The initial application of the pattern in the context of an EA management lecture at Technische Universität München showed, that students can greatly benefit from getting "hands on" experience in applying EA management techniques. Nevertheless, not only students might be an interesting target group for EA management experience, but also practitioners might profit from getting in-depth training before actually conducting EA management in their enterprise. This is further reflected by the fact that a majority of industry partners are willing to host student mini-projects for a second time. The pattern can further provide a valuable starting point for creating training material for consultancies willing to provide EA management trainings. A minor problem might arise in this context from the highly practice-oriented setting of the method, which may pose a problem, as practical cases for training might be hard to find.

## Literatur

- [BEL<sup>+</sup>07] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Kathrin Schneider und Christian M. Schweda. A pattern based Approach for constructing Enterprise Architecture Management Information Models. In *Wirtschaftsinformatik 2007*, Seiten 145–162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.
- [BEL<sup>+</sup>08] Sabine Buckl, Alexander Ernst, Josef Lankes, Florian Matthes und Christian M. Schweda. Enterprise Architecture Management Patterns – Exemplifying the Approach. In *The 12<sup>th</sup> IEEE International EDOC Conference (EDOC 2008)*, Munich, Germany, 2008. IEEE Computer Society.
- [BELM08] Sabine Buckl, Alexander M. Ernst, Josef Lankes und Florian Matthes. Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Bericht, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.
- [BEMS09] Sabine Buckl, Alexander M. Ernst, Florian Matthes und Christian M. Schweda. How to make your enterprise architecture management endeavor fail! In *Pattern Languages of Programs 2009 (PLoP 2009)*, Chicago, 2009.
- [BMSS09] Sabine Buckl, Florian Matthes, Christopher Schulz und Christian M. Schweda. Teaching Enterprise Architecture Management – A Practical Experience. Bericht,



- Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2009.
- [Cha09] Chair for Informatics 19 (sebis), Technische Universität München. EAM Pattern Catalog Wiki. <http://eampc-wiki.systemcartography.info> (cited 2010-02-25), 2009.
- [Cop96] James Coplien. *Software Patterns: Management Briefs*. Cambridge University Press, Cambridge, UK, 1996.
- [Ern08] Alexander Ernst. Enterprise Architecture Management Patterns. In *PLoP 08: Proceedings of the Pattern Languages of Programs Conference 2008*, Nashville, USA, 2008.
- [Ern10] Alexander M. Ernst. *A Pattern-Based Approach to Enterprise Architecture Management*. Dissertation, Technische Universität München, München, Germany, 2010. (in submission).
- [HW08] Marting Hafner und Robert Winter. Processes for Enterprise Application Architecture Management. In *41<sup>st</sup> Hawaii International International Conference on Systems Science (HICSS-41 2008), Proceedings, 7-10 January 2008, Waikoloa, Big Island, HI, USA*, Seite 396, Los Alamitos, CA, USA, 2008. IEEE Computer Society.
- [Int07] International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.
- [KBB08] Henk Koning, Rik Bos und Sjaak Brinkkemper. A Lightweight Method for the Modelling of Enterprise Architectures. In Springer, Hrsg., *Service-Oriented Computing – ICSOC 2008 Workshops*, Seiten 375–387, Berlin, Heidelberg, Germany, 2008.
- [Nie06] Klaus D. Niemann. *From Enterprise Architecture to IT Governance – Elements of Effective IT Management*. Vieweg+Teubner, Wiesbaden, Germany, 2006.
- [Sch08] Marten Schönherr. Towards a common terminology in the discipline of Enterprise Architecture. In Stephan Aier, Pontus Johnson und Joachim Schelp, Hrsg., *Pre-Proceedings of the 3<sup>rd</sup> Workshop on Trends in Enterprise Architecture Research*, Seiten 107–123, Sydney, Australia, 2008.
- [SS08] Axel Schmolitzky und Till Schümmer. Patterns for Supervising Thesis Projects. In *Proceedings of EuroPLoP 2008*, 2008.
- [SZ92] John F. Sowa und John A. Zachman. Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, 31(3):590–616, 1992.
- [The09] The Open Group. TOGAF ”Enterprise Edition” Version 9. <http://www.togaf.org> (cited 2010-02-25), 2009.
- [WMFH07] André Wittenburg, Florian Matthes, Florian Fischer und Thorsten Hallermeier. Building an integrated IT governance platform at the BMW Group. *International Journal Business Process Integration and Management*, 2(4), 2007.
- [WvdBLvS05] Roel Wagter, Martin van den Berg, Joost Luijpers und Marlies van Steenbergem. *Dynamic Enterprise Architecture: How to Make IT Work*. John Wiley, 2005.



**Produktlinien im Kontext: Technologie, Prozesse, Business  
und Organisation**

**(PIK 2010)**



# Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation

## PIK2010 Workshop-Einleitung und Überblick

Andreas Birk<sup>1</sup>, Klaus Schmid<sup>2</sup>, Markus Völter<sup>3</sup>

<sup>1</sup> Software.Process.Management, Usedomstraße 15, 70439 Stuttgart  
andreas.birk@swpm.de

<sup>2</sup> Universität Hildesheim, Institut für Informatik  
Marienburger Platz 22, 31141 Hildesheim  
schmid@sse.uni-hildesheim.de

<sup>3</sup> voelter - ingenieurbüro für softwaretechnologie/itemis  
Öztaler Straße 38, 70327 Stuttgart  
voelter@acm.org

**Abstract:** Produktlinien sind heute in vielen Bereichen der Software-Industrie vertreten. Dennoch bergen sie für viele Unternehmen noch bedeutende Herausforderungen und Risiken. Der Workshop beleuchtet aktuelle Erfahrungen mit Produktlinien und bietet ein Forum für den Dialog zwischen Praxis und anwendungsorientierter Forschung. Im Mittelpunkt steht das Wechselspiel zwischen technischen Fragestellungen und den geschäftlichen, organisatorischen und Prozessaspekten.

## 1 Produktlinien

Produktlinien sind heute in vielen Bereichen der Software-Industrie vertreten, von eingebetteten Systemen bis zu betrieblichen Informationssystemen. Sie ermöglichen höhere Produktivität, steigern die Qualität und verbessern die strategischen Positionen der Unternehmen, u.a. aufgrund eines hohen Grades an Wiederverwendung und Standardisierung.

Dennoch bergen Produktlinien für viele Unternehmen noch bedeutende Herausforderungen und Risiken. Die Gründe liegen teilweise im technischen Bereich. So sind viele Produktlinien-Technologien für den breiten Einsatz in der Praxis noch nicht genügend ausgereift und miteinander integriert. Die wohl größten Herausforderungen stellen sich in den Wechselwirkungen zwischen den technischen Verfahren mit den Prozessen sowie dem organisatorischen und geschäftlichen Kontext der Produktlinienentwicklung. —Wie müssen die technologischen Ansätze auf diese Wechselwirkungen ausgerichtet sein? Welche Potenziale bieten neue technologische Entwicklungen in unterschiedlichen Einsatzfeldern?

Der Workshop „Produktlinien im Kontext“ [PIK10] will aktuelle Erfahrungen mit Produktlinien beleuchten und den Dialog zwischen Praxis und anwendungsorientierter Forschung fördern. Im Mittelpunkt steht das Wechselspiel zwischen technischen Fragestellungen und den geschäftlichen, organisatorischen und Prozessaspekten. Daneben sollen auch neue technologische Entwicklungen vorgestellt und diskutiert werden.

## 2 Themengebiete

Die angenommenen Workshop-Beiträge behandeln die folgenden Themengebiete:

- Variantenmanagement bei Software- und System- Anforderungen
- Werkzeuginfrastrukturen für die modellgetriebene Software-Entwicklung
- Domänenspezifische Sprachen (Domain-specific Languages; DSL) in der Produktlinien-Entwicklung
- Testen und Testmanagement von Software-Produktlinien

Der Workshop beinhaltet einen ausführlichen Diskussionsteil. Die Diskussionen sind offen für Themen, die sich während des Workshops ergeben. Zusätzlich zu den oben genannten Themen betrachten die Diskussionen zumindest die folgenden Aspekte:

- Einführung und Evolution von Software-Produktlinien
- Die Besonderheiten von Produktlinien in verschiedenen Geschäftsdomänen (z.B. Embedded Systems versus IT-Anwendungssysteme)

## 3 Beiträge und Diskussionen

Dieses Kapitel stellt die Workshop-Beiträge kurz vor und präsentiert die vorgesehenen Diskussionsblöcke. Die Diskussionen werden nach dem Workshop zusammengefasst und auf der Workshop-Homepage veröffentlicht [PIK10].

Der Beitrag von Ekaterina Boutkova [Bo10] untersucht die Herausforderungen an die Variantenmodellierung in der Automobilindustrie. Ausgangspunkt ist eine Analyse der Anforderungsdokumente, wie sie bei der Daimler AG in der Pkw-Sparte verwendet werden. Darin werden verschiedene Verfahren zur Variabilitätsmodellierung identifiziert, zum Beispiel textuelle Variantendokumentation in Attributspalten oder durch Kapitelstruktur, jeweils ein separates Modul im Requirements-Management-Werkzeug oder der Code-Regel-Ansatz.

Die Untersuchung zeigt, dass in einer komplexen und langfristig ausgelegten Produktentwicklung die Zuordnung von Anforderungen zu Varianten, und damit auch die Wiederverwendung von Anforderungen keinesfalls trivial sind. Die Herausforderungen liegen in Bereichen wie der Größe und Heterogenität von Lastenheften, der Komplexität von Variabilität, der Etablierung der gewünschten Modellierungsverfahren bei den Ingenieuren, sowie der Einrichtung der erforderlichen IT-Infrastruktur. Die Autorin arbeitet an einer Lösung für die geschilderten Probleme bei der Zuordnung von Anforderungen zu Varianten.

Thomas Buchmann, Alexander Dotor und Bernhard Westfechtel [BDW10] stellen mit der modellgetriebenen Entwicklung von Versionskontrollsystemen ein Fallbeispiel für das Produktlinien-Engineering vor. Speziell untersuchen sie die Herausforderungen an eine umfassende Werkzeugkette, die den modellgetriebenen Ansatz unterstützt. Die Fallstudie veranschaulicht, dass Werkzeuge für Feature-basierte Variantenmodellierung und für die Modellierung von Einzelsystemen am Markt vorhanden sind. Eigene Entwicklungen waren erforderlich für die Verbindung von Feature- und Domänenmodell sowie für die Architekturmodellierung. Bedarf für Verbesserungen des existierenden Verfahrens in der Fallstudie sehen die Autoren insbesondere in einer besseren Unterstützung für die Architektur- und Komponentenmodellierung, um das Domänenmodell besser handhaben zu können.

Ein neues Verfahren zur Entwicklung von Produktlinien stellt Markus Völter vor [Vö10]. Es beruht auf sogenannten Projectional Language Workbenches. Dies sind Werkzeugumgebungen, die eine Sammlung miteinander verbundener domänenspezifischer Sprachen verwalten. Jede dieser Sprachen ist jeweils eine Sicht (bzw. Language Projection) auf eine zugrunde liegende zentrale Repräsentation. Bei dieser Repräsentation handelt es sich um direkt ausführbare bzw. compilierbare Programmiersprache (z.B. Java oder C). Der Modellierer arbeitet ausschließlich mit einer (Teil-)Menge von DSL. Die Projectional Language Workbench setzt seine Eingaben direkt auf der programmiersprachlichen Basisrepräsentation um.

Völter beschreibt ein Konzept, bei dem Projectional Language Workbenches für die Modellierung mit domänenspezifischen Sprachen kombiniert werden mit Konfigurationstabellen zur Variabilitätsmodellierung. Zurzeit erstellt er mit Kollegen eine Beispielimplementierung, die auf die industrielle Anwendung ausgerichtet ist.

Andreas Wübbeke und Sebastian Oster [WO10] schlagen ein Verfahren zum Testen von Software-Produktlinien vor, das im Domain-Engineering kombinatorische Plattformtests verwendet und beim Application-Engineering auf die selektive Wiederverwendung von Testfällen setzt. Dieser Ansatz zeichnet sich durch einen reduzierten Testaufwand aus. Dabei wird erwartet, dass das kombinatorische Testen eine als ausreichend hoch betrachtete Testabdeckung gerade bei qualitätskritischen Systemaspekten erreicht.

Das Verfahren wurde am Beispiel des Forschungsprototyps einer Mobiltelefon-Produktlinie entwickelt und damit evaluiert. Die Autoren planen den Ausbau des Verfahrens und weitere Evaluationen mit industriellen Szenarien.

Für die Diskussionen sind drei Sessions vorgesehen. Eine erste Diskussion behandelt Fragestellungen im Umfeld der Werkzeuginfrastruktur für die Produktlinien-Entwicklung. Sie geht aus von den Beiträgen von Buchmann, Dotor und Westfechtel, sowie von Völter. Die zweite Diskussion bezieht sich insbesondere auf die Beiträge von Boutkova sowie von Wübbecke und Oster. Ihre Themenschwerpunkte sind also das Testen von Produktlinien sowie die Beschreibung von Varianten im Rahmen des Anforderungsmanagements. Den Einstieg in diese Diskussion bildet ein Position Statement von Gerald Heller über Erfahrungen mit dem Testen von Software-Produktlinien in einer globalen industriellen Produktentwicklung.

Alle Diskussions-Sessions sollen insbesondere von den aktuellen Interessen und Anliegen der Workshop-Teilnehmer geprägt sein. Der dritte Diskussionsteil ist hierauf besonders ausgerichtet. Er besitzt keine verbindliche Themenvorgabe. Als Vorschläge von Seiten der Workshop-Organisatoren sind vorbereitet: Die Einführung von Software-Produktlinien sowie Unterschiede in verschiedenen Umfeldern der Produktlinien-Entwicklungen—wie beispielsweise Spezifika von Geschäftsdomänen und die Organisationsstruktur der Software-Entwicklung. Außerdem werden hier auch die relevanten Diskussionsergebnisse der zurückliegenden PIK-Tagungen eingebracht [BFS08][Bi09]. Ein abschließender Programmteil fasst die Diskussionen zusammen, identifiziert die Kernpunkte und zeigt Querbeziehungen zwischen den Teildiskussionen auf. Die Ergebnisse der Diskussion werden auf der PIK-Webseite veröffentlicht [PIK10].

## Literaturverzeichnis

- [BDW10] Buchmann, T.; Dotor, A.; Westfechtel, B.: Werkzeuge zur modellgetriebenen Entwicklung von Produktlinien: Ein Erfahrungsbericht am Beispiel von Versionskontrollsystemen. In: Software Engineering 2010 (Workshopband). Lecture Notes in Informatics (LNI), P-160, Bonn, Germany, 2010.
- [BFS08] Birk, A.; Fiege, L.; Schmid, K.: Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation. PIK2008 Workshop-Einleitung und Überblick. In: Software Engineering 2008 (Workshopband). Lecture Notes in Informatics (LNI), P-122, Bonn, Germany, 2008.
- [Bi09] Birk, A.; Fiege, L.; Kolagari, R.T.; Schmid, K.: Tagungsband der PIK 2009: Produktlinien im Kontext. Universität Hildesheim, Institut für Informatik, Hildesheim, Germany, 2009.
- [Bo10] Boutkova, E.: Herausforderungen für Variantenmanagement in Anforderungsdokumenten. In: Software Engineering 2010 (Workshopband). Lecture Notes in Informatics (LNI), P-160, Bonn, Germany, 2010.
- [PIK10] PIK 2010 - Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation. URL: <http://2010.pik-konf.de/>
- [Vö10] Völter, M.: Projectional Language Workbenches as a Foundation for Product Line Engineering. In: Software Engineering 2010 (Workshopband). Lecture Notes in Informatics (LNI), P-160, Bonn, Germany, 2010.
- [WO10] Wübbecke A.; Oster, S.: Verknüpfung von kombinatorischem Plattform- und individuellem Produkt-Test für Software-Produktlinien. In: Software Engineering 2010 (Workshopband). Lecture Notes in Informatics (LNI), P-160, Bonn, Germany, 2010.

## **PIK 2010 Workshop-Organisation**

### **Organisatoren**

Andreas Birk, SWPM

Klaus Schmid, Universität Hildesheim

Markus Völter, Völter Ingenieurbüro und itemis

### **Programmkomitee**

Danilo Beuche, pure-systems

Ulrich Eisenecker, Universität Leipzig

Ludger Fiege, Siemens

Samuel Fricker, Fuchs Informatik und  
Universität Zürich

Paul Grünbacher, Universität Linz

Gerald Heller,  
Software Process Consulting

Georg Herzwurm, Universität Stuttgart

Frank Houdek, Daimler

Dirk Janzen, Harman/Becker

Isabel John, Fraunhofer IESE

Olaf Kath, ikv++

Michael Kircher, Siemens

Hans Körber, Universität Leipzig

Andreas Kreß, HOOD-Group

Kim Lauenroth,  
Universität Duisburg-Essen

Klaus Marquardt, Dräger Medical

Thomas von der Maßen,  
Arvato/Bertelsmann

Klaus Pohl, Universität Duisburg-Essen

Mark-Oliver Reiser, TU Berlin

Michael Schulze-Ruhfus, oose

Dietmar Schütz, Siemens

Reinhard Stoiber, Universität Zürich

Ramin Tavakoli, TU Berlin

Matthias Weber, Carmeq





# Herausforderungen für Variantenmanagement in Anforderungsdokumenten

Ekaterina Boutkova

Group Research and Advanced Engineering  
Daimler AG  
Postfach 2360  
89013 Ulm

ekaterina.boutkova@daimler.com

**Abstract:** Effektive Methoden für Variantenmanagement in Anforderungsdokumenten sind die essentielle Voraussetzung für eine effektive und effiziente Wiederverwendung von Anforderungen, die zu einer Entwicklungszeitreduktion und somit zu einem Wettbewerbsvorteil führt. Dieser Beitrag beschreibt die aktuelle Vielfalt der Anforderungsdokumente eines Automobilherstellers am Beispiel der Daimler AG. Außerdem stellt dieser Beitrag die Herausforderungen für das Variantenmanagement in Anforderungsdokumenten vor, die während einer Studie zu existierenden Variabilitätsmanagementansätzen bei der Daimler AG identifiziert wurden.

## 1 Einleitung

Die Automobilindustrie ist durch eine große Anzahl an Produktvarianten charakterisiert. Die Gründe für die Variantenvielfalt sind vielseitig. Als ein prominentes Beispiel können im Fall der PKW-Sparte der Daimler AG die Modellreihen (z.B. C-Klasse, E-Klasse oder S-Klasse) betrachtet werden. Dabei werden für jedes Modell mehrere Ausführungsarten (z.B. Limousine, Coupé, T-Modell, Cabriolet) und Ausstattungsvarianten (z.B. Classic, Elegance) angeboten, die die unterschiedlichen Kunden- bzw. Marktanforderungen berücksichtigen.

Mit der steigenden Komplexität der Fahrzeuge sind auch die Anforderungen an die einzelnen Komponenten und Systemen immer vielfältiger geworden. Reichte früher eine Zeichnung und Stückliste für die Anforderungsdokumentation einer Komponente, so ist heute ein umfassendes Lastenheft erforderlich. Ein Lastenheft bündelt die Anforderungen aller im Produktentstehungsprozess beteiligten Bereiche (z.B. Entwicklung, Produktion) und bildet die Basis für die Ausschreibung. Die Schwächen eines Lastenheftes (z.B. unvollständige oder fehlerhafte Anforderungen) führen häufig zu einer Vielzahl von Änderungen und Terminverzögerungen im Entwicklungsprozess und damit zur Kostenüberschreitung.

Die Reduktion der Entwicklungszeit durch die Wiederverwendung ist der Leitgedanke der letzten Jahrzehnte in der Automobilindustrie. Die Anforderungsdokumente in der Automobilindustrie bieten ein großes Wiederverwendungspotential ist ein effektives und effizientes Variantenmanagement. Die effektive Methoden des Variantenmanagements für Anforderungsdokumente sind die Voraussetzung für die erfolgreiche Wiederverwendung von Anforderungen, weil das Variantenmanagement die Spezifizierung mehrere Produkte einer Produktlinie durch Wiederverwendung von Anforderungen unterstützt.

Das Ziel unserer Arbeit ist eine Verbesserung des Variantenmanagements in Lastenheften. In diesem Beitrag wird die Welt der Anforderungsdokumente bei der Daimler AG vorgestellt (Kapitel 2). Daraus folgende Herausforderungen für das Variantenmanagement werden im Kapitel 3 präsentiert. Kapitel 4 beschreibt die aktuellen und geplanten Arbeiten in diesem Gebiet.

## **2 Anforderungsmanagement bei der Daimler AG**

Dieses Kapitel präsentiert die Vielfalt der Anforderungsdokumente bei der Daimler AG (Kapitel 2.1) sowie die aktuellen Ansätze zum Variabilitätsmanagement (Kapitel 2.2).

### **2.1 Anforderungsdokumente**

Für die Dokumentation der Anforderungen bei der Daimler AG existieren im Wesentlichen drei Typen von Lastenheften: das Fahrzeuglastenheft, das Systemlastenheft sowie das Komponentenlastenheft. Für jeden Lastenheft-Typ existieren entsprechenden Standardvorlagen, die den Entwicklungs- und Spezifikationsprozess durch die Verwendung standardisierter Inhalte und Anforderungspakete unterstützen.

Ein Fahrzeuglastenheft (FLH) enthält eine umfangreiche Beschreibung des neuen Fahrzeugs. Im FLH werden zum Beispiel die Marketingaspekte (z.B. Zielgruppen, Produktpositionierung auf den Märkten), Designanforderungen und technischen Konzepte des neuen Fahrzeuges festgehalten.

Ein Systemlastenheft (SLH) bildet die Brücke zwischen den Anforderungen an das Gesamtfahrzeug und den Anforderungen an die einzelnen Komponenten. Ein System ist in diesem Fall kein physikalisches sondern ein logisches. Es besteht aus Fahrzeugfunktionen, die ein Kunde benutzen kann (z.B. Außenlicht, Sitzheizung), deren Eigenschaften sowie deren Komponentenbeschreibungen, die diese Funktionalität realisieren. Die Notwendigkeit des SLH entsteht aus der Problematik, die Komponentenanforderungen direkt aus Fahrzeuganforderungen vollständig und widerspruchsfrei abzuleiten. Das SLH ist eine interne Spezifikation und wird in der Regel nicht an den Lieferanten weiter gegeben.

Ein Komponentenlastenheft (KLH) dokumentiert die Anforderungen an eine einzelne Komponente (z.B. eine Sonnenblende, einen Stoßfänger). Es bildet die Basis für die Ausschreibung und den Vertrag mit dem Lieferanten. In einem KLH werden sowohl technische als auch nichttechnische (z.B. prozessuale) Anforderungen an die Komponententwicklung und spätere Serienproduktion dokumentiert. Die Anforderungen im KLH müssen alle beschlossenen Varianten (Farbvarianten, Ausstattungsvarianten, Ländervarianten etc.) umfassen.

Die Beziehungen zwischen einzelnen Lastenhefttypen sind nicht 1:1 (siehe Abbildung 1), weil ein System in mehreren Fahrzeugen implementiert werden kann. Des Weiteren können mithilfe ein und derselben Komponente die unterschiedlichen Systemfunktionen realisiert werden. Zum Beispiel die Komponente Türschloss realisiert sowohl die Funktionen des Systems Schließung als auch die des Systems Außenlicht. Zusätzlich kann das System Schließung mit gleichen Anforderungen für mehrere Baureihen verwendet werden.

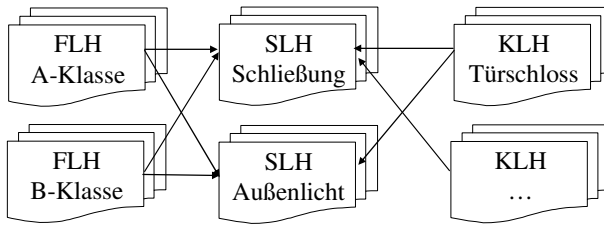


Abbildung 1: Die Beziehungen zwischen FLH-SLH-KLH

## 2.2 Variantenmanagement in Anforderungsdokumenten

Die Systeme und Komponenten für die verschiedenen Baureihen weisen oft nur geringe Unterschiede auf. Deshalb existiert ein großes Potenzial für die Wiederverwendung der Anforderungen. Diese Erkenntnis erlaubt es, die Varianten eines Systems bzw. einer Komponente im gleichen Lastenheft zu spezifizieren.

Wichtig ist, dass alle Anforderungen eines Lastenheftes zu den jeweiligen Baureihen und Varianten nachvollziehbar zugeordnet sein müssen, um die Erstellung von baureihenspezifischen Lastenheften für eine System- bzw. Komponentenvariante zu ermöglichen. Diese Lastenhefte werden für die Ausschreibungen benötigt, da die Lieferanten eine Spezifikation für eine konkrete Komponentenvariante, die für eine bestimmte Baureihe festgelegt wurde, benötigen.

In diesem Beitrag werden die existierenden Ansätze benannt und deren Vor- und Nachteile kurz vorgestellt. Die vollständige Studie zu dem State-of-the-Practice kann aus [Bo08] entnommen werden. Im Rahmen der Studie zum State-of-the-Practice wurden folgende Ansätze identifiziert, analysiert und bewertet:

1. Variantendokumentation im Anforderungstext

2. Variantendokumentation in Attributsspalten
3. Variantendokumentation mit Multi-Enumeration-Attribut
4. Variantendokumentation durch Kapitelstruktur
5. Variantendokumentation in unterschiedlichen Doors-Modulen
6. Variantendokumentation im Anforderungspool
7. Ansatz mit Stücklistencode zur Variabilitätsbeschreibung (Code-Regel-Ansatz)

Zu den Vorteilen dieser Ansätze gehören eine einfache Handhabung (1-4), eine transparente Zuordnung von Anforderungen zu den Varianten sowie die Verbindung mit Stücklisten (7). Zu den Nachteilen gehören ein hoher Aufwand und eine hohe Fehlerwahrscheinlichkeit bedingt durch die manuelle Zuordnung sowie die fehlende Möglichkeit diese Zuordnung zu überprüfen. Außerdem skalieren diese Ansätze schlecht. Bei einer großen Anzahl von Varianten geht der Überblick schnell verloren. Ein zusätzlicher Nachteil des Code-Regel-Ansatzes ist eine nicht intuitive Anwendung der verwendeten Codes.

### **3 Herausforderungen für Variantenmanagement**

In diesem Kapitel werden Herausforderungen für das Variantenmanagement beschrieben, die während der Analyse der existierenden Variantenmanagementansätze in Lastenheften bei der Daimler AG identifiziert wurden.

#### **3.1 Umfang und Heterogenität der Lastenhefte**

Die meisten Lastenhefte bei der Daimler AG sind sehr groß. Ein Lastenheft für eine Komponente beinhaltet zwischen 1.000 und mehreren 10.000 Anforderungen. Ein Lastenheft für ein System beinhaltet zwischen einigen 100 und mehreren 1.000 Anforderungen.

Die unterschiedlichen Komponenten bzw. Systeme haben eine unterschiedliche Anzahl von variablen Anforderungen. Auch die Anzahl der Variabilitätsmerkmalen (Variabilitätskriterien und deren Ausprägungen) ist bei allen Komponenten bzw. Systemen unterschiedlich. Die Begriffe Variabilitätskriterium und Ausprägung des Variabilitätskriteriums wurden als Spezifizierung des Begriffs Merkmal in [Bo08] definiert.

Ein Variantenmanagement-Ansatz für Lastenhefte muss alle diese Besonderheiten berücksichtigen. Der Ansatz muss skalierbar sein, so dass er sowohl für die Lastenhefte mit mehreren 10.000 Anforderungen als auch für die Lastenhefte mit paar 100 Anforderungen anwendbar ist. Der Ansatz muss Lastenhefte mit wenigen sowie auch solche mit vielen variablen Anforderungen im gleichen Maße unterstützen. Außerdem muss der gesuchte Ansatz es erlauben, mehrere hundert Variabilitätsmerkmale zu verwalten.

### 3.2 Use Cases für Variabilitätsmanagement

Das Variantenmanagement in Lastenheften kann mehrere Use Cases unterstützen, einige davon werden in diesem Abschnitt vorgestellt. Die Voraussetzung für alle Use Cases ist, dass die Zuordnung von jeder einzelnen Anforderung in der Spezifikation zu den ausgewählten Varianten immer bekannt ist. Im letzten Abschnitt wurden die typischen Umfänge von Lastenheften diskutiert. Somit es ist verständlich, dass der Aufwand für die Zuordnung von einzelnen Anforderungen zu Varianten enorm ist und dieses Zuordnungsproblem die größte Herausforderung für das Variabilitätsmanagement in Lastenheften darstellt.

Die Entwicklung bzw. Weiterentwicklung einer Komponente bzw. eines Systems geschieht in Rahmen der Entwicklung eines neuen Fahrzeuges (einer neuen Baureihe). Für jede neue Baureihe werden entsprechend die neuen Lastenhefte erstellt. Viele Komponenten bzw. Systeme unterscheiden sich von Baureihe zur Baureihe nicht existenziell. Deswegen werden oft die Anforderungen einer Baureihe als Basis für die neue Baureihe übernommen und angepasst. Zum Beispiel können die Anforderungen für den Fensterheber einer E-Klasse für den Fensterheber der C-Klasse kopiert und an wenigen Stellen (z.B. Geometriedaten) angepasst werden.

Eine andere Möglichkeit der Wiederverwendung von Anforderungen ist die Dokumentation von mehreren Varianten einer Komponente (bzw. eines Systems) im gleichen Lastenheft. Es ist möglich die Anforderungen für mehrere Baureihen (z.B. A-Klasse und B-Klasse) oder für unterschiedliche Varianten einer Komponente (z.B. Sonnenblende mit Beleuchtung und Sonnenblende ohne Beleuchtung) innerhalb einer Baureihe im gleichen Lastenheft zu dokumentieren. Natürlich muss es möglich sein, die Baureihen-Varianten und Komponenten-Varianten zusammen in einem Lastenheft zu dokumentieren. Für die Ausschreibung werden die Lastenhefte einer konkreten Variante bzw. einer Auswahl an Varianten benötigt, die für eine bestimmte Baureihe geplant sind. Um die Ausleitung von variantenspezifischen Lastenheften zu ermöglichen, ist eine nachvollziehbare Zuordnung von Anforderungen zu Varianten unabdingbar.

Ein weiterer Use Case ist die Verbindung der Anforderungen mit den Stücklistennummern. Das Wissen über die Zuordnung von Anforderungen zu Varianten kann die Verbindung zwischen Anforderungen und Stücklisten herstellen. Wenn diese Information zur Verfügung steht, erlaubt sie im Fall eines Fehlers, der während der Produktion auftritt, anhand der Komponenten-Teilenummer alle Anforderungen der fehlerhaften Komponente zu identifizieren. Danach kann die Korrektheit der Anforderung überprüft werden und bei Bedarf können die Gegenmaßnahmen durchgeführt werden. Außerdem wäre es möglich, anhand der Zuordnungsinformationen alle anderen Varianten dieser Komponenten, die die gleichen Anforderungen implementieren, zu identifizieren.

Die nachvollziehbare Zuordnung von Anforderungen zu Varianten unterstützt auch das Änderungsmanagement. Sollte eine Anforderung während des Entwicklungsprozesses geändert werden, so wäre es immer möglich, alle betroffenen Baureihen und Komponenten- bzw. System-Varianten zu identifizieren.

### **3.3 Menschlicher Faktor**

Die Komponenten- und System-Anforderungen werden bei der Daimler AG von den Entwicklern spezifiziert. Als Experten im Bereich der Produktentwicklung sind sie entsprechend als Maschinenbauer oder Ingenieure ausgebildet. Der Großteil dieser Spezialisten hat jedoch kein umfangreiches Know-how in den Informatik- und Anforderungsmanagement-Bereichen. Die Hauptaufgabe eines solchen Spezialisten besteht in der Entwicklung von Systemen und System-Komponenten. Für die Erstellung eines Lastenheftes haben die Entwickler 2-6 Monate zur Verfügung. Nach der Erstellungsphase sind die Entwickler „nur“ für die Änderungspflege in ihren Lastenheften zuständig. Die Zeitspanne zwischen der Erstellung von neuen Lastenheften beträgt 3-5 Jahre.

Daraus entsteht eine weitere Anforderung an den Variantenmanagement-Ansatz: Er muss die Fähigkeiten der zukünftigen Anwender berücksichtigen. Er muss einfach und intuitiv bedienbar sein. Die Software-Unterstützung muss es ermöglichen, das Variabilitätsmodell einfach grafisch zu visualisieren. Die Anwendung von Prädikaten-Logik oder programmier-ähnlichen Sprachen wird von den potenziellen Anwendern nicht akzeptiert.

### **3.4 IT-Infrastruktur**

Für die Verwaltung von SLH und KLH wird das Tool Doors von IBM/Telelogic verwendet. Ein SLH bzw. KLH besteht aus einem bzw. mehreren Doors-Modulen, die jeweils mehrere tausend Doors-Objekte beinhalten können. Die Standardvorlagen für SLH und KLH sind als Kopiervorlagen in Form von Doors-Modulen verfügbar und bestehen aus circa 150 Seiten. Ein SLH bzw. KLH hat 1-2 Autoren, aber es existieren auch Lastenhefte, an denen bis zu 40 Entwickler parallel arbeiten.

Doors ist ein Standard-Tool bei der Daimler AG für das Anforderungsmanagement auf der System und Komponenten-Ebene. Daraus folgt, dass die Kompatibilität des Variantenmanagementansatzes mit Doors eine notwendige Bedingung darstellt. Andererseits muss der Variantenmanagementansatz eine tool-unabhängige Basis haben, um auch in Zukunft mit einem potenziell anderen Anforderungsmanagement-Tool funktionsfähig zu bleiben.

### **3.5 Landschaft der Lastenhefte**

Eine weitere Herausforderung sind die Standardvorlagen, die für alle Lastenheftstypen existieren. Ein Standardlastenheft beinhaltet einigen Standardanforderungen die nicht gelöscht oder verändert werden dürfen, aber auch die Kapitel, deren Befüllung der Entwickler komponentenspezifisch vornehmen muss (Verantwortlichkeiten, Termine, Gewichte). Weiterhin enthält die Standardvorlage KLH Kapitel, die dem Entwickler im Sinne einer Checkliste dienen und auf deren Basis der Entwickler die technische Anforderungen beschreiben kann.

Die KLH-Standardvorlage besteht zurzeit aus circa 150 Seiten und wird ständig weiterentwickelt. Gründe für Änderungen der Standardvorlage sind Erfahrungen aus abgeschlossenen Projekten sowie Verbesserungen zu den identifizierten Defizienten in der existierenden Vorlage. In einigen Fällen kommt es auch zu Strukturänderungen der Standardvorlage. In einem solchen Fall müssen alle komponenten-spezifische Anforderungen neu angeordnet werden (siehe Abbildung 2). Derartige Umstrukturierungen von mehreren tausenden Anforderungen in mehreren hunderten Lastenheften sind sehr kostenintensiv. Deshalb wurde ein Konzept zur Trennung von Inhalten ausgearbeitet.

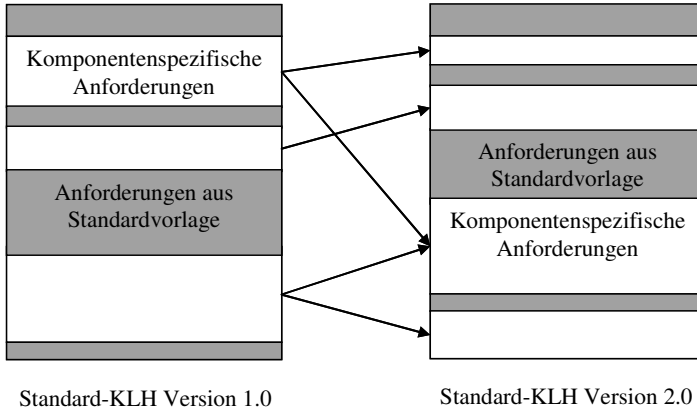


Abbildung 2: Verteilung der Anforderungen nach den Strukturänderungen

Die Grundidee ist die Teilung der Komponentenanforderungen in zwei Gruppen: die Anforderungen an die Komponente (definieren die Komponente selbst) und die komponentenspezifischen Projekt-, Produktions- und Logistik-Anforderungen. Die Anforderungen an eine Komponente werden in einem Anforderungspool verwaltet. Dabei sind in diesem Pool die Anforderungen für alle Baureihen sowie alle definierten Varianten enthalten. Bei der Erstellung eines KLH werden die relevanten Anforderungen aus dem Pool in die Kopie der Standardvorlage kopiert und bei Bedarf angepasst. Die komponentenspezifischen Projekt-, Produktions- und Logistik-Anforderungen werden weiterhin in einer Kopie der KLH-Standardvorlage gepflegt. Dieses Konzept bringt eine Reihe Vorteile mit sich. Zum Beispiel die Anforderungen an eine Komponente werden in einem Anforderungspool über den gesamten Produktentstehungsprozess dokumentiert. Der andere Vorteil ist, dass durch die Speicherung der Anforderungen in einem Anforderungspool die Wiederverwendung von Anforderungen unterstützt. Die Herausforderungen sind das Varianten- und Änderungsmanagement.

Der Variantenmanagement-Ansatz muss die bestehende Lastenheft-Landschaft (FLH-SLH-KLH) und weitere existierende Konzepte, sowie die Existenz der Standardvorlagen berücksichtigen.



### 3.6 Interne und externe Variabilität

Die aktuellen Spezifikationsprozesse sowie die Entwicklungsprozesse sind an die Entwicklung der Fahrzeugalette gekoppelt. So werden bei der Entwicklung einer neuen Baureihe die existierenden Anforderungen für eine Komponente im Hinblick auf die Eigenschaften und Funktionalitäten eines neuen Produktes bewertet. Als Produkt kennzeichnen wir eine Komponenten-Variante, die zur Produktion ausgewählt wurde.

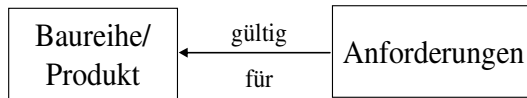


Abbildung 3: Aktuelle Zuordnung

Die Analyse der Studie zu aktuellen Variantendokumentationsansätzen hat gezeigt, dass eine Trennung der internen und externen Variabilität notwendig ist. Die interne Variabilität beschreibt die Merkmale einer Komponente bzw. eines Systems, die variieren können (z.B. Farbe, Material). Die externe Variabilität beschreibt solche Aspekte, die die Gründe für die Existenz der internen Variabilität sind. Zum Beispiel existieren zwei Varianten der Tachometeranzeige: die Geschwindigkeit wird in km/h oder in mph angezeigt. Das Merkmal der internen Variabilität ist Geschwindigkeitsanzeige. Die externe Variabilität ist zum Beispiel Markt (z.B. Europa oder USA). Welche Anzeige in einem Fahrzeug eingebaut wird, hängt vom Zielmarkt ab. Durch diese Trennung entsteht ein neues Prozessbild (siehe Abbildung 4).

Bei der neuen Vorgehensweise werden die Anforderungen den Merkmalen (Variabilitätskriterien bzw. deren Ausprägungen) zugeordnet. Diese Zuordnung ist über die Zeit konstant. Neue Produkte, die als Konfigurationen der Variabilitätsmerkmale definiert sind, können dann zur jeweiligen Baureihen zugeordnet werden. Eine ähnliche Vorgehensweise (Product-Sets-Ansatz) wurde von Reiser und Weber in [RW05] vorgeschlagen.

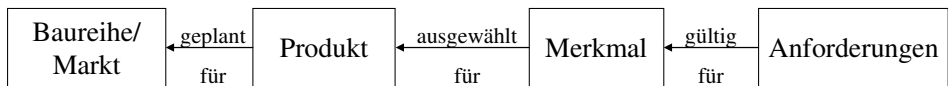


Abbildung 4: Zielzuordnung

Der Variantenmanagementansatz muss diese Trennung, sowie die Dokumentation der Zuordnungsentscheidungen unterstützen können.

### 3.7 Weitere Herausforderungen

Außer der davor diskutierten Herausforderungen müssen folgende Herausforderungen beachtet werden:

- Alle Lastenhefte bei der Daimler AG werden meistens in natürlicher Sprache spezifiziert.
- Die gleichen Anforderungen dürfen nicht mehrfach in einem Lastenheft dokumentiert werden, um die Größe von Lastenheften nicht zu steigern.
- Es muss möglich sein, innerhalb eines Lastenheftes einen Variantenvergleich durchzuführen. Also zu vergleichen, welche Anforderungen in einer Variante implementiert sind.
- Die Anforderungen, die für alle Varianten gültig sind, müssen erkennbar sein.
- Die Beziehungen zwischen einzelnen Variabilitätsmerkmalen müssen prüfbar sein, um die Konsistenz der Varianten überprüfen zu können.
- Die Zuordnung von Anforderungen zu Varianten muss automatisch überprüfbar sein, um die Qualität der Lastenhefte zu garantieren.

## 4 Ausblick

Parallel zur Analyse der existierenden Ansätze wurde eine umfangreiche Literaturrecherche durchgeführt, die in diesem Beitrag nicht vorgestellt wird. Basierend auf den Ergebnissen der Literaturrecherche und State-of-the-Practice wurden zwei Arbeitsthemen identifiziert. Das erste Thema beschäftigt sich mit dem Variantenmanagement (die Identifikation der Variabilitätsmerkmalen, Variabilitätsmodellierung und Auswahl der zulässigen Konfigurationen für die Systeme und Komponenten). Das zweite Thema beinhaltet die Zuordnung von Anforderungen zu Varianten in Lastenheften.

Für den Bereich des Variantenmanagements können Ansätze aus der Literatur (z.B. [Be03], [B01], [CE00], [Ka90], [PBL05], [Re08], [TH03]) übernommen werden. Natürlich müssen die in der Forschung ausgearbeiteten Ansätze an die Herausforderungen der Praxis (siehe z.B. Kapitel 2) angepasst werden. Für die Variabilitätsmodellierung haben wir die Merkmalsmodelle anvisiert. Die Merkmalsmodellierung hat eine große Akzeptanz in der Praxis erfahren [Li03]. Auch bei der Daimler AG wird dieser Ansatz in der Design- und Implementierungsphase der internen Softwareentwicklung verwendet.

Bei der Literaturrecherche wurde kein Ansatz gefunden, der sich mit der Zuordnung von Anforderungen zu Varianten beschäftigt. Basierend auf den identifizierten Problemen und Anforderungen der Entwickler sowie den Ergebnissen der Literaturrecherche wurde das Thema des Variabilitätsmanagements in Anforderungsdokumenten für meine Dissertation ausgewählt. Das Ziel meiner Arbeit ist es, eine Lösung für existierendes Problem der Zuordnung der Anforderungen zu Varianten zu finden.

## Literaturverzeichnis

- [Be03] M. Becker, "Towards a General Model of Variability in Product Families", Proceedings of the 1st Workshop on Software Variability Management, Groningen, Netherlands, 2003.

- [B001] J. Bosch, G. Florijn, D. Greefhorst, J. Kuusela, H. Obbink, K. Pohl, "Variability Issues in Software Product Lines", Proceedings of PFE-4, 2001, pp. 11-19.
- [Bo08] E. Boutkova, "Variantenmanagement in Anforderungsdokumenten: State of the Practice". GI Fachtreffen, Karlsruhe, 2008.
- [CE00] K. Czarnecki, U. Eisenecker, "Generative Programming". Addison-Wesley, 2000.
- [CN01] P. Clements, L. Northrop, "Software Product Lines: Practices and Patterns", SEI Series in Software Engineering. Addison-Wesley, 2001.
- [Ka90] K. Kang, S. Cohen, J. Hess, W. Novak, S. Peterson: "Feature Oriented Domain Analysis (FODA) Feasibility Study". Technical Report CMU/SEI-90-TR-021, 1990.
- [Li03] H. Lichter, T. von der Maßen, A. Nyßen, T. Weiler, "Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung", 2003.
- [PBL05] K. Pohl, G. Böckle, F. van der Linden, "Software Product Line Engineering: Foundations", Principles, and Techniques. Springer Verlag, 2005.
- [Re08] M.O. Reiser, "Managing Complex Variability in Automotive Software Product Lines with Subscoping and Conguration Links", Dissertation, TU Berlin, 2008.
- [Te03] J.C. Trigaux, P. Heymans, "Modelling Variability Requirements in Software Product Lines: A comparative survey", Technical report PLENTY project, Belgium, 2003.
- [RW05] M.O. Reiser, M. Weber, "Using Product Sets to Define Complex Product Decisions". In: H. Obbink and K. Pohl (Hrsg.): SPLC 2005, LNCS 3714, Springer-Verlag, pp. 21–32, 2005.

# Werkzeuge zur modellgetriebenen Entwicklung von Produktlinien: Ein Erfahrungsbericht am Beispiel von Versionskontrollsystemen

Thomas Buchmann, Alexander Dotor, Bernhard Westfechtel

*vorname.nachname@uni-bayreuth.de*

**Abstract:** Modellgetriebene Softwareentwicklung zielt darauf ab, den Aufwand zur Erstellung komplexer Softwaresysteme mit Hilfe von ausführbaren Modellen zu reduzieren, die auf einer höheren Abstraktionsebene liegen als Programmcode. Derzeit wird eine große Fülle von Ansätzen entwickelt, um modellgetriebene Entwicklung effizient zu unterstützen. Um die Tauglichkeit dieser Ansätze zu überprüfen, werden nichttriviale Fallstudien dringend benötigt. Dieser Aufsatz berichtet über ein Projekt, das sich die modellgetriebene Entwicklung von Versionskontrollsystemen zum Ziel setzt. Der Schwerpunkt liegt dabei auf den Werkzeugen, die zur Entwicklung wieder- verwendet bzw. neu entwickelt wurden.

## 1 Einleitung

Bei der Entwicklung komplexer Softwaresysteme spielen Modelle eine essentielle Rolle. Ein *Modell* ist eine Repräsentation des zu entwickelnden Systems, die auf einer höheren Abstraktionsebene liegt als der Programmcode. Hinsichtlich der Verwendung von Modellen im Softwareentwicklungsprozess unterscheidet man zwischen *modellbasierter Entwicklung*, bei der Modelle zur abstrakten Beschreibung des Zielsystems eingesetzt werden, und dem engeren Begriff der *modellgetriebenen Entwicklung*, die zusätzlich die Erstellung ausführbarer Modelle verlangt. Das derzeit starke Interesse an modellbasierter bzw. modellgetriebener Entwicklung wird durch eine Reihe einschlägiger Konferenzen und Workshops manifestiert.

Zunehmend komplexe Geschäftsanforderungen lassen sich nur mit verbesserten Methoden und Werkzeugen zur Softwareentwicklung bewältigen. Modellgetriebene Entwicklung soll dazu beitragen, die Produktivität der Softwareentwicklung zu steigern, indem konventionelles Programmieren durch die Erstellung ausführbarer Modelle auf einer hohen Abstraktionsebene ersetzt wird. Derzeit wird eine große Fülle von Ansätzen entwickelt, um modellgetriebene Entwicklung effizient zu unterstützen. Um die Tauglichkeit dieser Ansätze zu überprüfen, werden nichttriviale *Fallstudien* dringend benötigt.

Dieser Aufsatz berichtet über ein Projekt, das sich die *modellgetriebene Entwicklung von Versionskontrollsystemen* zum Ziel setzt. In der Softwarekonfigurationsverwaltung spielt die *Versionskontrolle* eine zentrale Rolle. Für die Versionskontrolle wurde eine Vielzahl von *Versionsmodellen* entwickelt, die in kommerziellen und frei verfügbaren Systemen so-

wie in Forschungsprototypen implementiert wurden [CW98]. Dabei wurden in der Regel die zugrunde liegenden Versionsmodelle nicht explizit definiert, sondern sind nur implizit in den implementierten Systemen vorhanden. Der Aufwand zur Programmierung eines Versionskontrollsystems ist beträchtlich. Dabei wird in der Regel jedes System von Grund auf neu implementiert, auch wenn bereits ein System mit ähnlichem Versionsmodell und ähnlicher Funktionalität existiert. Beispielsweise wurde das Versionskontrollsystem Subversion [CSFP04] von Grund auf neu implementiert, obwohl es starke Ähnlichkeiten zu seinem Vorgänger CVS [Ves06] aufweist.

Vor diesem Hintergrund verfolgt das *MOD2-SCM-Projekt* (*Model-Driven and Modular Software Configuration Management Systems* [BDW09]) folgende Ziele:

- Durch einen *modularen Ansatz* soll es ermöglicht werden, Versionskontrollsysteme aus wiederverwendbaren Bausteinen zusammenzusetzen. Damit wird die Voraussetzung für eine *Produktlinie* für Versionskontrollsysteme geschaffen.
- Durch *modellgetriebene Entwicklung* soll zum einen erreicht werden, dass Versionsmodelle explizit definiert werden, und zum anderen die Produktivität gesteigert werden, indem konventionelle Programmierung durch die Erstellung ausführbarer Modelle auf hoher Abstraktionsebene ersetzt wird.

Das MOD2-SCM-Projekt stellt eine nichttriviale *Fallstudie* für die Anwendung modellgetriebener Entwicklung dar. In [BDW09] haben wir einen Überblick über unseren Modellierungsansatz gegeben und dessen Vorteile und Beschränkungen diskutiert. Der vorliegende Aufsatz konzentriert sich dagegen komplementär dazu auf einen Überblick über die *Werkzeuge* zur modellgetriebenen Entwicklung. Soweit möglich, haben wir bereits existierende Werkzeuge wiederverwendet. Die Entwicklung neuer Werkzeuge war nicht das primäre Ziel des Projekts, sondern die Anwendung von Modellierungssprachen und Werkzeugen zur Lösung eines Anwendungsproblems. Im Laufe des Projekts stellte sich jedoch heraus, dass die Werkzeugunterstützung signifikante Lücken aufwies, die durch Eigenentwicklungen geschlossen werden mussten.

## 2 Überblick über den Modellierungsansatz

Wie in Softwareproduktlinien üblich, unterscheiden wir die beiden Ebenen Domänenentwicklung und Anwendungsentwicklung:

**Domänenentwicklung** Zunächst wird die Domäne analysiert und das Ergebnis der Analyse in einem *Featuremodell* [CCH<sup>+</sup>90] dokumentiert. Das Featuremodell beschreibt, durch welche Eigenschaften sich Versionskontrollsysteme charakterisieren lassen, welche alternativen Ausprägungen diese Eigenschaften haben können und wie die Eigenschaften zueinander in Beziehung stehen. Ausgehend vom Featuremodell wird anschließend ein *Domänenmodell* entwickelt, in dem die Features des Featuremodells realisiert sind.

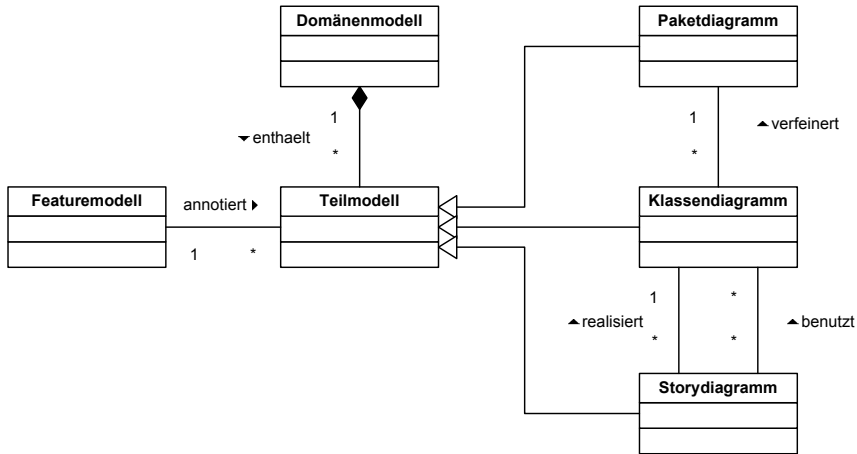


Abbildung 1: Modelle und deren Beziehungen

**Anwendungsentwicklung** In unserem Ansatz wird die Anwendungsentwicklung zu einem *Konfigurierungsprozess* reduziert. Im ersten Schritt wird das Featuremodell konfiguriert, indem die zu unterstützenden Eigenschaften des gewünschten Versionskontrollsystems spezifiziert werden. Die dadurch erstellte *Featurekonfiguration* wird anschließend benutzt, um das ausführbare Domänenmodell zu konfigurieren. Aus dem *konfigurierten Domänenmodell* wird schließlich ausführbarer Code erzeugt. In diesem Konfigurierungsprozess muss lediglich der erste Schritt interaktiv ausgeführt werden, die weiteren Schritte sind vollständig automatisiert.

Abbildung 1 vermittelt einen Überblick über die zur Modellierung verwendeten Modelltypen und die zwischen ihnen bestehenden grobgranularen Beziehungen. Zur Definition der Eigenschaften von Versionskontrollsystemen wird ein *globales Featuremodell* erstellt. Alle Teilmodelle des Domänenmodells nehmen auf dieses Modell Bezug. Zum *Modellieren im Großen* verwenden wir *Paketdiagramme*. Jedes Paket wird durch ein *Klassendiagramm* verfeinert. Das Verhaltensmodell wird schließlich durch *Storydiagramme* beschrieben [Zün01]. Jedes Storydiagramm realisiert das Verhalten einer Methode einer Klasse. Ein Storydiagramm ähnelt einem UML2-Interaktionsübersichtsdiagramm und besteht aus einem Aktivitätsdiagramm, dessen Knoten durch Storymuster (ähnlich zu Kommunikationsdiagrammen) verfeinert werden. Da Storydiagramme in ausführbaren Java-Code übersetzt werden, ist das konfigurierte Domänenmodell ausführbar.

Abbildung 2 zeigt beispielhaft einige Ausschnitte aus dem Modell, das im Rahmen von MOD2-SCM entstanden ist. Im Folgenden gehen wir nur kurz auf dieses Modell ein; für weitere Details sei auf [BDW09] verwiesen. Die linke Seite der Abbildung zeigt einen Ausschnitt aus einer Konfiguration eines Featuremodells, das die Eigenschaften der mit der Produktlinie erzeugbaren Versionskontrollsysteme beschreibt. Das der Konfiguration zugrunde liegende Featuremodell definiert z.B. Varianten für die Entwicklungsgeschichte (Feature History) und für die Abspeicherung (Feature Storage). Rechts oben ist ein Ausschnitt aus dem Paketdiagramm dargestellt, das die Modellstruktur im Großen definiert. Pakete, die ein spezifisches Feature des Featuremodells realisieren, werden mit Features annotiert (so wird beispielsweise dem Paket Directed Deltas das gleichnamige Feature zu-



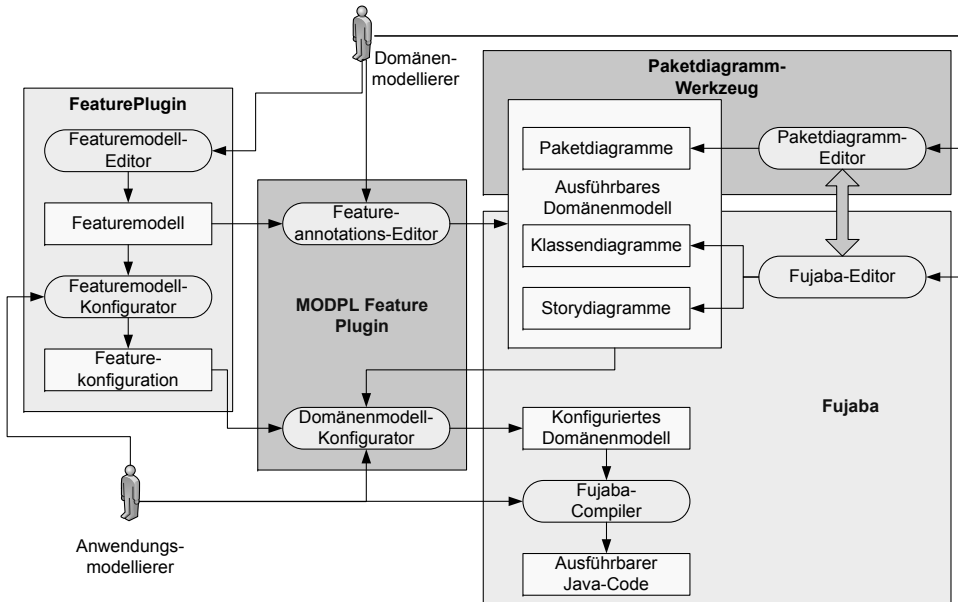


Abbildung 3: Modellierungswerkzeuge

linie die Entwicklung von Versionskontrollsystemen signifikant erleichtert werden kann. Es war nicht das primäre Ziel des Projekts, neue Werkzeuge (und ggf. neue Modellierungssprachen) zu entwerfen und zu realisieren. Daher wurden, soweit dies möglich war, existierende Werkzeuge wiederverwendet.

Im Laufe des Projekts zeigte sich jedoch, dass die zur Verfügung stehende Werkzeugunterstützung gravierende Lücken aufwies, die sich nur durch Eigenentwicklung schließen ließen. Insgesamt ist somit eine *Modellierungsumgebung* entstanden, die sich teilweise aus bereits existierenden und teilweise aus selbstentwickelten Werkzeugen zusammensetzt (siehe Abbildung 3, in der die selbstentwickelten Werkzeuge grau hinterlegt sind). Featuremodelle werden mit dem *FeaturePlugin* erstellt und konfiguriert, und das Domänenmodell wird mit *Fujaba* erstellt. Die von uns entwickelten Werkzeuge dienen dazu, diese Werkzeuge zu integrieren — dies ist die Aufgabe des *MODPL Feature Plugin* — bzw. bei der Erstellung des Domänenmodells das Modellieren im Großen zu unterstützen — dies leistet das *Paketdiagramm-Werkzeug*. In den folgenden Unterabschnitten werden die Werkzeuge der Modellierungsumgebung beschrieben. Dabei liegt der Schwerpunkt auf den selbstentwickelten Werkzeugen.

### 3.1 Featuremodellierung und -konfigurierung

*Featuremodelle* wurden ursprünglich in [CCH<sup>+</sup>90] eingeführt. Aus dem ursprünglichen FODA-Ansatz hat sich mittlerweile eine Reihe von Varianten entwickelt, s. z.B. [BHST04]. In unseren Arbeiten benutzen wir eine dieser Varianten, die im Werkzeug *FeaturePlugin* [AC04] implementiert ist.



Der linke Teil von Abbildung 2 ist ein Ausschnitt aus einem Bildschirm-Schnappschuss dieses Werkzeugs. Er zeigt eine *Featurekonfiguration*, die durch Auswahl von Features aus dem Featuremodell für MOD2-SCM entsteht. Die Featurekonfiguration besteht aus den mit Häkchen gekennzeichneten Features und ihren übergeordneten Features im Featuremodell. Die durch Kreuze markierten Features sind Bestandteil des Featuremodells, wurden jedoch in der dargestellten Featurekonfiguration ausgeschlossen. Die Featurekonfiguration repräsentiert ein CVS-ähnliches System: Für die Versionshistorie wurde eine zweistufige Versionsgeschichte mit expliziten Zweigen ausgewählt. Als Produktmodell wird das Dateisystem festgelegt. Zur Speicherung werden gemischte Deltas verwendet (Rückwärtsdeltas auf dem Hauptzweig, Vorwärtsdeltas auf den Seitenzweigen). Die Synchronisation erfolgt optimistisch. Versionen werden lokal identifiziert, d.h. jedes versionierte Objekt vergibt seine Versionsbezeichner unabhängig von anderen versionierten Objekten. "Tagging" bedeutet, dass Versionen durch symbolische Namen markiert werden können.

FeaturePlugin unterstützt die Erstellung von Featuremodellen und deren Konfigurierung, ohne dass jedoch Beziehungen zum Domänenmodell hergestellt werden. Diese Aufgabe übernimmt das von uns entwickelte MODPL Feature Plugin (Abschnitt 3.4).

## 3.2 Fujaba

Im Zentrum der Werkzeugunterstützung steht *Fujaba* [Zün01], die mit Abstand größte Komponente der Modellierungsumgebung. Fujaba unterstützt die Erstellung von *Klassen- und Storydiagrammen*, aus denen der Fujaba-Compiler ausführbaren Code generiert. Dabei stellen die Storydiagramme und die Erzeugung von ausführbarem Code einen beträchtlichen Mehrwert gegenüber anderen Modellierungswerkzeugen dar. Unsere Wahl fiel auf Fujaba, weil die zugrunde liegende Technologie über einen langen Zeitraum gereift ist.

Dennoch waren mit Fujaba allein (und dem oben beschriebenen FeaturePlugin) die Herausforderungen des MOD2-SCM-Projekts nicht zu bewältigen. Zum einen wird das Modellieren im Großen durch Fujaba nicht unterstützt, zum anderen bietet Fujaba keine spezifische Unterstützung von Produktlinien.

## 3.3 Paketdiagramm-Werkzeug

Ein Feature-Modell, wie es auf der linken Seite in Abbildung 2 dargestellt wurde, suggeriert, dass das zu entwickelnde System einfach ist. Dies ist aber keineswegs der Fall: Es ist nämlich insbesondere zu klären, wie das Featuremodell auf ein Domänenmodell so abgebildet werden kann, dass im Featuremodell als unabhängig deklarierte Features im Domänenmodell tatsächlich voneinander entkoppelt sind. Dies stellt große Anforderungen an die *Architekturmodellierung*.

Im MOD2-SCM-Projekt zeigte sich schon früh, dass das Domänenmodell mit den von Fujaba angebotenen Werkzeugen allein nicht strukturiert werden kann [BDW08]. Bereits

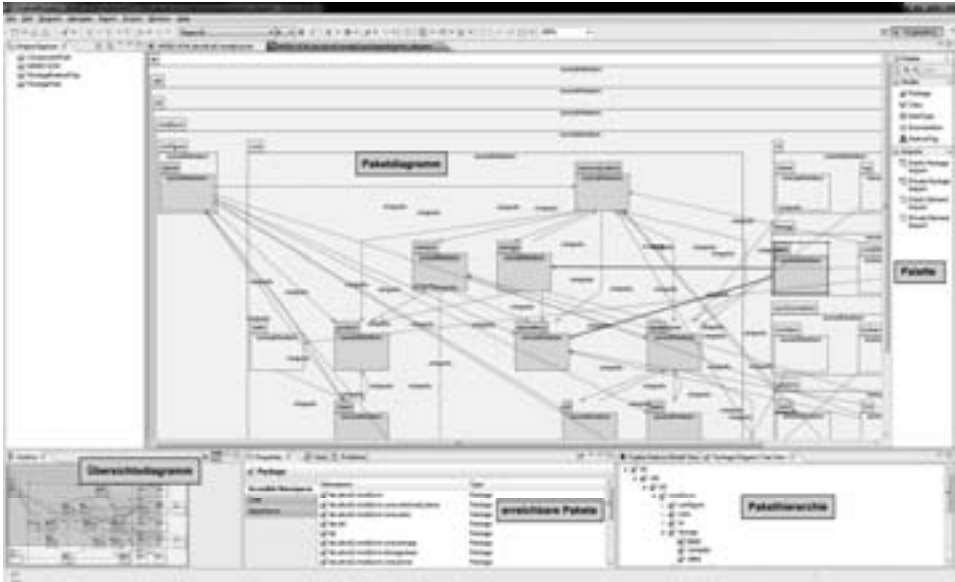


Abbildung 4: Paketdiagramm-Editor

jetzt ist das Domänenmodell so groß, dass man sehr schnell den Überblick über die zahlreichen Klassen- und Storydiagramme verliert. Insbesondere beim Bau einer Produktlinie muss man darauf achten, die Abhängigkeiten zwischen den Modellkomponenten präzise zu überwachen; ansonsten entstehen bei der Konfigurierung des Domänenmodells sehr leicht unbrauchbare Resultate.

Deshalb haben wir im Zuge des MOD2-SCM-Projekts ein *Paketdiagramm-Werkzeug* entwickelt, dessen technische Realisierung in [BDK09] dargestellt ist.

Den Kern des Paketdiagramm-Werkzeugs bildet ein *Paketdiagramm-Editor*, der die Erstellung von UML2-Paketdiagrammen unterstützt. Abbildung 4 zeigt einen Bildschirmabzug dieses Editors, der mit Hilfe von GMF [EF09] realisiert wurde. Während der Bildschirmabzug nur Pakete als Elemente des Diagramms zeigt, unterstützt der Editor auch das Einfügen von Klassen und Interfaces in ein Paket. Abgesehen von der hierarchischen Schachtelung, lassen sich mit dem Editor sowohl private als auch öffentliche (transitiv wirkende) Importe eintragen — und zwar sowohl Paket- als auch Elementimporte.

Der Paketdiagramm-Editor kann auch unabhängig von Fujaba genutzt werden. Die zweite, im Kontext des MOD2-SCM-Projekts wichtige Komponente des Paketdiagramm-Werkzeugs ist der *Integrator*, der in Abbildung 3 durch einen Doppelpfeil repräsentiert wird. Im Einzelnen bietet der Integrator folgende Funktionalitäten an:

**Forward Engineering** Aus einem Paketdiagramm wird das Gerüst eines Fujaba-Modells erzeugt. Man beachte, dass das Fujaba zugrunde liegende Metamodell die Zuordnung von Klassen und Interfaces zu Paketen zwar unterstützt, jedoch kein Fujaba-Werkzeug zur Erstellung und Visualisierung von hierarchischen Paketdiagrammen zur Verfügung steht.

**Reverse Engineering** Ein Paketdiagramm wird aus einem zuvor entwickelten Fujaba-Modell erstellt. Dazu wählt der Modellierer den gewünschten Importtyp aus (öffentliche oder private bzw. Element- oder Paketimporte).

**Inkrementelles Roundtrip-Engineering** Neben den batchartigen Prozessen des Forward und Reverse Engineering wird auch inkrementelles Roundtrip-Engineering unterstützt. Hierbei werden sowohl Änderungen am Paketdiagramm als auch Änderungen am Fujaba-Modell in die jeweilige Richtung propagiert.

**Konsistenzüberprüfungen im Fujaba-Editor** Wird ein Modellelement in einem Klassen- oder Storydiagramm benutzt, so wird überprüft, ob das benutzte Element gemäß dem Paketdiagramm sichtbar ist. Ist dies nicht der Fall, so erscheint ein Dialog, in dem der Modellierer das Kommando entweder abbrechen kann oder den Typ des einzufügenden Imports auswählen kann. Diese Kontrollfunktion ist von essenzieller Bedeutung, um die Einhaltung von Architekturvorgaben überwachen zu können und einer Architekturerosion Einhalt zu gebieten.

### 3.4 Verbindung von Feature- und Domänenmodell

Die zweite große Lücke in der Werkzeugunterstützung betraf die *Verbindung von Feature- und Domänenmodell*. Diese Lücke wird durch das *MODPL Feature Plugin* geschlossen, dessen technische Realisierung in [BD09b] beschrieben wird. Wie beim Paketdiagramm-Werkzeug kommt es uns im Rahmen des vorliegenden Papiers jedoch auf die zugrunde liegenden Konzepte und die Einbettung in die Gesamtumgebung an.

Das MODPL Feature Plugin besteht aus zwei Teilen: einem Editor und einem Konfigurator. Abbildung 5 zeigt einen Bildschirm-Schnappschuss des *Editors*. Genauer sollte man hier von einer Erweiterung des Fujaba-Editors sprechen, die die Annotation von Modellelementen mit Features unterstützt. Es sei darauf hingewiesen, dass eine analoge Erweiterung auch in den Paketdiagramm-Editor eingebaut wurde.

Um das Plugin im Editor benutzen zu können, muss zunächst ein Featuremodell geladen werden. Dies geschieht mit Hilfe eines Exports aus dem Werkzeug *FeaturePlugin*, das in Abschnitt 3.1 erläutert wurde. Das untere Fenster in Abbildung 5 zeigt das Ergebnis dieser Transformation in einer Baumsicht an. Nach dem Laden des Feature-Modells können Modellelemente mit Features dekoriert werden. Dies geschieht in Fujaba mit Hilfe von *Annotationen*<sup>1</sup>, die einer spezifischen Syntax genügen. Falls einem Modellelement mehrere Features zugeordnet werden, so gelten diese als konjunktiv verknüpft: Ein Modellelement  $m$  mit Features  $f_1, \dots, f_n$  ist nur Teil des konfigurierten Modells, wenn die entsprechende Feature-Konfiguration alle Features  $f_1, \dots, f_n$  enthält.

Der Editor bietet eine Reihe von Funktionalitäten für die *Konsistenzkontrolle zwischen Feature- und Domänenmodell* an:

---

<sup>1</sup>Für die Darstellung von Annotationen wurde von den Fujaba-Entwicklern das graphische Element für Kommentare verwendet. Kommentare und Annotationen sind aber unterschiedliche Klassen des Fujaba-Metamodells.



Hängt ein Modellelement  $e_1$  von einem Modellelement  $e_2$  ab, dann darf  $e_1$  nur selektiert werden (Teil eines konfigurierten Modells sein), wenn  $e_2$  ebenfalls selektiert wird.

Beispielsweise darf eine Assoziation nur selektiert werden, wenn auch die Klassen an den Assoziationsenden selektiert werden. Diese Regel lässt sich wie folgt auf Feature-Annotationen übersetzen (man beachte, dass mehrere einem Element zugeordneten Features konjunktiv verknüpft sind, s.o.):

Hängt  $e_1$  von  $e_2$  ab und ist  $e_2$  mit  $f$  annotiert, dann muss auch  $e_1$  mit  $f$  annotiert werden.

MODPL Feature Plugin überwacht diese Regel und bietet Reparaturaktionen an, um Feature-Annotationen automatisch zu propagieren.

Die zweite Komponente von MODPL Feature Plugin ist der *Konfigurator* für das Domänenmodell. Um ihn benutzen zu können, muss man zuvor eine Feature-Konfiguration laden. Der Domänenmodell-Konfigurator ist als Präprozessor des Fujaba-Compilers realisiert. Er übergibt dem Fujaba-Compiler nur Modellelemente zur Codeerzeugung, die zur spezifizierten Konfiguration gehören (dazu zählen stets die nicht annotierten Modellelemente). Dabei überprüft der Konfigurator die oben angegebene Regel für Annotationen voneinander abhängiger Modellelemente und stellt deren Einhaltung durch Propagierung von Features sicher, um zu vermeiden, dass inkonsistenter Code generiert wird.

## 4 Vergleich mit anderen Ansätzen

In der Domäne der Versionskontrollsysteme sind nur wenige auf Produktlinien basierende Ansätze entwickelt worden. In [vdLvdH03] wurde ein komponentenbasierter Ansatz beschrieben, der wie MOD2-SCM darauf abzielt, ein Versionskontrollsystem aus austauschbaren Komponenten mit wohldefinierten Schnittstellen zusammzusetzen. Sowohl in [Kov05] als auch in [WGP04] lassen sich Versionsmodelle strukturell definieren (durch Klassendiagramme bzw. ER-ähnliche Diagramme); aus einem strukturellen Modell lässt sich dann ein anwendungsspezifisches Versionskontrollsystem erzeugen. Nur das MOD2-SCM-Projekt verfolgt jedoch einen modularen Produktlinienansatz, der auf einem ausführbaren Domänenmodell basiert.

Für die Produktlinienentwicklung wurden unterschiedliche Prozessmodelle vorgeschlagen. In [PBvdL05] werden unterschiedliche Ebenen für die Domänen- und Anwendungsentwicklung eingeführt, die durch ein einheitliches Lebenszyklusmodell strukturiert werden. In MOD2-SCM wird dagegen der Aufwand in der Anwendungsentwicklung drastisch reduziert, da hier lediglich eine Konfiguration des Domänenmodells durchzuführen ist. Die Konfiguration erfolgt durch Auswahl der gewünschten Eigenschaften im Featuremodell und einer Codeerzeugung auf dieser Basis. Auf einem ähnlichen Ansatz basieren beispielsweise auch die kommerziellen Systeme BigLever Software Gears [Kru08] und pure::variants [Beu08].

Im Produktlinienkontext bietet der hier vorgestellte Ansatz im Vergleich mit dem domänen-spezifischen Ansatz [JHS<sup>+</sup>09], [Voe08] eine erhöhte Flexibilität. Mit Hilfe von Fujaba lassen sich domänenspezifische Sprachen definieren. Mit Hilfe unserer Erweiterungen lassen sich darüber hinaus Familien von domänenspezifischen Sprachen unterstützen (Variabilität auf zwei Ebenen).

Das MODPLFeaturePlugin zeichnet sich insbesondere durch eine Reihe von Konsistenzanalysen für die Feature-Annotationen von Modellelementen aus. Der z.B. in [HKW08] dargestellte FeatureMapper verfolgt ähnliche Ziele in einem anderen Modellierungskontext (EMF-Modelle), unterstützt jedoch nicht die in [BD09a] beschriebenen Analysen und Korrekturen, die die Konsistenz des konfigurierten Domänenmodells (und des daraus generierten Codes) mit dem Metamodell sicherstellen sollen. Eine direkte Nutzung (und Erweiterung) des FeatureMapper war im Kontext des MOD2-SCM-Projekts nicht möglich, da Fujaba nicht auf EMF-Modellen basiert.

Das Paketdiagramm-Werkzeug unterstützt das Modellieren im Großen, das in fast allen Modellierungswerkzeugen nur unzureichend Beachtung findet. In vielen Fällen werden Paketdiagramme nicht angeboten, sondern es ist lediglich möglich, ein Modell hierarchisch in Pakete zu zerlegen. In einigen Fällen (z.B. Rational Rose) lassen sich zwar Paketdiagramme erstellen, aber es wird nicht überprüft, ob das erstellte Modell die im Paketdiagramm definierten Sichtbarkeiten beachtet. Lediglich MOFLON [AKRS06] bietet ein ähnliches Paketdiagramm-Werkzeug an. Allerdings erlaubt es auf Ebene der Paketdiagramme nur das Anlegen von Paketimporten. Elementimporte hingegen werden implizit erzeugt, sobald auf nicht sichtbare Elemente zugegriffen wird. Diese Elementimporte werden aber nicht im Paketdiagramm dargestellt. Reverse Engineering von Paketdiagrammen wird von MOFLON nicht angeboten.

## 5 Zusammenfassung

Wir haben das MOD2-SCM-Projekt vorgestellt, das sich mit der modellgetriebenen Entwicklung einer Produktlinie für Versionskontrollsysteme befasst. Im Rahmen dieses Projekts haben wir teilweise von bereits existierenden Werkzeugen Gebrauch gemacht und diese um selbstentwickelte Werkzeuge ergänzt. Insgesamt steht nun eine integrierte Umgebung für die modellgetriebene Entwicklung von Produktlinien zur Verfügung, die sich auch für andere Anwendungsdomänen einsetzen lässt. Weiteren Handlungsbedarf sehen wir vor insbesondere in folgenden Bereichen:

**Modellieren im Großen** Mit den heute verfügbaren Sprachen führt der Einsatz modellgetriebener Entwicklung nicht zu kleinen Modellen (was man ja eigentlich hoffen würde). Bisher haben wir Paketdiagramme für das Modellieren im Großen verwendet. Diese haben sich bei der Strukturierung von Modellen durchaus als hilfreich erwiesen. Dennoch werden wir den Einsatz bzw. die Entwicklung einer ausdrucksstärkeren Modellierungssprache prüfen, mit der sich die Schnittstellen besser ausdrücken bzw. Abhängigkeiten wirksamer kontrollieren lassen. Ein geeigneter Kandidat für diesen Zweck scheinen UML Komponentendiagramme zu sein, erlau-

ben sie doch, modulare Teile eines Systems zu kapseln und wohldefinierte Schnittstellen nach außen zu definieren.

**Prozessunterstützung** Da sämtliche Werkzeuge, die in unserem modellgetriebenen Entwicklungsprozess für Produktlinien verwendet werden, als Plugin für Eclipse verfügbar sind, bietet es sich an, ein weiteres Werkzeug auf Eclipse-Basis zu realisieren, mit dessen Hilfe der gesamte Prozess geführt werden kann. Offene bzw. fehlende Schritte im Entwicklungsprozess können somit entsprechend visualisiert werden. Ebenfalls wird eine sinnvolle Reihenfolge der einzelnen Entwicklungsschritte vorgegeben und ein Einstieg in die modellgetriebene Entwicklung von Produktlinien somit erheblich erleichtert.

## Literatur

- [AC04] Michal Antkiewicz und Krzysztof Czarnecki. FeaturePlugin: Feature Modeling Plugin for Eclipse. In *OOPSLA Workshop on Eclipse Technology eXchange (ETX 2004)*, Seiten 67–72, British Columbia, 2004.
- [AKRS06] Carsten Amelunxen, Alexander Königs, Tobias Röttschke und Andy Schürr. MOFLON: A Standard-Compliant Metamodeling Framework with Graph Transformations. In *MDA - Foundations and Applications*, LNCS 4066, Seiten 361–375, Genua, 2006.
- [BD09a] Thomas Buchmann und Alexander Dotor. Constraints for a Fine-Grained Mapping of Feature Models and Executable Domain Models. In *1st International Workshop on Model-Driven Product Line Engineering (MDPLE'09)*, Seiten 9–17, Twente, 2009.
- [BD09b] Thomas Buchmann und Alexander Dotor. Mapping Features to Domain Models in Fujaba. In *7th International Fujaba Days*, Seiten 20–24, Eindhoven, 2009.
- [BDK09] Thomas Buchmann, Alexander Dotor und Martin Klinke. Supporting Modeling in the Large in Fujaba. In *7th International Fujaba Days*, Seiten 59–63, Eindhoven, 2009.
- [BDW08] Thomas Buchmann, Alexander Dotor und Bernhard Westfechtel. Experiences with Modeling in the Large in Fujaba. In *6th International Fujaba Days*, Seiten 5–9, Dresden, 2008.
- [BDW09] Thomas Buchmann, Alexander Dotor und Bernhard Westfechtel. Model-Driven Development of Software Configuration Management Systems — A Case Study in Model-Driven Engineering. In *4th International Conference on Software and Data Technologies*, Seiten 309–316, Sofia, 2009.
- [Beu08] Danilo Beuche. Modeling and Building Software Product Lines with pure::variants. In *12th International Software Product Line Conference*, Seite 358, Limerick, 2008.
- [BHST04] Yves Bontemps, Patrick Heymans, Pierre-Yves Schobbens und Jean-Christophe Trigaux. Semantics of Feature Diagrams. In *Workshop on Software Variability Management for Product Derivation (Towards Tool Support)*, Boston, 2004.
- [CCH<sup>+</sup>90] Kyo C. Chang, Sholom G. Cohen, James A. Hess, William E. Novak und A. Spencer Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Bericht CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1990.

- [CSFP04] Ben Collins-Sussman, Brian W. Fitzpatrick und C. Michael Pilato. *Version Control with Subversion*. O'Reilly & Associates, Sebastopol, 2004.
- [CW98] Reidar Conradi und Bernhard Westfechtel. Version Models for Software Configuration Management. *ACM Computing Surveys*, 30(2):232–282, 1998.
- [EF09] Eclipse-Foundation. *Graphical Modeling Framework*. Ottawa, 2009. <http://www.eclipse.org/modeling/gmf/>.
- [HKW08] Florian Heidenreich, Jan Kopcsek und Christian Wende. FeatureMapper: Mapping Features to Models. In *30th International Conference on Software Engineering*, Seiten 943–944, Leipzig, 2008.
- [JHS<sup>+</sup>09] JulesWhite, James H.Hill, SumantTambe, Aniruddha S.Gokhale, Douglas C.Schmidt und JeffGray. Improving Domain-Specific Language Reuse with Software Product Line Techniques. *IEEE Software*, 26(4):47–53, 2009.
- [Kov05] Jernej Kovše. *Model-Driven Development of Versioning Systems*. Dissertation, Universität Kaiserslautern, 2005.
- [Kru08] Charles W. Krueger. The BigLever Software Gears Unified Software Product Line Engineering Framework. In *12th International Software Product Line Conference*, Seite 353, Limerick, 2008.
- [PBvdL05] Klaus Pohl, Günther Böckle und Frank van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, Berlin, 2005.
- [vdLvdH03] Ronald van der Lingen und André van der Hoek. Dissecting Configuration Management Policies. In *Software Configuration Management: ICSE Workshops SCM 2001 and SCM 2003*, LNCS 2649, Seiten 177–190, Portland, 2003.
- [Ves06] Jennifer Vesperman. *Essential CVS*. O'Reilly & Associates, Sebastopol, California, 2006.
- [Voe08] Markus Voelter. A Family of Languages for Architecture Description. In *OOPSLA Workshop on Domain-Specific Modeling*, Seiten 86–93, 2008.
- [WGP04] E. James Whitehead, Guozheng Ge und Kai Pan. Automatic generation of hypertext system repositories: a model driven approach. In *15th ACM Conference on Hypertext and Hypermedia*, Seiten 205–214, Santa Cruz, 2004.
- [Zün01] Albert Zündorf. Rigorous Object Oriented Software Development. Bericht, Universität Paderborn, 2001.





# Projectional Language Workbenches as a Foundation for Product Line Engineering

Markus Voelter

independent/itemis  
Oetztaler Strasse 38  
70327 Stuttgart  
voelter@acm.org

**Abstract:** In this paper I explain the benefits of projectional language workbenches for product line engineering. The ability to extend programming languages with domain specific concepts, mix programs (i.e. descriptions written in general purpose languages) and models (i.e. descriptions expressed with DSLs) and also overlay configurative variability to both of these promises highly integrated and productive tools for product line development.

## 1 Variability in PLE

In the context of product line engineering, DSLs are used to bind variability. We distinguish between two kinds of variability: structural and non-structural. Structural variability is described using creative construction DSLs, whereas non-structural variability can be described using configuration tools. Structural variability is unbounded, whereas configuration variability is usually bounded. Figure 1 illustrates the spectrum of languages commonly used for expressing and binding variability, reusing an illustration by Krzysztof Czarnecki [6].

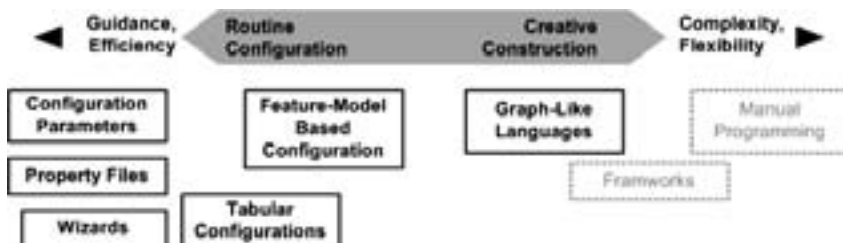


Figure 1 Spectrum of Variability

Let's look at two well-known ways of representing variability. Figure 2 shows the meta model of a creative construction DSL. Any number of models can be defined, by instantiating meta model elements. Figure 3 presents an example using the familiar concrete syntax of UML.

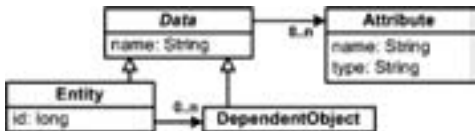


Figure 2 A simple meta model for a creative construction language

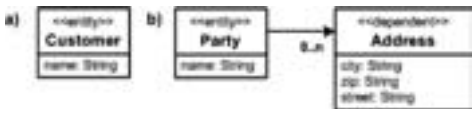


Figure 3 Two example models

Figure 4 shows a feature model of a weather station using the well-known feature modeling notation [6]. The feature model expresses a certain configuration space, i.e. the model is an expression of configurative variability. A specific configuration is described by selecting a valid subset of those features.

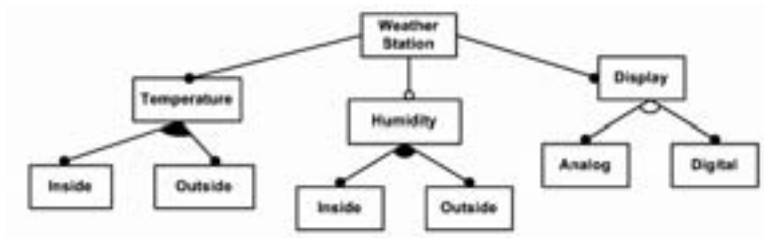


Figure 4 Weather station feature model

Product line development often involves both kinds of variability although there is a bias towards configurative variability in mature product lines. In practice, however, product line development tools need to be able to represent both kinds of variability.

More specifically, it should be possible to define arbitrary modeling languages to express product domain specific abstractions. On top of these, it should be possible to express configurative variations of these domain specific descriptions. In this way configurative and structural variability can be combined exploiting the advantages of each.

## 2 Projectional Language Workbenches

Language workbenches are tools that allow the creation, extension and integration of separate (general purpose and domain specific) languages in a common editing and validation environment. The most important characteristic of *projectional* language workbenches [1] is that all text, symbols, and graphics are projected. Projection means that programs are stored as an abstract data structure and users edit a projection, a visual or graphical representation of the data structure. Users never edit text that is subsequently parsed, rather, users always edit a tree directly. Consequently, editing things that look textual and editing things that look graphical happens based on the same technological approach, making the various notational paradigms integratable. Figure 5 shows an example of what's possible with projectional editing based on a Pension workbench developed with Intentional's Domain Workbench [2]. It showcases several different languages, all of them symbolically integrated.

Figure 5 Screenshots from a pension system built with the Intentional Domain Workbench

### 3 Handling Structural Variability

The core to handling structural variability is the ability to define new, domain specific languages to express aspects of a product in a product line or to extend existing language with new, domain specific concepts. During domain engineering, the new languages are defined and integrated with other existing languages. In application engineering developers use these languages to efficiently express products. Figure 5 shows this approach in the domain of pension contracts. Using suitable language workbenches, this kind of "language mixing" is straightforward to do - this is what language workbenches are built for. Here are some advantages of the approach, relating to handling structural variability in product line engineering.

As you can see in Figure 5, this system has several sublanguages (mathematical, rule script, tabular). All of these various syntax forms are integrated in one workbench, symbols defined with one language can refer to symbols defined with other languages.

Because no parsing is required, language composition is easily possible (the problems of grammar combination don't exit). Separate languages or language modules can be used together only when necessary. A user only sees those language features that are relevant to his tasks. Language adaptation and extension can be incremental and iterative.

Notations are more flexible than pure ASCII or Unicode. Graphical, semi-graphical and textual notations can be mixed and combined. In our case above, mathematical symbols, rule languages and spreadsheets are mixed.

Because the model can be projected in different ways, it is possible to store "out of band" information in the model (pointers to configuration features, traces back to requirements). The information can be stored in the model and only projected (i.e. "shown in the code") when the programmer specifically wants to see this information. More generally, the same model can be shown in different (textual and graphical) ways.

The nature of projectional editing means that language extension goes along with IDE extension. So, for the language concepts mentioned above the IDE will provide code completion, error checking and syntax highlighting.

Note also that by representing general purpose programming languages in language workbenches, domain-specific descriptions ("models") and things expressed with general purpose programming languages ("code") can be handled the same way. This makes mixing the two trivial and removes the (arbitrary) distinction between programming and modeling. To illustrate this, Figure 6 shows a version of Java that has been extended to include state machines.

```

TrafficLight/TrafficLight > class TrafficLight extends <name> implements <name> {
<static fields>

<static initialise>
<event go ( ) >
<event stop ( ) >
<state stop > {
  call this.setRed(true);
  go --> transitioning;
  call this.setRed(false);
}
<state transitioning > {
  call this.setYellow(true);
  after ( ) --> go;
  call this.setYellow(false);
}
<state go > {
  call this.setGreen(true);
  stop --> stop;
  call this.setGreen(false);
}
private boolean green;
private boolean red;
private boolean yellow;
<dependencies>
<initialise>
public TrafficLight() {
  <no statements>
}

protected void setRed(boolean b) {
  this.red = b;
}

protected void setGreen(boolean b) {
  this.green = b;
}

```

Figure 6 Java extended with State Machines

## 4 Handling Configurative Variability

As mentioned in the previous section it is possible with projectional editors to store information in the model that are out of band with the actual model or program. This is extremely useful for overlaying configurative variability. The following screenshots show an example with MPS [3] that supports configurative variability on top of the statemachine-Java. The screenshot on the left shows a “normal” program (a Java class with an embedded state machine). The middle screenshot shows the code annotated with feature expressions that determine under which conditions the respective elements are present in the system - it represents a product line asset. The screenshot on the right shows the pedestrian variant of the traffic lights: all the non-pedestrian code has been removed.



Figure 7 Configurative Variability for models and code

Note that all three representations are "live", i.e. changes can be made in the programs in each of the projections.

In this prototype the features are simply listed in the model (not shown). Subsequent versions of the feature-annotation facility will connect with a feature modeling to integrate the variable models/code with the overall variability management approach. Adding this functionality is straightforward to do by adding a plugin to MPS that interacts with pure-variants' XML files.

## 5 Practical Experience

Projectional editing is relatively new. Not too much practical experience exists. Here are some cases. Intentional Software has built several systems using this approach, although to my knowledge none of them uses the overlaid configurative variability. Well publicized is the Pension Workbench illustrated above [4]. JetBrains, developers of MPS, have used the tool to develop their new YouTrack issue management system [5].

Personally, I am currently working on adapting this approach for embedded system development by adding C to MPS and extending it to make it more suitable for embedded systems. Together with Bernhard Merkle (Sick AG), Alexander Bernauer (ETH Zuerich) and Jochen Hiller, I am working on a proof of concept based on C, Osek and Lego Mindstorms.

The following is a screenshot of a Hello World program written with the C implementation in MPS. As you can see, a reasonable subset of C is already available. A generator that creates C code (in text files) as an input to the compiler is available. The program shown below can be compiled and run.

```

Resource testMethodCasil projectionLevel all, outline, architecture
①
<< ... >>
②

module test imports [useHelper] util { ③

    struct intHolder {
        int a;
    }

    test testAdding / try to add two numbers { ④
        int v1 = 10;
        int i = 10;
        intHolder h;
        h.a = 10;
        assert 30 == add( v1, h.a ) => adding failed
    }

    int add( int p1, int p2 ) {
        [useHelper] return p1 + p2;
        [useHelper] return addHelper( p1, p2 ); ⑤
    }

    task startup scheduled onceUponStartup { ⑥
    ⑦ log this is my first program :-)
        run test testAdding;
    }

}

[useHelper] module util imports -[imports]- {

    exported int addHelper( int a, int b ) {
        return a + b;
    }

}

```

Figure 8 C in MPS

Let us look at some specifics of the code:

- ① Resources are the concept corresponding to files. Resources contain modules.
- ② There are different presentations modes available, the outline view is shown in the illustration below on the left.
- ③ Modules act as namespace and scope. They can depend on other modules.
- ④ A simple testing language is available (this is a separate language module that can be optionally included in programs).



- ⑤ For expressing feature variability, sections of the code can be annotated with feature expressions. The feature expressions are a separate, fully tool-supported expression language. Note that feature expressions can be evaluated and variants can be shown in the editor (see illustration below on the right) and also influence code generation.
- ⑥ A special abstraction for tasks is available. They are translated specific to the target platform. The *onceUponStartup* tasks become the contents of a *main()* function on win32, for other platforms this will be translated differently.
- ⑦ A log statement is available. It is also translated specifically for the platform.

Of course, the variability mechanism illustrated in section 4 based on Java is also available for C in MPS.

## 6 Summary and Outlook

As shown in the above examples, projectional editing, language workbenches, modular languages and the integration of (what's classically called) programming and modeling is a very powerful approach for programming in general. Because of the inherent domain-specificness of product line projects, it is especially suitable for product line development. The ability to express configurative variability on any kind of models or code is very useful.

Future work will address an industrial-strength implementation of the feature-annotation facilities for MPS together with a connection to pure::variants. We will also complete the MPS-implementation of C and use it for a real-world demo of the approach for embedded systems product line engineering.

## References

- [1] Martin Fowler, Language Workbenches, <http://martinfowler.com/articles/languageWorkbench.html>
- [2] Intentional Software, <http://intentsoft.com>
- [3] JetBrains, Meta Programming System, <http://jetbrains.com/mps>
- [4] <http://blog.intentsoft.com/files/intentional-dsl-devcon-2009-04-15---as-presented.pdf>
- [5] <http://www.jetbrains.com/youtrack/>
- [6] Krzysztof Czarnecki, Ulrich Eisenacker, Generative Programming, Addison-Wesley 2000

# Verknüpfung von kombinatorischem Plattform- und individuellem Produkt-Test für Software-Produktlinien

Andreas Wübbeke<sup>1</sup>, Sebastian Oster<sup>2</sup>

<sup>1</sup>Software Quality Lab – s-lab, Universität Paderborn  
Warburgerstraße 100, 33098 Paderborn  
awuebbeke@s-lab.de

<sup>2</sup>Fachgebiet Echtzeitsysteme, Technische Universität Darmstadt  
Merckstraße 25, 64283 Darmstadt  
oster@es.tu-darmstadt.de

**Abstract:** Das Software-Produktlinien Paradigma verspricht durch organisierte Wiederverwendung von Entwicklungsartefakten eine schnelle, kosteneffiziente und qualitativ hochwertige Entwicklung von ähnlichen Produkten auf Basis einer gemeinsamen Produktlinien-Plattform. Dabei entstehen für das Testen von Software-Produktlinien neue Herausforderungen: Zum einen entsteht die Frage, wie die wiederverwendbaren, variablen Artefakte der Produktlinien-Plattform getestet werden sollen und zum anderen, wie produktindividuelle Anforderungen im Test berücksichtigt werden können. Beide Fragestellungen müssen auch unter dem Gesichtspunkt der effektiven Spezifikation und Wiederverwendung von Testfällen mit Variabilität untersucht werden. Dieser Beitrag skizziert zur Lösung dieser Fragestellungen eine Verknüpfung aus kombinatorischem Testen der Produktlinien-Plattform und der Wiederverwendung von Testfällen für das Testen individueller Produktanforderungen. Durch die Verknüpfung von Plattform- und Produkttest kann die Effizienz des gesamten SPL-Tests gesteigert werden. Dies wird dadurch erreicht, dass im Produkttest die im Plattfortest bereits getestete Anforderungen nur unter bestimmten Umständen berücksichtigt werden.

## 1 Einleitung

Software-Produktlinien (SPL) bieten die Möglichkeit der organisierten Wiederverwendung von Entwicklungsartefakten für verschiedene Produkte auf Basis einer Produktlinien-Plattform ([CN01], [PBL05]). Dabei spielt die in der Plattform vorhandene Variabilität und deren Management eine zentrale Rolle. Diese Variabilität ermöglicht die Ableitung verschiedener Produkte aus der Produktlinien-Plattform, um die Wünsche des jeweiligen Kunden erfüllen zu können. Die Erstellung der Produktlinien-Plattform ist Teil des Domain-Engineering. Neben den aus der Plattform ausgewählten Funktionalitäten können nach der Produktableitung noch weitere, produktindividuelle Anforderungen/ Funktionalitäten zum Produkt hinzugefügt werden.

Dies geschieht im sogenannten Application-Engineering. Um die Qualität der Produktlinien-Plattform und von jedem daraus abgeleiteten Produkt zu überprüfen, können Tests spezifiziert und durchgeführt werden.

Dieser Beitrag beschreibt einen Ansatz zur effektiven Spezifikation von Tests für die Qualitätssicherung der Plattform- und von produktindividuellen Funktionalitäten. Dabei wird die in der Plattform vorhandene Variabilität berücksichtigt und ein Vorgehen für eine effektive Wiederverwendung von Testfällen spezifiziert. Durch die Abstimmung von Plattform- und Produkttest aufeinander kann die Effizienz des Testens gesteigert werden, da in Abhängigkeit von der Art und dem Umfang des Plattformentests nicht alle Testfälle im Rahmen des Produkttests berücksichtigt werden müssen.

Der weitere Aufbau dieses Beitrags ist wie folgt: In Kapitel 2 wird ein Überblick über verwandte Arbeiten gegeben. Kapitel 3 definiert die diesem Papier zugrunde liegende Problemstellung und ein laufendes Beispiel. Darauf folgt in Kapitel 4 die Vorstellung des Ansatzes zum kombinatorischen Plattformentest und in Kapitel 5 der Ansatz zum Testen mit produktindividuellen Anforderungen. Kapitel 6 fasst den Beitrag zusammen und gibt einen Ausblick auf weitere Arbeiten.

## 2 Verwandte Arbeiten

### 2.1 Testen von Software-Produktlinien

Für das Testen von SPLs existieren verschiedene Ansätze. Eine strukturierte Zusammenfassung ist in [TTK04] aufgeführt, welche die diversen Ansätze in vier Kategorien unterteilt:

Der Kategorie *product-by-product Testing* werden die Testansätze zugeordnet, die sich dem individuellen Test von einzelnen Produktinstanzen verschrieben haben. Dabei können bekannte Verfahren aus der Software-Engineering-Community eingesetzt werden. Der Test einer einzelnen Produktinstanz entspricht dem gängigen Software Test.

Beim *inkrementellen Testen* werden ebenfalls die Produkte einzeln getestet, jedoch wird der Testaufwand durch Anwendung des Regressionstestverfahrens reduziert.

Für die *Wiederverwendung von Test-Assets* werden im Domain Engineering bei dieser Testmethode wiederverwendbare Testartefakte entwickelt, die dann im Application Engineering für den Test der einzelnen Produkte verwendet werden. Dieser Testmethodik haben sich viele Ansätze verschrieben: In Bertolino und Gnesi [BG03] wird die Category-Partitioning-Methode um Variabilität erweitert, um diese wiederverwenden zu können. Abstrakte Domänentestfälle, die dann im Application-Engineering angepasst werden müssen, werden in [McG01] vorgestellt. In dem CADeT-[Ol08] als auch in dem ScnTED-Verfahren [RKPR05] werden ähnlich dazu wiederverwendbare Testmodelle im Domain-Engineering generiert und im Anschluss für jede Variante adaptiert und konkretisiert.

In [NLJ06] generieren die Autoren Tests auf Basis von Use Cases mit integrierter Variabilität um eine Wiederverwendung zu ermöglichen. Eine andere Art der Wiederverwendung wird in [WSS08] praktiziert. Dort dient eine einzige State-Machine um die Funktionalität der gesamten Produktlinie zu modellieren. Diese wird dazu verwendet um produktspezifische Testfälle zu generieren. Die bei dieser Art des SPL-Tests entstehenden Herausforderungen, sowie eine ausführliche Zusammenfassung der existierenden Ansätze bietet [Wu08] an.

**Aufteilung nach Verantwortlichkeit:** Bei diesem Testverfahren wird die Verantwortlichkeit für die verschiedenen Test-Level (Unit-, Integration-, System- und Akzeptanz-Test) dem Domain- und dem Application-Engineering Prozess zugewiesen. Zum Beispiel kann der Unit-Test dem Domain-Engineering und die restlichen Test-Level dem Application-Engineering zugewiesen werden.

Allen Testverfahren ist gemeinsam, dass jedes Produkt einzeln getestet werden muss, sei es nun individuell oder mit Hilfe von Wiederverwendungs- oder durch Regressionstechniken. Auf Grund der Variabilität und der immens hohen Anzahl von möglichen Produktinstanzen, ist es häufig nicht möglich alle Produkte einzeln zu testen. Um die Menge der zu testenden Produkte zu minimieren, kann eine repräsentative Menge von Produkten generiert werden, die kleiner ist, als alle möglichen Produktinstanzen. Die Generierung einer solchen Menge ist NP-vollständig [Sc07] was zur Folge hat, dass nicht die absolut minimale Menge von Produkten erstellt wird. In [Sc07] wird eine Heuristik zur Generierung einer repräsentativen Menge von Produkten auf Basis von Anforderungen vorgestellt. Eine weitere Heuristik basiert auf kombinatorischen Testen [OS09] und wird in diesem Beitrag als kombinatorischer Plattformtest verwendet.

## **2.2 Kombinatorisches Testen**

Anstatt alle möglichen Kombinationen von Eingabeparametern eines Programms zu testen, kann kombinatorisches Testen eingesetzt werden um den Testaufwand zu reduzieren. Für das Testen aller Eingabekombinationen bei einem Programm mit 12 Eingabefeldern mit jeweils drei Möglichkeiten sind 531.441 Testfälle nötig, um alle Kombinationen zu erfassen. Eine Übersicht von kombinatorischen Algorithmen mit dem Ziel den Testaufwand zu reduzieren, ist in [CDS07] gegeben. Dabei ist die Zielsetzung eines jeden Ansatzes immer, eine hohe Testabdeckung sicherzustellen. Eine Anwendung des kombinatorischen Testens ist das paarweise Testen. Diesem Ansatz liegt die Annahme zu Grunde, dass ein Großteil aller Softwarefehler aus einzelnen Datenwerten oder der Interaktion zweier Datenwerte resultiert [SM98]. Daher werden Eingabefelder nur paarweise miteinander getestet, wobei zwischen diesen zwei Feldern weiterhin alle möglichen Kombinationen gebildet werden. Zwei sehr bekannte Ansätze sind AETG [Co94] und IPO [LT98].

## 2.3 Variabilitätsmanagement für wiederverwendbare Testfälle

Ein Kunde wählt die Eigenschaften seiner Produktinstanz aus der Produktlinien-Plattform typischerweise anhand von Anforderungen oder Features aus. Daher muss für die Auswahl von Testfällen für bestimmte Produktinstanzen einer Produktlinie die Verfolgbarkeit der Variabilität von Anforderungen/ Features (Teil des Problemraums) zu Testfällen (Teil des Lösungsraums) durch das Variabilitätsmanagement ermöglicht werden. Das Problem der Verfolgbarkeit von Variabilität wird zum Beispiel in [BBM05] und [BM05] beschrieben. Dabei ist die Modellierung von Variabilität in Anforderungs- und Testartefakten ein wichtiger Bestandteil, um diese wiederverwendbar zu machen. Einige der in Abschnitt 2.1 vorgestellten Ansätze bieten Lösungen für die Modellierung von Variabilität in diesen Dokumenten an. Im Kontext des Variabilitätsmanagements existieren wenige Ansätze, die die geforderte Verfolgbarkeit zwischen Anforderungs- und Testfalldokumente adressieren: Einige Ansätze verwenden dazu ein orthogonales Variationsmodell (OVM) für die Verfolgbarkeit ([PM06] und [SJ03]). Das Featuremodell [Ka90] stellt in einigen Ansätzen und auch in diesem Beitrag die Modellierungsform für das OVM dar. Im nächsten Kapitel wird zunächst auf Basis eines Featuremodells eine Beispielproduktlinie eingeführt und mit deren Hilfe im zweiten Abschnitt des Kapitels die dem Beitrag zugrunde liegende Problemstellung motiviert.

## 3 Laufendes Beispiel und Problemdefinition

### 3.1 Laufendes Beispiel

Als Anwendungsszenario für die Verknüpfung von kombinatorischem Plattform- und individuellem Produkt-Test bei SPLs dient ein Subset der Android-Handy-Produktlinie, welche für die Forschung und Lehre an der TU Darmstadt entwickelt wird. Basis für diese SPL ist die Google Android-Plattform. Für die Veranschaulichung wird zunächst die Handy-SPL als Produktlinien-Plattform vorgestellt und anschließend eine produktindividuelle Erweiterung. Die Handy-SPL wird in Abbildung 1 als Featuremodell dargestellt und im Folgenden näher erläutert.

Die Handy-SPL verfügt über diverse Grundfunktionen, verschiedene Datenübertragungsmöglichkeiten sowie einige Extras. Die Features *Telefon* und *SMS* sind Pflicht-Features und in jeder Instanz dieser SPL enthalten. Das Versenden von *MMS* ist jedoch optional. Für den Datenaustausch stehen *WLAN*, *Bluetooth* und *UMTS* zur Verfügung. Da es sich bei diesem Handy um ein UMTS-Handy handeln soll, ist dieses Feature verpflichtend. Als Extras werden dem Kunden ein integrierter *MP3*-Player und eine *Kamera* angeboten. Diese sind in einer Oder-Gruppe enthalten, was bedeutet, dass eine beliebige Kombination dieser Features ausgewählt werden kann. Es muss aber mindestens eines der Beiden bei der Produktinstanziierung ausgewählt werden. Für die Kamera kann entweder eine 3 Megapixel (*3MP*) oder ein 8 Megapixel (*8 MP*) Chip verbaut werden. Desweiteren enthält die Handy-SPL noch zwei zusätzliche Constraints, die festlegen, dass die Auswahl des *MMS*-Features nur möglich ist, wenn eine *Kamera* ausgewählt wird (require-Kante).

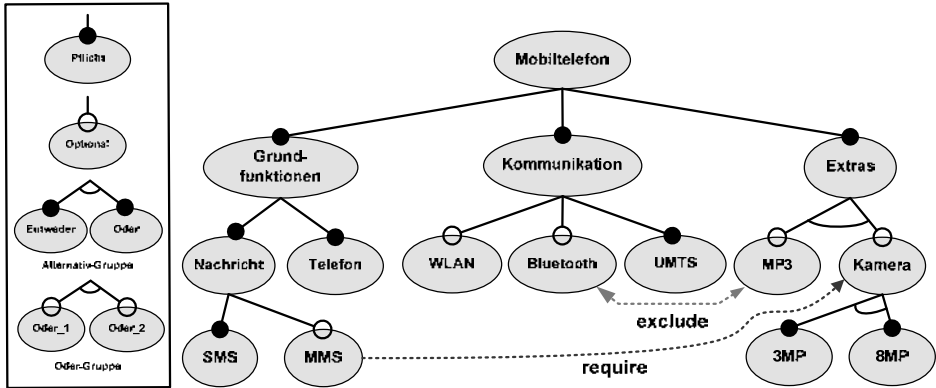


Abbildung 1 Featuremodell der Handy-SPL

Weiterhin schließen sich bei der Instanziierung *Bluetooth* und *MP3*-Player aus (exclude-Kante), was bedeutet, dass diese niemals gemeinsam ausgewählt werden dürfen. Diese Produktlinie ermöglicht die Instanziierung von 26 gültigen Produktinstanzen.

Der zweite Teil unseres Beispiels stellt eine produktindividuelle Erweiterung für bestimmte Produktinstanzen dar. Es handelt sich dabei um das Spiel *Atomic Bomberman*<sup>1</sup>, welches auf bestimmten Produktinstanzen der Handy-SPL spielbar sein soll. Dies bedeutet, dass das Spiel nicht Teil der Handy-SPL selbst ist, sondern nach der Produktinstanziierung im Rahmen des Application-Engineering zum Produkt hinzugefügt wird. In unserem Beispiel wird das Spiel für eine existierende Produktinstanz im Rahmen des Application-Engineering hinzugefügt. Die Produktinstanz ist als Featuremodell in Abbildung 2 dargestellt. In diesem Fall ist die Variabilität im Modell gebunden worden.

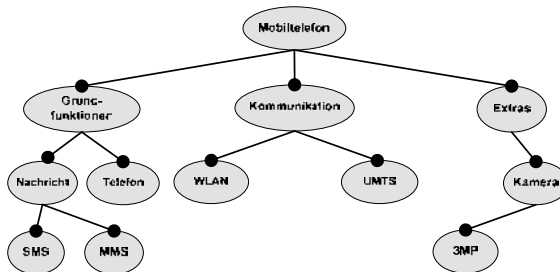


Abbildung 2 Produktinstanzen für das Spiel Atomic Bomberman

<sup>1</sup> [http://en.wikipedia.org/wiki/Atomic\\_Bomberman](http://en.wikipedia.org/wiki/Atomic_Bomberman)

### 3.2 Problemdefinition

Das Testen einer SPL stellt Herausforderungen an die Qualitätssicherung, welche über das Testen von Einzelsystemen hinausgeht. Dabei ist zunächst zu beachten, dass es nicht möglich sein kann, jedes einzelne, instanziierbare Produkt der SPL zu testen, da aufgrund des mit jedem Feature wachsenden Variationsgrades [Ma07] die Anzahl der zu testenden Instanzen sehr schnell wächst. Bereits in unserem kleinen Beispiel aus Abschnitt 3.1 wären dies 26 zu testende Instanzen. Somit ist es beim Test der Plattform zunächst wichtig, die Anzahl der zu testenden Feature-Kombinationen zu reduzieren und dabei die Aussagekraft des Tests zu erhalten. Die Problemstellung im Domain-Engineering wird in Abbildung 3 oben illustriert. Dabei sollen in unserem Beispiel aus dem Featuremodell der Handy-SPL auf kombinatorischem Wege Produktinstanzen erstellt werden. Zu diesen werden dann Testfälle spezifiziert. Diese Testfälle testen dann die kombinatorischen Produktinstanzen, d. h. in unserem Fall: Handys.

In Abbildung 3 unten ist die Problemstellung im Application-Engineering dargestellt: Wenn nun eine bestimmte Produktinstanz aus der SPL-Plattform abgeleitet worden ist, kann diese im Rahmen des Application-Engineerings um produktindividuelle Anforderungen erweitert werden. In unserem Beispiel ist dies das Spiel Atomic Bomberman.

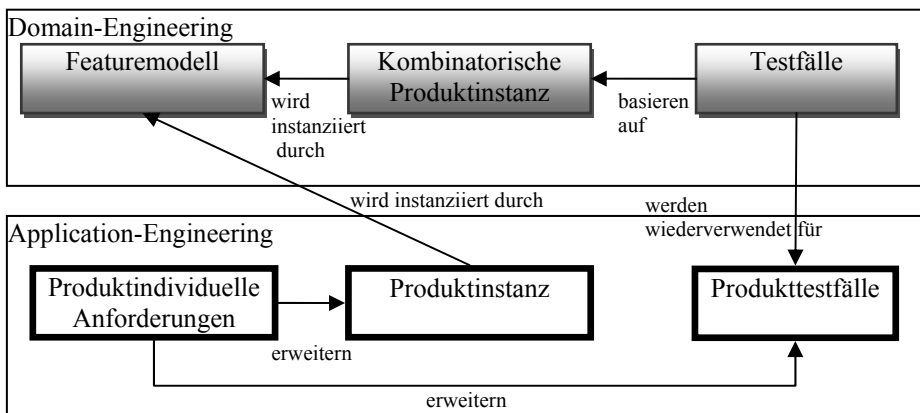


Abbildung 3 Problemdefinition für das Testen von Software-Produktlinien

An dieser Stelle ist es von Interesse diese neuen Funktionalitäten im Hinblick auf die Integrationen mit existierenden Funktionalitäten aus der Plattform, zu testen. Für diese Aufgabe sollten die Testfälle des Domain-Engineerings wiederverwendbar gestaltet sein und um Aspekte, basierend auf den produktindividuellen Anforderungen, erweitert werden. Somit entstehen Produkttestfälle. Diese können dann z.B. durch wiederverwendbare Testartefakte, durch Regressionstest oder durch product-by-product testing getestet werden.

Der Ansatz zur Lösung der beiden Teilprobleme wird nun in Kapitel 4 und 5 anhand des laufenden Beispiels erläutert.

## 4 Kombinatorischer Plattfortest im Plattform-Engineering

Das kombinatorische SPL-Testverfahren ([OS09], [ORS10]) basiert auf der Idee, dass Features als Parameter der dazugehörigen SPL angesehen werden können. Es scheint daher vielversprechend, bekannte Verfahren aus dem Software Test auf SPL-Tests anzuwenden, die auf die Reduktion des Testaufwands auf Basis der Parameter abzielen. Daher werden in [ORS10] Algorithmen vorgestellt, die das Featuremodell für das paarweise Testen vorbereiten und anschließend paarweises Testen realisieren. Letzterer ist in der Lage, Constraints zwischen Features auszuwerten und durch die Anwendung von Forward Checking die Erstellung von ungültigen Featurekombinationen zu verhindern. Somit werden ausschließlich gültige Produkte generiert. Diese Produkte können dann mit gängigen Testmethoden getestet werden, da sie keine Variabilität mehr enthalten. In diesem Abschnitt wird das Verfahren grob beschrieben. Für eine detaillierte Beschreibung wird auf [ORS10] verwiesen.

### 4.1 Tiefenreduktion des Featuremodells

Paarweises Testen benötigt für die Ausführung Parameter mit korrespondierenden Werten. Die Parameter werden dann paarweise kombiniert und ihre Werte miteinander getestet. Das Featuremodell muss für die Anwendung des paarweisen Testens angepasst werden, was die Tiefenreduktion realisiert [ORS10]. Diese Tiefenreduktion setzt sich aus zwei Schritten zusammen.

- 1) Alle Features inklusive Ihrer Notation werden iterativ nach oben gezogen, so dass sie direkt dem Wurzelknoten untergeordnet sind (vgl. Abbildung 4). Die semantische Äquivalenz zum Ursprungs-Featuremodell kann durch Übersetzung in logische Ausdrücke nachgewiesen werden.
- 2) Anschließend werden allen Features Kind-Knoten zugeordnet, die den möglichen Belegungen der Features entsprechen (vgl. Abbildung 5). Zum Beispiel erhält ein optionales Feature, wie *WLAN* die Werte *WLAN* und  $\neg$ *WLAN*. Diese werden als eine Alternativ-Gruppe dargestellt, da genau einer dieser Werte immer gewählt werden muss.



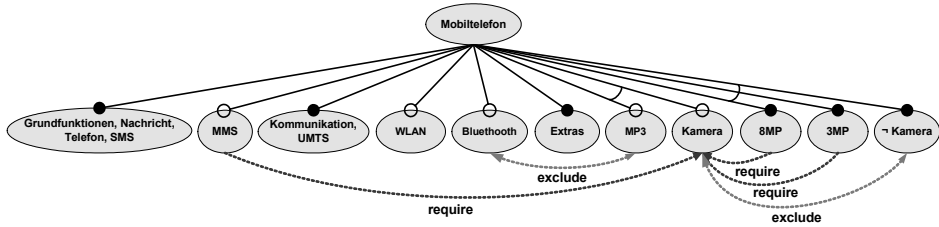


Abbildung 4: Tiefenreduziertes Featuremodell des Anwendungsbeispiels

Nach der Tiefenreduktion verfügt das Featuremodell über Parameter (Zeile b in Abbildung 5) und Parameterwerten (Zeile c in Abbildung 5).

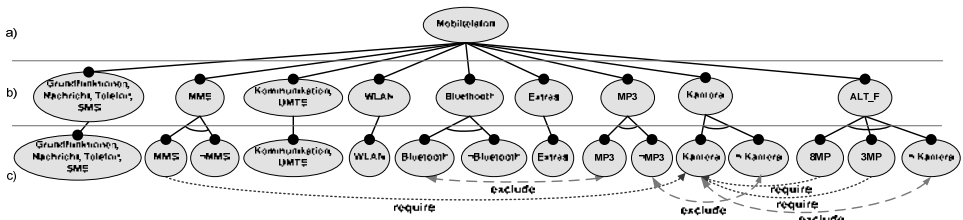


Abbildung 5: Tiefenreduziertes Featuremodell mit Parameterwerten

Wählt man für jeden Parameter genau einen Wert aus, dann entstehen, unter Berücksichtigung der require- und exclude-Kanten, Produktinstanzen der SPL. Für diese Handy-SPL ergeben sich 26 gültige Produktinstanzen.

## 4.2 Produktinstanz Erzeugung

Unter Verwendung der Parameter und Parameterwerten des tiefenreduzierten Featuremodells, kann anschließend paarweises Testen angewendet werden. Zur Einhaltung der Constraints zwischen den einzelnen Features wurde in [ORS10] ein Algorithmus entwickelt, der zum einen die binären Constraints zwischen Features auswerten und berücksichtigen kann und durch Forward Checking verhindert, dass unzulässige Produktinstanzen gebildet werden.

G,N,T,S	Ko,UMTS	Extras	MMS	WLAN	BT	MP3	Kamera	ALT_F
G,N,T,S	Ko,UMTS	Extras	MMS	WLAN	BT	~MP3	Kamera	8MP
G,N,T,S	Ko,UMTS	Extras	~MMS	~WLAN	~BT	MP3	~Kamera	~Kamera
G,N,T,S	Ko,UMTS	Extras	~MMS	WLAN	~BT	~MP3	Kamera	3MP
G,N,T,S	Ko,UMTS	Extras	MMS	~WLAN	~BT	MP3	Kamera	3MP
G,N,T,S	Ko,UMTS	Extras	~MMS	~WLAN	BT	~MP3	Kamera	8MP
G,N,T,S	Ko,UMTS	Extras	~MMS	WLAN	~BT	MP3	~Kamera	~Kamera
G,N,T,S	Ko,UMTS	Extras	~MMS	WLAN	BT	~MP3	Kamera	3MP
G.N.T.S	Ko.UMTS	Extras	~MMS	WLAN	~BT	MP3	Kamera	8MP

Abbildung 6 Resultierende Kombinationen aus dem paarweisen Testansatz für die Handy-SPL

Angewendet auf das Beispiel berechnet der Algorithmus die in Abbildung 6 angegebenen 8 Produktinstanzen, mit denen alle paarweisen Kombinationen von Parametern und Parameterwerten der Handy-SPL abgedeckt sind. Dieses Testverfahren wurde bereits als pure::variants<sup>2</sup> Plug-In realisiert.

## 5 Individueller Produkttest im Application-Engineering

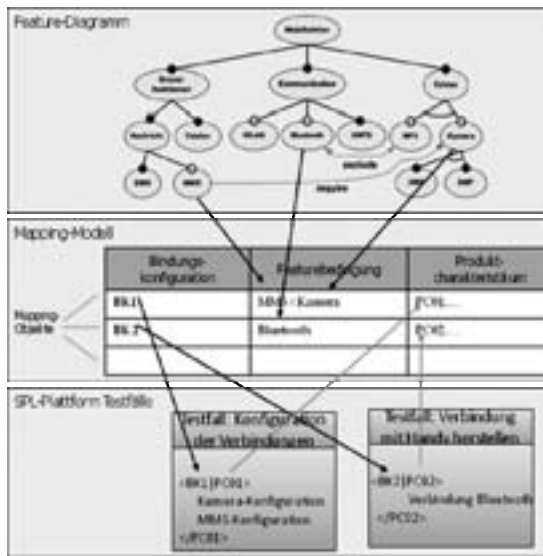


Abbildung 7 Mapping zwischen Featuremodell und SPL-Plattformtestfällen für die Beispiel-SPL

Für den individuellen Produkttest müssen zunächst die Testfälle des Plattformtests auf die Features des Featuremodells gemapped werden. Dieses Mapping wurde in [Ob09] entwickelt. Abbildung 7 zeigt die Funktionsweise des Mappings. Dabei werden Features des Featuremodells zusammen mit Produktcharakteristika von Testfällen in sogenannten Mapping Objekten miteinander in Beziehung gesetzt, wie in der Mitte der Abbildung dargestellt ist. Die Features werden in Form einer Feature-Bedingung zusammen mit den jeweils davon abhängigen Produktcharakteristika im Mapping Objekt notiert. Jedes Mapping-Objekt hat darüber hinaus einen eindeutigen Bezeichner. Dieser ist die Bindungskonfiguration. Die Bindungskonfiguration gemeinsam mit dem Bezeichner des Produktcharakteristikums ermöglicht in einem Testfall die Auswahl des Produktcharakteristikums in Abhängigkeit zu einer Feature-Auswahl. Alle Mapping Objekte zusammen bilden das Mapping-Modell der SPL. Im Beispiel in Abbildung 7 sind zwei Bindungskonfigurationen angegeben. Falls das instanziierte Handy über die Features *MMS* und *Kamera* verfügt, wird über die Bindungskonfiguration BK1 im Testfall *Testfall: Konfiguration der Verbindungen* das Produktcharakteristikum zur Kamera- und MMS-Konfiguration hinzugefügt.

<sup>2</sup> <http://www.pure-systems.com/>

Ein zweites Beispiel ist die Bindungskonfiguration BK2. Hier wird in Abhängigkeit vom Feature *Bluetooth* im Testfall *Testfall: Verbindung mit Handy herstellen* ein Test mit der Verbindung via Bluetooth hinzugefügt.

Mit Hilfe des Mapping Modells können nun die für unsere Beispiel-Produktkonfiguration aus Abbildung 2 die zum Testen notwendigen Testfälle konfiguriert werden. In unserem Beispiel aus Abbildung 7 wäre dies nur der Testfall *Testfall: Konfiguration der Verbindungen*, da das Feature Bluetooth nicht Teil der Produktinstanz ist. Dieser Testfall muss nun um die produktindividuellen Anforderungen für das Spiel Atomic Bomberman erweitert werden. Das Spiel benötigt eine spezielle Serververbindung, die über WLAN während des Spiels eine Videokonferenz mit den anderen Mitspielern durchführt. Dabei sehen sich die Spieler gegenseitig, was über die im Handy vorhandene Kamera ermöglicht wird. Aus diesem Grund wird der Testfall *Testfall: Konfiguration der Verbindungen* um einen weiteren Aspekt *Bombermann-Kamera-Server-Konfiguration* erweitert, wie in Abbildung 8 illustriert wird.

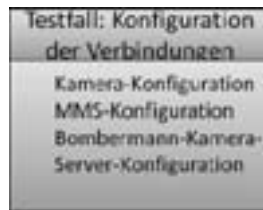


Abbildung 8 Produkt-Testfall für Handy mit Atomic Bomberman

Alle Testfälle die im Rahmen des Application-Engineerings aufgrund von produktindividuellen Anforderungen erweitert wurden, werden im Rahmen des Produkttest noch einmal durchgeführt. Der vorgestellte Ansatz für den Produkttest wurde im Rahmen von [Ob09] anhand eines Praxisbeispiels evaluiert.

## 6 Zusammenfassung und Ausblick

In diesem Beitrag wurde in Kapitel 3 die Problemstellung des Plattform- und Produkttests unter Berücksichtigung von individuellen Produkthanforderungen skizziert. Der Ansatz zum Plattformtest wurde bereits in [OS09] und [ORS10] beschrieben und in Kapitel 4 skizziert. In diesem Beitrag wurde der Ansatz zum Plattformtest um den Test von Produktinstanzen erweitert, sodass produktindividuelle Anforderungen berücksichtigt werden können. Die Problemstellung des Testens von Produktinstanzen wurde in [Wu09] beschrieben. Der Ansatz für den Produkttest wurde in Kapitel 5 beschrieben. Hierbei wurde eine Wiederverwendung von Plattformtestfällen über ein Mapping-Modell-Ansatz realisiert. Durch die Wiederverwendung von Testfällen für bestimmte Produkte und die Integration von produktindividuellen Anforderungen in diese, können die im Produkttest auszuführenden Testfälle bestimmt werden.

Durch die Kombination beider Ansätze kann der Testaufwand reduziert werden. Anstatt alle möglichen Kombinationen inklusive aller produktindividuellen Eigenschaften zu testen, kann der Ansatz auf einem kleinen Subset ausgeführt werden. Der Ansatz bietet im Vergleich zu den in [TTK04] aufgeführten Test-Kategorien (siehe Kapitel 2) Vorteile: *Product by Product Testing* testet jedes Produkt für sich selbst und schafft somit keine Reduktion des Testaufwands im Vergleich zum vorgestellten Ansatz. *Inkrementelles Testen* reduziert den Aufwand, berücksichtigt aber bereits getestete Assets. Dadurch ist der Aufwand höher als im hier gezeigten Ansatz. Die *Wiederverwendung von Test-Assets* führt zu einer Reduktion des Aufwands bei der Testfallspezifikation, es werden aber für jedes Produkt alle Testfälle wiederum durchgeführt. Der hier gezeigte Ansatz hingegen reduziert die auf Produktebene durchzuführenden Testfälle auf die, die neue Anforderungen testen. Der hier gezeigte Ansatz reduziert also die Menge durchzuführenden Testfälle. Bei der *Aufteilung nach Verantwortlichkeit* werden manche Teststufen auf Plattformebene durchgeführt. Dadurch werden die im Produkttest durchzuführenden Testfälle reduziert. Es werden aber keine Testfälle aus der Plattform im Produkt wiederverwendet. Der hier beschriebene Ansatz verwendet die im Plattformtest erstellten Testfälle im Produkttest wieder, wodurch der Spezifikationsaufwand für Produkt-Testfälle reduziert wird. Zusammengefasst bedeutet dies, dass der beschriebene Ansatz durch kombinatorischen Plattformtest die Anzahl der im Produkttest durchzuführenden Testfälle reduziert und durch die Wiederverwendung von Testfällen aus der Plattform für Produkte den Spezifikationsaufwand für Produkt-Testfälle zusätzlich reduziert.

Für beide Ansätze werden, unabhängig von Ihrer Kombination, Weiterentwicklungen und Evaluationen forciert. Unter anderem, wird der kombinatorische Plattformtest auf triple-weises Testen, also auf dreier-Kombinationen, erweitert, um eine noch höhere Testabdeckung zu erreichen. Des Weiteren soll der individuelle Produkttest in die bereits bestehende *pure::variants* Implementierung des kombinatorischen Plattformtests integriert werden. Weiterhin gilt es den vorgestellten Ansatz in zukünftigen Arbeiten anhand von realen Industrieszenarien zu evaluieren, um das Potential der Reduktion zu bestimmen.

## Literaturverzeichnis

- [Be06] Bertolino, A.; Fantechi, A.; Gnesi, S.; Lami, G.: Product Line Use Cases: Scenario-Based Specification and Testing of Requirements. In (Käkölä, T.; Ducnas, J. C.; Hrsg.): Software Product Lines: Research Issues in Engineering and Requirements, Springer Verlag, pp. 425-445, 2006.
- [BBM05] Berg, K.; Bishop, J.; Muthig, D.: Tracing Software Product Line Variability: from Problem to Solution Space. In: Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT'05), South African Institute for Computer Scientists and Information Technologists, pp. 182-191, 2005.
- [BM05] Berg, K.; Muthig, D.: A Critical Analysis of Using Feature Models for Variability Management, Univerity of Pretoria, 2005.

- [CDS07] Cohen, M. B.; Dwyerund, M. B.; Shi, J.: Interaction Testing of Highly-Configurable Systems in the Presence of Constraints. In: Intl. Symp. on Software Testing and Analysis, pp. 129–139, 2007.
- [CN01] Clements, P.; Northrop, L.: Software Product Lines: Practices and Patterns, Addison-Wesley Professional, 2001.
- [Co94] Cohen, D. M.; Dalal, S. R.; Kajla, A.; Patton, G. C.: The Automatic Efficient Tests Generator. In: Fifth Intl. Symposium on Software Reliability Engineering, IEEE, pp. 303–309, 1994.
- [Ka90] Kang, K. C.; Cohen S.; Hess, J. A.; Novak, W. E.; Peterson, A. S.: Feature Oriented Domain Analysis (FODA) Feasibility Study, Technical Report CMU/SEI-90-TR21, 1990.
- [LT98] Lei, Y.; Tai, K. C.: In-Parameter Order: a Test Generation Strategy for Pairwise Testing. In: IEEE High Assurance Systems Engineering Symposium, pp. 254–261, 1998.
- [Ma07] von der Maßen, T.: Feature-basierte Modellierung und Analyse von Variabilität in Produktlinienanforderungen, Dissertation RWTH Aachen, 2007.
- [Mc01] McGregor, J. D.: Testing a Software Product Line. Carnegie Mellon Software Engineering Institute, 2001.
- [NLJ06] Nebut, C.; Le Traon, Y.; Jézéquel, J.-M.: Software Product Lines System Testing of Product Families: from Requirements to Test Cases. Springer Verlag, In (Käkölä, T.; Ducnas, J. C.; Hrsg.): Software Product Lines: Research Issues in Engineering and Requirements, Springer Verlag, pp. 447-478, 2006.
- [Ob09] Oberhokamp, C.: Konzeption und Vergleich von Testdesign-Strategien für Software-Produktlinien, Diplomarbeit Universität Paderborn, 2009.
- [OI08] Olimpiew, E. M.: Model-Based Testing for Software Product Lines. Doctoral Thesis George Mason University, 2008.
- [ORS10] Oster, S.; Ritter, P.; Schürr, A.: Featuremodellbasiertes und kombinatorisches Testen von Software-Produktlinien. In: Proceeding Software Engineering 2010, angenommen zur Veröffentlichung, 2010.
- [OS09] Oster, S.; Schürr, A.: Architekturgetriebenes Pairwise-Testing für Software-Produktlinien In: Proceeding Software Engineering 2009, pp. 131-134, März 2009.
- [PBL05] Pohl, K.; Böckle, G.; van der Linden, F. J.: Software Product Line Engineering: Foundations, Principles and Techniques, Springer, 2005.
- [PM06] Pohl, K.; Metzger, A.: Variability Management in Software Product Line Engineering. In: Proceedings of the 28th international conference on Software engineering (ICSE '06), ACM, pp. 1049-1050, 2006.
- [Re05] Reuys, A.; Kamsties, E.; Pohl, K.; Reis, S.: Szenario-basierter Systemtest von Software-Produktfamilien. In: Inform., Forsch. Entwickl., vol. 20, pp. 33-44, 2005.
- [Sc07] Scheidemann, K.: Verifying Families of System Configurations. Doctoral Thesis Technical University of Munich, 2007.
- [SM98] Stevens, B.; Mendelsohn, E.: Efficient Software Testing Protocols. In: Conference of the Centre for Advanced Studies on Collaborative research. IBM Press, 1998.
- [SJ03] Schmid, K.; John, I.: A Practical Approach to Full Life-Cycle Variability Management. In: International Workshop on Software Variability Management (SVM'03) at International Conference on Software Engineering, pp. 41-46, 2003.
- [TTK04] Tevanlinna, A.; Taina, J.; Kauppinen, R.: Product family testing: a survey. In: SIGSOFT Softw. Eng. Notes, ACM, vol. 29, pp. 12-17, 2004.
- [WSS08] Weißleder, S.; Sokenou, D.; Schlinglo, B.: Reusing State Machines for Automatic Test Generation in Product Lines. In: Proceedings of the 1st Workshop on Model-based Testing in Practice (MoTiP2008), 2008
- [Wu08] Wübbecke, A.: Towards an Efficient Reuse of Test Cases for Software Product Lines. In: (Thiel, S.; Pohl, K.; Hrsg.) Proceedings of the 12th International Software Product Line Conference (2008) Second Volume, Limerick (Ireland), Lero, pp. 361-368, 2008.



**Innovative Systeme zur Unterstützung der zivilen  
Sicherheit  
(Public Safety)**



# Workshop 'Innovative Systeme zur Unterstützung der zivilen Sicherheit: Architekturen und Gestaltungskonzepte' Geleitwort

Prof. Dr.-Ing. Rainer Koch

Zunehmend stehen sowohl im nationalen als auch im internationalen Forschungsumfeld Themen der zivilen Sicherheit im Mittelpunkt des Interesses. Verwiesen sei auf die Forschungsanstrengungen des Programms '*Forschung für die zivile Sicherheit*' im Rahmen der High-Tech-Strategie der Bundesregierung sowie auf diverse Forschungsinitiativen auf EU-Ebene. Ein besonderes Augenmerk liegt hierbei auf lösungsorientierten Ansätzen des Software Engineerings, um den Herausforderungen in den Bereichen flexibler Architekturen, der Anforderungsanalyse sowie den Entwicklungs- und Wartungsmethoden zu begegnen.

Mit dem Workshop '*Innovative Systeme zur Unterstützung der zivilen Sicherheit: Architekturen und Gestaltungskonzepte*' hat das Fachgebiet Computeranwendung und Integration in Konstruktion und Planung (C.I.K.) als ein in diesem Forschungsbereich etabliertes Institut den Wunsch nach verstärktem wissenschaftlichem Austausch in diesem Bereich aufgenommen.

Im Rahmen des Workshops wurden dazu Aspekte aus unterschiedlichen Projekten diskutiert und somit ein Querschnitt der aktuellen Forschungslandschaft abgebildet. Einreichungen stammen aus den Projekten VeRSiert<sup>1</sup>, MobisPro<sup>2</sup>, SinoVE, OrGaMIR<sup>3</sup>, ALARM<sup>4</sup> und PRONTO<sup>5</sup>.

Die positiven Ergebnisse dieses ersten Workshops motivieren dazu, den Rahmen der Tagungsreihe '*Software Engineering*' für eine Fortsetzung des wissenschaftlichen Dialogs in diesem Themenbereich zu nutzen.

An dieser Stelle gilt mein Dank den Co-Organisatoren, Frau Donovan-Kuhlisch und Herrn Benedikt Birkhäuser, sowie den Mitgliedern des Programmkomitees Herrn Prof. Dr.-Ing. habil. Sahin Albayrak, Herrn Dr.-Ing. Christian Erfurth, Frau Prof. Dr. Birgitta König-Ries, Herrn Dr. Jobst Löffler, Herrn Prof. Dr. Johannes Magenheimer, Frau Anna Maria Japs, Herrn Carsten Wirth, Herrn Tom Zentek und Herrn Dr. Thomas Ziegert für die jederzeit unkomplizierte und reibungslose Zusammenarbeit.

---

<sup>1</sup><http://www.versiert.info>

<sup>2</sup><http://www.mobis-pro.de>

<sup>3</sup><http://www.orgamir.de>

<sup>4</sup><http://www.alarm-projekt.de>

<sup>5</sup><http://www.ict-pronto.org>





# Smarte Systeme und Services für Freiheit, Recht und Ordnung

**Margarete Donovang-Kuhlisch**

Public Sector

IBM Deutschland GmbH

Gorch-Fock-Str. 4

D-53229 Bonn

mdk@de.ibm.com

**Abstract:** Mit diesem Impulsvortrag für den Workshop „Innovative Systeme zur Unterstützung der zivilen Sicherheit: Architekturen und Gestaltungskonzepte“ möchte ich die Diskussion in Bezug auf unterschiedliche Aspekte der zivilen Sicherheit anregen, so z.B. den Bereich „Schutz kritischer Infrastrukturen“. Darüber hinaus werde ich auch innovative Konzepte und Technologien für „smarte Verwaltung im Dienste von Bürgern und Wirtschaft“ vorstellen.

## 1 Problemstellung

Freiheit, Recht und Ordnung in einer globalisierten Gesellschaft sicher zu stellen, ist Aufgabe von Regierungen und Verwaltungen, die sich immer neuen demographischen und gesellschaftlichen Anforderungen ausgesetzt sehen<sup>1</sup>:

- Bedürfnisse alternder Bevölkerungen vs. Wahrung gesunder Finanzstrukturen
- Kompetenzkrise – Talent- und Belegschaftsmanagement
- Datenexplosion und Informationsflut
- Global-integrierte öffentliche Verwaltung – ohne Grenzen
- Transition vom e-Government zur dienstleistungsorientierten Verwaltung
- Private Vertragspartner in der ÖV – neuartige „public private partnerships“
- Notwendigkeit eines verstärkten Performance-Management in der Verwaltung
- Verpflichtung zur Verantwortung und Vordenkerschaft im Dienste der Umwelt
- Gewährleistung von Sicherheit und Privatsphäre in einer „flachen“ Welt
- Befähigung zum agilen und nachhaltigen Krisenmanagement.

## 2 Schutz kritischer Infrastrukturen

Abbildung 1 gibt einen Überblick über verschiedene „smart“ Systeme zum Schutz kritischer Infrastrukturen. Sie zeichnen sich durch die Attribute „instrumentiert“ (Sensoren), „vernetzt“ und „intelligent“ (Entscheidungsunterstützung und Effektoren) aus.

---

<sup>1</sup> Adaptiert aus: IBM Center for the Business of Government, 2009

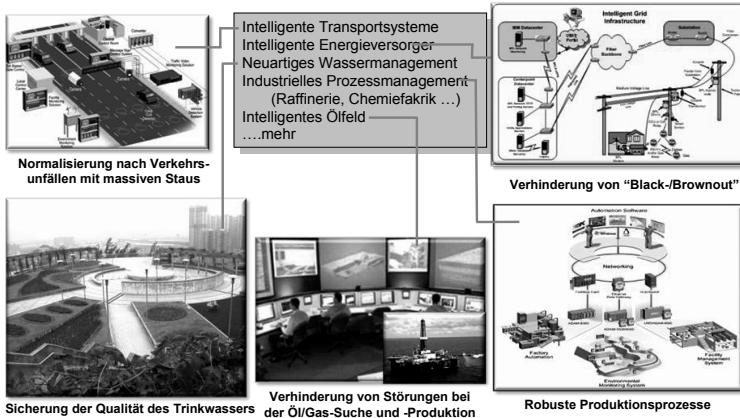


Abbildung 1: Schutz kritischer Infrastrukturen

## Smarte Dienstleistungen durch Intentionserkennung

In [DS09] haben wir einen Ansatz zur Erstellung und Implementierung context-sensitiver Prozess-Kontroll-Sprachen beschrieben, die sich durch die Erkennung von Intention in Texten in natürlicher Sprache abstützen. Dieser Ansatz kann für alle hoheitlichen Aufgaben (wie in Abbildung 2 aufgeführt) verfolgt werden.



Abbildung 2: Schutz kritischer Infrastrukturen

## Literaturverzeichnis

[DS09] Donovang-Kuhlisch, M.; Schade, U.: Smart e-Government Services through the Recognition of Common Intent. ICSOC Submission, Bonn, 2009.

# Ein Konzept zur organisationsübergreifenden Integration von IT-Systemen für die zivile Sicherheit

Wolf Engelbach, Heiko Roßnagel, Sandra Frings

Competence Team Informationsmanagement  
Fraunhofer IAO, Universität Stuttgart IAT  
Nobelstr. 12,  
70569 Stuttgart  
[vorname.nachname]@iao.fraunhofer.de

**Abstract:** IT-Systeme zur Verbesserung der Reaktionsfähigkeit bei ungeplanten Ereignissen im Umfeld von Großveranstaltungen und öffentlichen Verkehrsinfrastrukturen müssen mehrere Bedingungen erfüllen: 1) die Zusammenarbeit zahlreicher, teilweise wechselnder Organisationen unterstützen, 2) auch im Normalbetrieb konkrete Vorteile bringen und 3) untereinander verbindbar sein, um komplexere Fragestellungen behandeln zu können. Die dafür vorgestellte Konzeption verfolgt einen offenen Ansatz für fachliche und informationstechnische Schnittstellen, der im Kontext konkreter Veranstaltungs- und Ereignisszenarien für Köln zusammen mit einem Rollenmodell hergeleitet wurde und erprobt wird.

## 1 Einleitung

Der öffentliche Personennahverkehr (ÖPNV) in deutschen Großstädten und Ballungsräumen bietet eine Infrastruktur, die sich sehr individuell und flexibel von Bewohnern, Beschäftigten und Besuchern dieser Regionen verwenden lässt. Dementsprechend hoch sind die Anforderungen an die Abwicklung des normalen Verkehrsgeschehens und zugleich die Herausforderungen, wenn vermehrte Anstrengungen zur Gewährleistung eines höheren Sicherheitsniveaus gefordert werden [BM09]. In vielen Metropolregionen finden zudem immer häufiger Großveranstaltungen statt, bei denen enorme Besucherströme bewältigt werden müssen. Dabei treten Belastungsspitzen im öffentlichen Personennahverkehr (ÖPNV) auf, was zu zahlreichen verkehrs- und sicherheitstechnischen Herausforderungen führen kann. Durch die steigende Anzahl an Großveranstaltungen und immer kürzere Vorlaufzeiten werden die Organisation und die Durchführung immer komplexer und zeitkritischer. Schwierigkeiten entstehen beispielsweise durch unzureichenden Informationsaustausch zwischen den Verantwortlichen, fehlende Informationsweitergabe an Teilnehmer und Fahrgäste, mangelhafte Schulungen des angeworbenen Sicherheitspersonals, knappe Finanzmittel der beteiligten Institutionen, uneinheitliches Datenmanagement der einzelnen Einsatzzentralen, eingeschränkten Informationsaustausch im Krisenfall sowie einer späten Erkennung von Krisenfällen [RE08]. Etablierte und neuartige IT-Instrumente bieten in dieser Situation die Chance, mehr Transparenz zu erlangen, virtuell Optionen und ihre Auswirkungen durchzuspielen sowie die Kommunikation zwischen Akteuren zu verbessern [TC04] [YD05].

Ziel des durch das BMBF im Rahmen des Programms "Forschung für die zivile Sicherheit" [BM09] als Teil der High-Tech-Strategie geförderten Forschungsprojekts VeRSiert [PV08] ist es daher, eine bessere organisatorische und informationstechnische Vernetzung von Nahverkehrsgesellschaften, Einsatzkräften, Veranstaltern und Fahrgästen zu erreichen, um die Sicherheit im ÖPNV bei Großveranstaltungen zu erhöhen. Eine wesentliche Voraussetzung dafür ist es ein Konzept zur organisationsübergreifenden Integration von IT-Systemen. Dieser Beitrag stellt ein solches Konzept vor. Dafür werden in Abschnitt 2 verwandte Arbeiten betrachtet. In Abschnitt 3 werden der Untersuchungskontext dieser Arbeit präsentiert und die besonderen Herausforderungen in Bezug auf zivile Sicherheit im ÖPNV abgeleitet. Für diesen Untersuchungskontext wurde ein Rollenmodell erarbeitet, das in Abschnitt 4 vorgestellt wird. Dieses Rollenmodell dient als Grundlage für die IT-Gesamtkonzeption, die in Abschnitt 5 präsentiert wird, bevor die wesentlichen Ergebnisse des Beitrags in Abschnitt 6 zusammengefasst werden.

## 2 Verwandte Arbeiten

Mit den Herausforderungen zur Gestaltung von IT-Systemen für Sicherheitsfragen im Zusammenhang mit ÖPNV haben sich teilweise auch schon andere Projekte, insbesondere mit Förderung der EU, auseinander gesetzt [DS09]; derzeit laufen zudem zahlreiche Projekte zum Schutz kritischer Infrastrukturen innerhalb des Programms "Forschung für die zivile Sicherheit". Sie haben dabei allerdings immer bestimmte Schwerpunkte gelegt, die aufgegriffen werden können, aber den Kern des hier vorgestellten Ansatzes nicht berühren. Wichtige Anregungen enthalten:

**MODURBAN**, das bis 2008 lief, zielte auf die Definition von Systemarchitekturen und IT-Schnittstellen, um die Betriebskosten von Nahverkehrssystemen zu reduzieren. Hierbei ging es insbesondere um die Schnittstellen innerhalb von und zwischen Verkehrsunternehmen sowie innerhalb einer Gattung von IT-Systemen wie Fahrgastinformationen oder Videoüberwachung [MOD09]. Der Ansatz dieses Artikels hat einen Schwerpunkt auf der Kooperation verschiedener Akteursgruppen und heterogener IT-Systeme.

Das laufende Projekt **MODSAFE** adressiert die Definition von Sicherheitsanforderungen, beispielsweise Sicherheit vor kriminellen Aktivitäten, und der dafür sinnvollen Rollenmodelle und Zertifizierungsprozesse im ÖPNV [MB09]. Der in diesem Artikel vorgestellte Ansatz geht hinsichtlich der Umsetzung in IT-Systemen weiter, kann aber die inhaltlichen Vorstellungen aufgreifen.

Das 2009 beendete Projekt **COUNTERACT** bereitete Grundlagen zur Verbesserung der Sicherheit gegen terroristische Anschläge auf, u.a. für Akteure im ÖPNV [CO09]. Die Problemanalysen und Handlungsanregungen sind stark organisatorisch ausgerichtet und können in die hier vorgestellte IT-Architektur eingebracht werden.

Das 2010 endende Projekt **DEMASST** erarbeitet eine Vision für System-of-System-Lösungen bezogen auf Sicherheitsanforderungen im öffentlichen Personenverkehr [AN09]. Die vorliegenden Zwischenergebnisse werden für den hier vorgestellten Ansatz aufgegriffen und erweitert.

### 3 Erprobungskontext Großveranstaltungen in Köln

Das Forschungsprojekt VeRSiert zielt für das Beispiel Köln darauf ab, bei Großveranstaltungen durch eine optimierte Vernetzung von Verkehrsgesellschaften, Einsatzkräften, Veranstaltern und Fahrgästen die Sicherheit insbesondere im öffentlichen Personennahverkehr zu erhöhen. Die Kölner Verkehrsbetriebe (KVB), die Stadt Köln, der Nahverkehr Rheinland als Projektkoordinator sowie anderer Anwender in Köln bringen dazu ihre vielfältigen Erfahrungen bezüglich des Managements von Großveranstaltungen, der Zusammenarbeit und des Umgangs mit ungeplanten Ereignissen ein. Zur Gewährleistung größtmöglicher Sicherheit bei Großveranstaltungen werden dabei die Phasen der Vorbereitung, Durchführung (inkl. An- und Abreise) und Nachbereitung betrachtet [HE04]. Im Projekt werden organisatorische Maßnahmen und die differenzierte Nutzung von Informations- und Kommunikationstechniken sowie Schulungskonzepte analysiert, konzipiert, umgesetzt, modellhaft erprobt und empirisch evaluiert. Die folgenden organisatorischen und informationstechnischen Bausteine von VeRSiert sind für sich selbstständig konzipiert, können aber auch fachlich und informationstechnisch miteinander vernetzt eingesetzt werden: Informations- und Kooperationsportal, Simulations-, Videoanalyse-, Mobile-Dienste- und Sicherheitsbefragungs-Plattformen sowie Bausteine zur Mitarbeiterschulung und zum Verkehrsmanagement [PV08].

Leitfadengestützte Befragungen und veranstaltungsorientierte Prozess- sowie Aufgabenanalysen bei Kölner Akteuren haben einen Überblick zur Planung, Durchführung und Nachbereitung von Großveranstaltungen verschafft, insbesondere bezogen auf organisationsübergreifende Zusammenarbeit. Dabei bestehen seitens der Veranstalter, der Genehmigungsbehörde, der Fachbehörden, der Verkehrseinrichtungen sowie der Sicherheitskräfte unterschiedliche Interessen und Anforderungen hinsichtlich einer informationstechnischen Unterstützung der eigenen Aufgaben sowie der Zusammenarbeit. Damit die IT-Potenziale konstruktiv genutzt werden können, lassen sich aus den Analysen mehrere Anforderungen ableiten und durch frühere Untersuchungen bestätigen:

1) Es ist die Zusammenarbeit zahlreicher, teilweise wechselnder Organisationen zu unterstützen, die für den Verkehrsablauf oder für Sicherheitsfragen in einer Region oder für eine bestimmte Zeit verantwortlich sind [MJ07]. Innerhalb komplexer Zusammenhänge von Verkehrsnetzen und Stadtsystemen haben Entscheidungen einzelner Akteure gravierende Auswirkungen auf andere Verantwortungsbereiche und sollten entsprechend zwischen ihnen abgestimmt oder zumindest untereinander transparent sein.

2) Die dafür einzusetzenden IT-Systeme sollen nicht nur in Krisensituation zum Einsatz kommen, sondern gerade auch im Normalbetrieb konkrete Vorteile bringen [MH07]. Nur dann ist zu erwarten, dass in sie investiert wird und dass Benutzer mit diesen Systemen umgehen können. Für Verkehrsunternehmen sind dabei Vandalismus und Schlägereien als alltägliche Sicherheitsherausforderungen eher eine Motivation zum Handeln als ungewiss erscheinende Ereignisse wie terroristische Attentate. Generell sind daher eher für viele Risiken einsetzbare IT-Systeme als sehr spezialisierte zu bevorzugen.

3) IT-Systeme sollen untereinander flexible verbindbar sein, um auch komplexere Fragestellungen behandeln zu können. Dabei ist die heterogene Ausgangslage hinsichtlich der

vorhandenen IT-Infrastruktur bei den beteiligten Akteuren zu berücksichtigen. Zugleich können so die vielen möglichen zukünftigen Sicherheitslösungen einfacher integriert werden, egal ob sie sich auf Arbeitsabläufe, organisatorische Zuständigkeiten oder auf neue IT-Systeme beziehen.

Um das Zusammenwirken aller Konzepte im Blick zu behalten und die exemplarische Demonstration des Zusammenspiels zu fördern, wurden zwei wesentliche verbindende Grundlagen abgestimmt, auf denen alle konzeptionellen Aktivitäten aufbauen und durch die miteinander kompatible Ergebnisse gesichert werden. Durch diesen Ansatz können durchgängige Arbeitsprozesse einfacher informationstechnisch demonstriert werden:

- **Arbeitsszenarien:**  
Alle Projektbausteine orientieren sich an den drei Veranstaltungsszenarien „Fußball-Bundesligaspiel“, „Kölner Lichter“ und „Deutscher Evangelischer Kirchentag“ sowie an sieben Szenarien ungeplanter Ereignisse: Terroristischer Anschlag, Unwetter, Personenschaden mit Todesfolge, Defekt einer U-Bahn, Amoklauf in Straßenbahn, Bombendrohung, Sperrung des Hauptbahnhofes aufgrund Fehleinschätzung [RO08].
- **Untersuchungsräume:**  
Empirische Erhebungen und Demonstrationselemente, die sich nicht auf ganz Köln beziehen, werden exemplarisch für die gemeinsamen Untersuchungsräume „Kölner Hauptbahnhof“ sowie „Haltestellenbereich Rhein-Energie-Stadion“ konkretisiert.

## 4      **Gemeinsames Rollenmodell**

Für die stringente Verwendung der verschiedenen Bausteine ist darüber hinaus ein gemeinsames Rollenmodell entwickelt worden. Dies dient der begrifflichen Klarheit der Kommunikation und schärft das gemeinsame Verständnis über Zielgruppen und Anwendungsfälle. Zugleich stellt es sicher, dass zentrale Akteursarten und Gesichtspunkte für alle Bausteine und Plattformen beachtet werden. Ergänzend lassen sich so Überlegungen zu einer teilweise gemeinsamen Benutzerverwaltung und abgestimmter Berechtigungsstufen anstellen. Die Diskussion von Rollen für die Bausteine und Plattformen umfasst drei klar unterscheidbare Abstraktionsebenen:

- **Veranstaltungswelt (Objektebene):** Personen, die bei einer Veranstaltung unterwegs sind, d.h. sich währenddessen in den relevanten Gebieten bewegen.
- **Organisationswelt (Managementebene):** Personen, die für die Planung, Durchführung und Nachbereitung einer Veranstaltung zuständig sind und sich währenddessen teilweise in diversen Leitstellen und Organisationszentralen befinden.
- **Systemwelt (Metaebene):** Personen, die inhaltlich oder informationstechnisch mit den Plattformen arbeiten, teilweise jedoch nicht während einer Veranstaltung.

Für diese drei Abstraktionsebenen wurde spezifisch für die Bausteine und Plattformen diskutiert, welche Rollen relevant sein können. Für alle diese Bausteine wurde das folgende gemeinsame Rollenmodell abgeleitet und jeweils instanziiert (siehe Abb. 1).

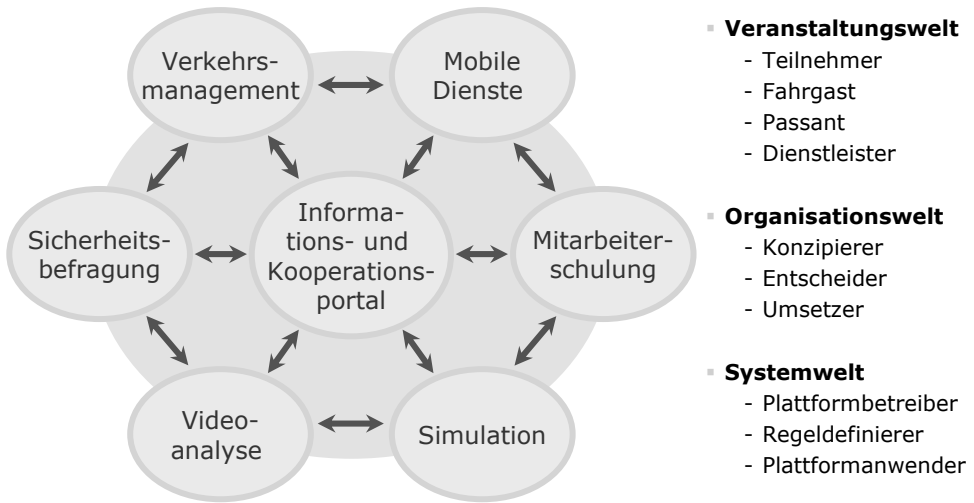


Abb. 1: Gemeinsames Rollenmodell der Bausteine von VeRSiert

## 5 IT-Gesamtkonzeption

IT-Systeme müssen modular aufgebaut und einfach kombinierbar sein, um den oben genannten Bedingungen zu genügen. Der Ansatz eines solchen „Systems-of-Systems“ erlaubt dabei die Interoperabilität von unabhängig voneinander gestalteten IT-Systemen. So können auch IT-Systeme für das alltägliche Verkehrsmanagement mit solchen interagieren, die für spezielle Sicherheitsherausforderungen ausgelegt sind. Alle Systeme behalten dabei ihre operative Eigenständigkeit und bleiben in der Verantwortung unterschiedlicher Akteure, können aber zugleich Informationen austauschen, um bessere Grundlagen für Entscheidungen in gerade in kritischen Situationen bereit zu stellen. Zugleich sind auf Grundlage solcher modularer IT-Systeme die Realisierung unterschiedlicher organisatorischer Konzepte (Zuständigkeiten und Arbeitsabläufe) sowie die Unterstützung konkreter Betriebsaufgaben und Sicherheitsanliegen möglich. Dies wurde für den geschilderten Untersuchungskontext ausgearbeitet und wird derzeit erprobt, die folgenden Ansätze erscheinen aber über den Einzelfall hinaus übertragbar zu sein.

Die technische (syntaktische) Interoperabilität bildet so die Basis für die inhaltliche (semantische) Interoperabilität, auch über Organisationsgrenzen hinweg. Um eine solche Modularität von IT-Systemen sicherzustellen, muss eine Gesamtarchitektur die notwendigen Schnittstellen und einige gemeinsam zu verwendende Standards und Begriffskonventionen abstimmen [BA2007]. Darüber hinaus sind die Zugriffsrechte zwischen Institutionen ein zentrales Anliegen, das der Klärung bedarf. Letztendlich sind zur Erschließung der Synergiepotenziale abgestimmte Arbeitsabläufe für spezifische Sicherheitsanliegen notwendig, die auf gemeinsamen konzeptionellen Modellen beruhen sollten. Aus diesen vielschichtigen Anforderungen wird deutlich, dass ein solcher Abstimmungsprozess langwierig und komplex werden kann.



## 5.1 Fachliche Schnittstellen

Je klarer gemeinsame fachliche Sicherheitsanliegen formuliert sind, desto einfacher können die erforderlichen IT-Anforderungen definiert werden. Ein abstrakter Klärungsprozess kann hingegen schnell an politischen und organisationalen Grundsätzen oder Partikularinteressen scheitern. Im Projekt VeRSiert wurden daher ausgehend von den definierten Szenarien, Untersuchungsräumen und Rollenmodellen fachliche Schnittstellen zwischen den Bausteinen zur Unterstützung von Einsatzfällen definiert. Anschließend erfolgte die erforderlichen Festlegungen auf IT-Standards und IT-Schnittstellen.

Abbildung 2 benennt zentrale inhaltliche Schnittstellen; beispielsweise werden Zählergebnisse der Videoanalyse an die Simulation übergeben (unterste Zeile der Abbildung). Hierbei erfolgt auf Seiten der Videoanalyse eine Aggregation und Auswertung, z.B. auf 5-Minuten-Intervalle. Die Übertragung kann im einfachsten Falle täglich durch Ablage einer strukturierten Datei an einem definierten Ort erfolgen, oder aktiv für jedes Zählergebnis übermittelt werden. Darüber hinaus wurden zwischen fast allen Bausteinen in beide Richtungen interessante fachliche Berührungspunkte identifiziert werden. Ob ein IT-unterstützter Austausch dabei über zentrale Schnittstellen der Gesamtarchitektur, durch bilaterale Lösungen oder überhaupt nicht erfolgt, ist in einigen Fällen noch offen.

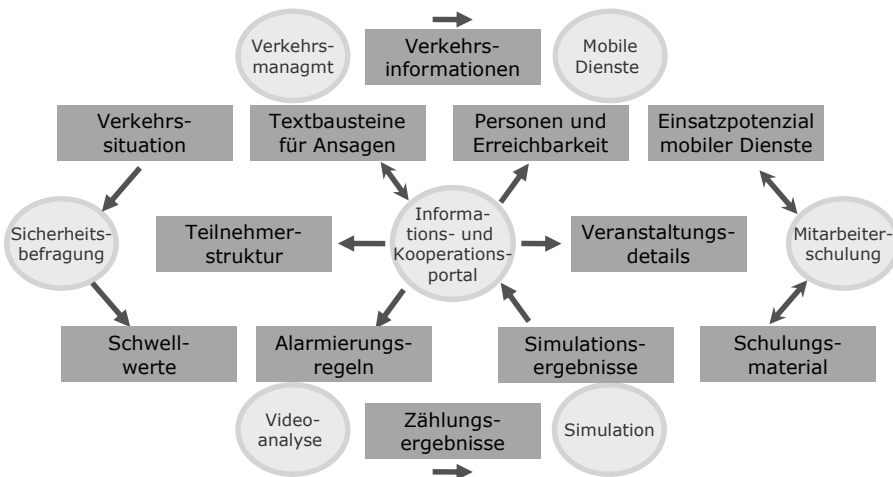


Abb. 2: Fachliche Schnittstellen im Gesamtsystem von VeRSiert

## 5.2 Informationstechnische Standards

Insgesamt ist zu erkennen, dass jeder Baustein fachliche Module anderer Bausteine zur Steigerung seines Wertes verwenden kann. Die Ansätze können teilweise auch rein organisatorisch unterstützt werden. Weitergehend können Inhalte, Funktionalitäten oder Regeln einzelner Plattformen auch informationstechnisch über andere Plattformen zur Verfügung gestellt und so eine durchgängige Unterstützung von Rolleninhabern bei Veranstaltungs- und Ereignisszenarien realisiert werden. Das Gesamtsystem bietet so mehr Funktionalitäten und Komfort als die Summe der einzelnen Plattformen.

Die inhaltlichen Berührungspunkte müssen dafür informationstechnisch realisierbar sein. Die jeweiligen Entwicklungsumgebungen der Plattformen standen jedoch überwiegend schon vor Projektbeginn fest und können durch das Projekt nicht vereinheitlicht werden. Entsprechend wurden für den Austausch im Projekt klare Richtlinien und etablierte Formate definiert, wie Daten oder Informationen von einer Plattform auf eine andere übergeben (materielle Integration) bzw. plattformübergreifend verwendet (virtuelle Integration) werden können. Die Semantik (inhaltlichen Anforderungen) ist zwischen den kommunizierenden Plattformen jeweils bilateral abzustimmen (z.B. Zeitstempel, Ort, Anzahl, ...). Dabei ist neben der gleichen Datenstruktur insbesondere die fachliche Kongruenz zu beachten, also dass gleich benannte Inhalte auch die gleiche Bedeutung haben. Als einheitliche Syntax (Datenstruktur) wird XML angestrebt, da es wegen der Baumstruktur flexibler als einfacher zu erzeugende Datenformate wie csv ist. Der Informationsaustausch soll über offene Protokolle realisiert werden (http, ftp, SOAP).

Als aufruforientierte Übergabeform („pull“) werden XML-RPC (Remote Procedure Calls) oder WeBservices angestrebt, als einfache aktive Übergabeform („push“) von Inhalten können RSS-Feeds verwendet werden; dieser hybride Ansatz wird den bestehenden fachlichen Anforderungen am besten gerecht. Für Video-Sequenzen werden Standardformate wie MPEG eingesetzt, für Audio-Aufnahmen MP3. Bilder werden als JPEG, als Referenz gedachte Pläne und Dokumente als PDF abgelegt. Sofern jeweils bilateral geeignet, können auch die lokalen Interfaces der einzelnen Plattformen verwendet werden. Es können zudem systemweite Software-Agenten eingesetzt werden, die den Datenfluss automatisch überwachen und dann nach vordefinierten Regeln Ereignisse in bestimmten IT-Systemen auslösen. Auch gemeinsame Modellbibliotheken, beispielsweise zur Ablage von Regeln und Funktionen, sind möglich.

Bei der konkreten Klärung von Austauschwegen für die fachlichen Anforderungen ist beispielsweise zu prüfen, ob ein Datenaustausch „in Echtzeit“ oder „zeitversetzt“ erfolgen soll. Für eine weitere Verfolgung der Integrationsoptionen kann auch auf Erfahrungen und Lösungen anderer Kölner Integrationsprojekte zurückgegriffen werden, insbesondere für den Ansatz von „Mobil im Rheinland“ [MR09]. Je nach weiterer Konkretisierung von Geschäfts- und Betreibermodellen ist eine interne Verrechnung von gegenseitig zur Verfügung gestellten Informationen oder Funktionalitäten denkbar, für die dann entsprechende Monitoring- und Auswertungsstandards definiert werden müssten.

## **6 Zusammenfassung und Ausblick**

Dieser Artikel leitet die Anforderungen an eine IT-Architektur her, die für Sicherheits-szenarien bei Großveranstaltungen eine pragmatische Unterstützung der vielfältigen beteiligten Akteure bieten. Diese Konzeption wird derzeit im Rahmen des Projektes VerSiert mit verschiedenen zusammenspielenden IT-Bausteinen erprobt. Wenn ein solcher Ansatz den Status von Demonstratoren verlässt und in den operativen Einsatz übergeht, sind weitere Hürden zu nehmen, angefangen von der Konfiguration von Firewalls bis hin zu möglichen haftungsrechtlichen Fragen bei zunehmender Informations-transparenz. Auch jede einzelne Organisation kann ihre Arbeitsprozesse ggf. mit Hilfe solcher interoperablen IT-Systeme optimieren.

Der Ansatz ist insbesondere dann erfolgreich, wenn er die alltäglichen Aufgaben ebenso unterstützt wie eher hypothetische Krisenszenarien. Zudem könnte sich die Frage nach Kostenaufteilungen zwischen Informationsanbietern und Informationsnutzern stellen. Ggf. haben die Informationsanbieter aber auch ein Interesse an der Nutzung ihres Wissens durch andere Akteure, weil so gemeinsam bessere Handlungsweisen möglich werden. Wenn die Anbieter von IT-Systemen für Verkehrsmanagementaufgaben wie für Sicherheitsfragestellung die notwendigen Schnittstellen unterstützen, ist es für zukünftige Einsatzszenarien einfacher, sie entsprechend der geschilderten Vision von Kooperationen zu realisieren.

## 7 Literatur

- [AN09] Eriksson, E.A.: Briefing for EC/DE workshop: System-of-Systems Demonstration & Experimentation for Mass Transport Security, Sept. 2009 (DEMASST Deliverable 6.1)
- [BA07] Botterell, A. and Addams-Moring, R. (2007) Public Warning in the Networked Age: Open Standards to the rescue?, *Communications of the ACM*, 50, 3, 59-60.
- [BM09] Bundesministerium für Bildung und Forschung (BMBF, Hrsg.): Forschung für die zivile Sicherheit. Schutz von Verkehrsinfrastrukturen, 2009
- [CA09] Cluster Of User Networks in Transport and Energy Relating to Anti-terrorist ACTivities, <http://www.counteract.eu/>
- [DS09] Delle Site, P.; Salucci, M.V.: Urban Transport. Thematic Research Summary. Transport Research Knowledge Center, 2009
- [HE04] Heinze, G. W. (2004) Klassifikation von Events, in H. Dienel and J. Schmithals (Eds.), *Handbuch Eventverkehr - Planung, Gestaltung, Arbeitshilfen*, Erich Schmidt Verlag, Berlin, 25-35.
- [MB09] Model-Based Safety Evaluation of Automation Systems, <http://www.modsafe.eu/>
- [MH07] Manoj, B. S. and Hubenko Baker, A. (2007) Communication Challenges in Emergency Response, *Communications of the ACM*, 50, 3, 51-53.
- [MJ07] Mendonca, D., Jefferson, T. and Harrald, J. (2007) Collaborative Adhocracies and Mix-and-Match Technologies in Emergency Management: Using the emergent interoperability approach to address unanticipated contingencies during emergency response, *Communications of the ACM*, 50, 3, 45-49.
- [MR09] Mobil im Rheinland, <http://www.mobil-im-rheinland.de/>
- [MU09] Modular Urban Guided Rail System, <http://www.modurban.org/>
- [PV08] Projekt VeRSiert (2008) Homepage, <http://www.versiert.info/>, accessed 2009-08-01.
- [RE08] Roßnagel, H., Engelbach, W., Frings, S. and Weisbecker, A. (2008) Mobile Dienste zur Erhöhung der Sicherheit bei Großveranstaltungen, in D. Spath, O. Höß and A. Weisbecker (Eds.), *Stuttgarter Softwaretechnik Forum 2008: Science meets Business*, 2008-11-28, Fraunhofer IRB Verlag, Stuttgart, 91-102.
- [RO08] Rossnagel, H., Engelbach, W., Frings, S.: Ortsbezogene mobile Dienste zur Verbesserung der Sicherheit bei Großveranstaltungen, in J. Roth (Eds.), *Tagungsband 5. Fachgespräch Ortsbezogene Anwendungen und Dienste*, Nürnberg, 2008, S. 35–40
- [TC04] Turoff, M., Chumer, M., Van de Walle, B. and Yao, X. (2004) The Design of a Dynamic Emergency Response Management Information Systems (DERMIS), *Journal of Information Technology Theory and Application (JITTA)*, 5, 4, 1-36.
- [YD05] Yuan, Y. and Detlor, B. (2005) Intelligent Mobile Crisis Response Systems: Systems to help coordinate responder communication and response efforts in order to minimize the threat to human life and damage to property, *Communications of the ACM*, 48, 2, 95-98.

# Rahmenbedingungen für Informationssysteme im Kontext der nicht-polizeilichen Gefahrenabwehr

Marco Plaß, Christina Schäfer

C.I.K.

Universität Paderborn

Pohlweg 47-49

33098 Paderborn

m.plass@cik.uni-paderborn.de

c.schaefer@cik.uni-paderborn.de

**Abstract:** In komplexen Situationen können Einsatzkräfte der nicht-polizeilichen Gefahrenabwehr durch Informationssysteme unterstützt werden. Das Hauptinteresse richtet sich daher auf die Untersuchung von Rahmenbedingungen in diesem Gegenstandsbereich. Die Einflussgrößen auf der Entwicklerseite des Informationssystems und den daraus ableitbaren Verallgemeinerungen für die Konzeption eines solchen Systems fließen ebenfalls in die Betrachtung ein.

## 1 Einleitung

In der nicht-polizeilichen Gefahrenabwehr werden zunehmend Informationssysteme verwendet, um den Einsatzkräften eine Hilfestellung innerhalb der Organisation und Durchführung eines Einsatzes zu bieten. Durch Bereitstellen von relevanten Informationen kann ein effizienteres und effektiveres Vorgehen im konkreten Einsatzfall erreicht werden.

Ein Informationssystem ist dabei „eine Sammlung von Informationen oder Daten zu einer vorgegebenen Thematik, die aufgrund eines eindeutigen Ordnungsschemas dem Benutzer einen schnellen Zugriff erlaubt.“[Li99] Hier werden allerdings nur Informationssysteme betrachtet, die sich durch eine informationstechnologische Komponente kennzeichnen, also liegt ein Telefonbuch, welches durchaus auch als Informationssystem gesehen werden kann, nicht im Zentrum dieser Betrachtung. Darüber hinaus grenzt der Kontext, nicht-polizeiliche Gefahrenabwehr, den Anwendungsfall ein. Außerdem muss es klar von einem Kommunikationssystem abgegrenzt werden, da nicht die Unterstützung von Kommunikation zwischen Nutzern im Vordergrund steht, sondern die Informationsbereitstellung.

Um den Rahmen der Nutzung eines entsprechenden Informationssystems ganzheitlich zu erfassen muss der Fokus auf den Menschen, dessen Aufgaben, unter Umständen Verwendung findende IT und den organisatorischen Kontext gerichtet werden [BJ07].

Im Zusammenhang der anschließenden Betrachtung wird der Mensch durch Mitglieder von Organisationen der nicht-polizeilichen Gefahrenabwehr repräsentiert. Da als Kontext das Szenario einer größeren Schadenslage angenommen wird, liegt die Aufgabe der Nutzer im Retten von gefährdeten Personen und der Wiederherstellung einer zuvor herrschenden Ordnung. Die Schadenslage wird dabei von verschiedenen Faktoren gekennzeichnet. So kann die Anzahl der Verletzten, die mögliche Gefährdung der Bevölkerung durch giftige, ätzende Stoffe oder die Schädigung von bedeutender Infrastruktur zu der Kategorisierung als größere Schadenslage führen [NDSL04].

Im Folgenden werden die Rahmenbedingungen untersucht, welche auf die Situation einer größeren Schadenslage, den darin handelnden Akteur, dem jeweiligen Informationssystem und seinem Entwickler einwirken. Die daraus entstehenden Wechselwirkungen bauen ein Spannungsfeld zwischen den Einflussgrößen, welche auf die entsprechenden Einsatzkräfte wirken und dem Wunsch nach Wirtschaftlichkeit seitens des Entwicklers eines Informationssystems auf. Insbesondere für den Entwickler und dem Benutzer eines Informationssystems ist eine solche Betrachtung von Interesse, da zum einen eine Erleichterung in der Entwicklung und zum anderen eine Verbesserung der Nutzungssituation erzielt werden kann. Weitere Stakeholder sollen an dieser Stelle keine Berücksichtigung finden.

## **2 Analyse der Rahmenbedingungen in einer größeren Schadenslage**

Wie in anderen Bereichen der Softwareentwicklung ist die Kenntnis des Gegenstandsbereichs, in dem Nutzer einer Software handeln, bei der Planung, der Analyse und dem Entwurf essentiell wichtig. Da sich der Gegenstandsbereich in der nicht-polizeilichen Gefahrenabwehr von anderen Bereichen deutlich unterscheidet, lohnt eine Betrachtung im Hinblick auf die Rahmenbedingungen, unter denen Nutzer agieren, und die bei der Entwicklung berücksichtigt werden müssen. Hieraus lassen sich Anforderungen an Software im Allgemeinen und auf die Informationssysteme, die im Einsatz Verwendung finden, im Speziellen ableiten.

### **2.1 Rahmenbedingungen bezüglich des Gegenstandsbereichs**

Im Gegenstandsbereich beschreiben beispielsweise Untersuchungen des Einsatzes während und nach dem Hurrikan Katrina in den USA die Bedingungen für die Akteure im Einsatz als komplex und dynamisch [BGXR08]. Weitere Betrachtungen bezeichnen diese Art Schadensereignisse als kritische Situationen, die sich für Individuen als komplex darstellen [z.B. Doer09]. Dabei ist Komplexität in einer Situation immer personen gebunden. Es existiert keine komplexe Situation, die aus personenunabhängiger Sicht definierbar ist. Diese Situationen sind über Eigenschaften charakterisierbar.

Untersuchungen aus unterschiedlichen Bereichen und Perspektiven, wie Aufgaben im Krisenmanagement für den Bevölkerungsschutz [Fran02], Krisenbewältigungsaufgaben [BGXR08], Planspielen [Doer09], und Entscheidungen von Feuerwehreinsatzkräften [Klei03] zeigen eine wiederkehrende Untermenge von gemeinsamen Eigenschaften.

Diese sind in zwei Kategorien aufteilbar. Es sind Eigenschaften, die für alle Menschen gleich gelten und andere, die von Mensch zu Mensch verschieden sind.

Eine Situation ist für alle Menschen komplex, wenn

- eine große Anzahl von Merkmalen (z.B. Objekten oder Personen) existiert (Vielzahl),
- die Anzahl der Zustände, die diese Merkmale einnehmen können, hoch ist (Varietät),
- die Merkmale miteinander in hohem Maße verknüpft sind (Rückkopplung) und
- die Merkmale sich in Abhängigkeit von der Zeit verändern (Dynamik).

Die von individuellen Merkmalen wie z.B. Erfahrungen abhängigen Eigenschaften prägen die Komplexität von Situationen, wenn

- die Vielzahl von verfügbaren Informationen nicht verarbeitet werden kann (Informationsüberfluss) [Doer09, SBH05],
- zum Bewerten von Situationen zu wenig Informationen in zu geringer Vollständigkeit, Korrektheit oder Aktualität verfügbar sind (Informationsmenge und -beschaffenheit) [Doer09, SBH05, BL07, BGXR08, Fran02],
- eine große Anzahl von Zielen existieren unter denen Zielkonflikte herrschen [Stroh07],
- Zusammenhänge von Merkmalen nicht bekannt (Unkenntnis) oder nicht durchschaubar (Intransparenz) sind [Doer09, Fran02],
- eine Vereinfachung der Verknüpfung von Merkmalen durch Linearisierung nicht gültig ist (Nichtlinearität) [Doer09] und [Hofi07]
- die Auswirkungen von Handlungen bedeutsam sind (Bedeutsamkeit) [BGXR08, Fran02].

Die genannten Merkmale charakterisieren Situationen, in denen Akteure der nicht-polizeilichen Gefahrenabwehr handeln müssen. Zum Verständnis dieser Handlungen wurden Modelle zur Abstraktion und Vereinfachung entwickelt. Diese folgen dabei zwei wesentlichen Strömungen.

Dörner geht davon aus, dass Handlungen einem bewussten, zielgerichteten Ablauf folgen, in dem die Schritte Situationsanalyse, Modellbildung, Zielbildung, Planung/Entscheidung und Ausführung nacheinander ausgeführt und in einem mehr oder weniger vollständigen Kreislauf wiederholt werden [Doer09]. Diesem Modell folgt beispielsweise die Ausbildung von Führungskräften der Feuerwehr [AFW99]. Im Modell von Klein [Klei03] folgen Entscheidungen für Handlungen einer Erkennung typischer Situationen mittels Intuition und Abwägung von Handlungsoptionen durch mentale Simulation. Bewusste oder unbewusste Ziele der Handelnden werden nicht betrachtet.

Der von Klein beschriebene, unbewusste Prozess kann als Situationsanalyse/Modellbildung Dörners (durch Intuition abgekürzt) und folgend Planung/Entscheidung (durch Abbruch mentaler Simulation nach der ersten brauchbaren Lösung) betrachtet werden. Die bewusste Zielbildung entfällt, da schon unbewusste Ziele vorhanden sind. Diese können den eigentlichen objektiven Zielen sogar entgegenstehen, aber aus Sicht des Akteurs rational sein [Stro02].

Die besonderen Situationen, in denen Akteure in der nicht-polizeilichen Gefahrenabwehr handeln, lassen für die einzelnen Handlungsschritte sehr wenig Zeit. Dies erhöht zwar nicht die Komplexität der Situation in den oben genannten Merkmalen, doch es fördert das Verhalten zum frühzeitigen Abbruch eines Handlungsschritts beim Vorhandensein eines ersten praktikablen Ergebnisses. Das Handeln sollte der Idealvorstellung einer ressourcenorientierten Abwägung vorhandener Risiken mit den vorhandenen Informationen unter Zeitdruck [Weth02] folgen. Allerdings haben verschiedene Faktoren Anteil daran, dass diese Idealvorstellung nicht immer erreicht wird.

Zusammenfassend ist für die Entwicklung von Informationssystemen wichtig, dass Nutzer wegen ihrer besonderen Einsatzsituation und großen Zeitdrucks unter erheblichem psychischen Druck stehen. Dieser wird durch das Agieren unter erschwerten Umweltbedingungen (die auch eine physische Belastung für Einsatzkraft und Endgerät erzeugen) weiter erhöht. Hinzu kommt die Gefahr für das eigene Leben, möglicherweise schwere Folgen von Fehlhandlungen und daraus folgende rechtliche Situation. Um dem Druck standzuhalten sind individuelle Fähigkeiten zur Komplexitätsreduktion (z. B. Superierung) [Doer09], Intuition [Klei03], Vertrauen in die Fähigkeiten anderer Akteure und Erfahrungen [Klei03, Weth02, Um01] hilfreich.

Werden Akteure in komplexe Situationen aus Sicht eines zu entwickelnden Informationssystems, welches in diesen Situationen eingesetzt werden soll, betrachtet, so kann das generelle Ziel einer Unterstützung der Akteure, die nun als Nutzer betrachtet werden, durch Komplexitätsreduktion erreicht werden.

Einige der oben genannten Merkmale kommen für die Unterstützung durch ein Informationssystem besonders in Frage; andere können durch die Nutzung eines Informationssystems nur bedingt beeinflusst werden. Dabei können die Merkmale, die für alle Menschen Komplexität ausmachen, generell durch Erhöhung der Transparenz einer Situation, z.B. durch Visualisierung von Zusammenhängen, erreicht werden. Eine konkretere Aussage ist für die ganze Kategorie Informationssysteme schwer zu treffen.

Informationssysteme können allerdings auf die Merkmale Informationsüberfluss und Informationsmenge und –beschaffenheit einen entscheidenden Einfluss haben. Beim Entwurf ist darauf zu achten, dass die Informationen, die ein Informationssystem bereitstellt, eine möglichst hohe Informationsqualität [RKMP07] in den Dimensionen Relevanz, angemessener Umfang, Vollständigkeit, Fehlerfreiheit und Aktualität aufweisen.

Neben der Beschaffenheit von Informationen, gefolgt aus der Komplexität von Situationen, können weitere Anforderungen aus der Nutzungssituation abgeleitet werden. Der in jedem Handlungsschritt vorherrschende Zeitdruck erfordert einen schnellen Zugang zu Informationen aus einem Informationssystem. Aus der Strategie, bei Verfügbarkeit eines ersten befriedigenden Ergebnisses den aktuellen Handlungsschritt abzubrechen, um unter Zeitdruck handlungsfähig zu bleiben, folgt, dass Informationen von im Einsatz Verwendung findenden Informationssystemen in geeigneter Reihenfolge darzustellen sind.

Es ist ein hohes Maß an Usability erforderlich und somit ist der Gestaltung von Dialogkomponenten besondere Aufmerksamkeit zu schenken.

Darüber hinaus ist die Kenntnis des Stellenwerts von Erfahrungen für die Entwicklung von Informationssystemen wichtig. Die Erfahrungen der Akteure und der Aufbau von individuellen Fähigkeiten zur Komplexitätsreduktion werden durch Trainings (z.B. in Landesfeuerwehrschulen) in komplexen Situationen gefördert. Vom kleinsten gemeinsamen Erfahrungsschatz kann innerhalb einer Rolle ausgegangen werden, da für das Bekleiden einer Rolle die Teilnahme an entsprechenden Trainings Voraussetzung ist. Soll ein Informationssystem durch Informationen die individuell vorhandene Situation reduzieren helfen, sind minimal die Nutzer in verschiedene Rollen aufzuteilen, für die jeweils unterschiedliche Informationen präsentiert werden. Der wünschenswerteste Ansatz wäre sogar eine auf individuelle Bedürfnisse angepasste Informationspräsentation.

Die hier beschriebenen Anforderungen an Informationssysteme aus Sicht der Psychologie in komplexen Situationen können nicht als vollständig angesehen werden. Einige weitere Anforderungen, die für Gruppen-Kommunikationssysteme gelten und aus jahrelangen Erfahrungen aus dem amerikanischen Office of Emergency Preparedness (OEP) ermittelt wurden, gelten für Informationssysteme gleichermaßen. Besonders die Prämissen Informationsfokus („have no tolerance and time for things unrelated“), Krisengedächtnis („Learning and understanding what actually happened before, during and after a crisis is extremely important [...]“) und Information Validity and Timeliness („[...] by supplying the best possible up-to-date information [...]“) [TCWY04] sind hier zu nennen und lassen sich aus den oben dargelegten Gründen bestätigen.

## **2.2 Rahmenbedingungen aus Sicht des Entwicklers**

Während sich Anforderungen durch die Einwirkung einer hinreichend großen Anzahl an Rahmenbedingungen auf die Situation und den darin agierenden Nutzer ergeben, wird im Folgenden die Problematik aus Entwicklersicht eines Informationssystems geprüft.

Hier stellt die Heterogenität von unterschiedlichen, im Kontext enthaltenen Aspekten eine entscheidende Einflussgröße dar.

### **Heterogenität und Wirtschaftlichkeit**

Zum einen sind die beteiligten Organisationen an dem skizzierten Szenario einer größeren Schadenslage sehr verschieden. Auch wenn das übergeordnete Ziel, das Retten und Sichern der Lage identisch ist, so ist bei differenzierter Betrachtung die Aufgabenstellung von zum Beispiel Feuerwehr, Rettungsdienst und Technischem Hilfswerk nicht identisch. Alle Organisationen sind mit unterschiedlichen Rechten und Pflichten ausgestattet, müssen aber dennoch in der entsprechenden Krisensituation zu einem Team zusammensuchen, um das Gesamtziel zu erreichen.

Zum anderen wird die Heterogenität nicht nur durch die Vielzahl der beteiligten Organisationen hergestellt, sondern ebenfalls durch die voneinander abweichende interne Strukturierung der jeweiligen Organisationen.



Dies kann am Beispiel der Feuerwehr näher erläutert werden. Nach §9 des Feuerwehrgesetz Nordrhein Westfalen ist eine Gliederung in Berufsfeuerwehr, Freiwilliger Feuerwehr, Werksfeuerwehr, Betriebsfeuerwehr und Pflichtfeuerwehr vorgesehen, wobei dann unterschiedliche Aus- und Weiterbildung, Rechtsgrundlagen und Ziele festzustellen sind [FSHG97]. Die Strukturierung ist zwar bundesweit ähnlich, aber wie an dem genannten Gesetz zu sehen ist, wird sie auf Landesebene entschieden. Die Feuerwehrdienstvorschriften geben ebenfalls Anhaltspunkte zur Umsetzung einer praxisnahen Ausbildung in Organisation und Ablauf eines Einsatzes, aber die Einführung und Durchsetzung wird gleichermaßen von jedem Bundesland selbst bestimmt [LFWS73].

Im Weiteren wird berücksichtigt, dass verschiedene Rollen innerhalb der Feuerwehr in einem Einsatz vorgesehen sind, so dass die vorherrschende Heterogenität noch gesteigert werden kann. Ein Angriffstrupp, der eine operative Einheit repräsentiert, hat einen anderen Informationsbedarf, insbesondere in Art und Aggregationsgrad, als der Einsatzleiter, der übergeordnete organisatorische Aufgaben zu erfüllen hat.

Daraus resultiert eine Vielzahl von IT-Systemen, um jeder Situation und jedem Nutzer gerecht zu werden. Dies kann von einer automatischen Brandmeldeanlage bis zu einem Stabsunterstützungssystem reichen.

Welche Information relevant ist, entscheidet der Nutzer und so sind im Zusammenhang der nicht-polizeilichen Gefahrenabwehr gänzlich verschiedene Datenarten von Bedeutung. Es können georeferenzierte Sensordaten genauso wie statistische oder personenbezogene Daten von Interesse sein. Diese werden unter Umständen von unterschiedlichen Ämtern, wie dem entsprechenden Kataster- oder Einwohnermeldeamt vorgehalten und müssen in ein Informationssystem integriert werden.

Mit diesen Ausführungen ist gezeigt worden, dass die Heterogenität ein nicht zu vernachlässigendes Problem darstellt und auf verschiedenen Ebenen zu finden ist, den Organisationen, den Rollen innerhalb der Organisation und dem Informationsbedarf des Einzelnen in seiner Rolle. Gerade im Hinblick auf den verständlichen Wunsch nach Wirtschaftlichkeit bei der Entwicklung eines Systems, ist auf die mögliche Vermarktung an hinreichend viele Organisationen der nicht-polizeilichen Gefahrenabwehr zu achten.

Ziel des Entwicklers ist es, mit einem System eine möglichst breite Masse anzusprechen. Wird zu stark abstrahiert, ist die Gefahr groß, ein System anzubieten, das auf keinen Endanwender wirklich zugeschnitten ist. Mit Berücksichtigung dieser Problematik können dennoch Verallgemeinerungen für eine Softwarearchitektur getroffen werden, die zwar eine Änderung der Inhalte erforderlich macht, aber Konzepte für unterschiedliche Organisationen und Rollen trotzdem adaptierbar sind.

Im Anschluss werden daher aus den beschriebenen Rahmenbedingungen, die entstehenden Anforderungen, abgeleitet.

## Resultierende Anforderungen an die Architektur eines Informationssystems

Um die Wirtschaftlichkeit in Bezug auf die Vermarktung und den Verkauf von Informationssystemen im beschriebenen Kontext gewährleisten zu können, muss eine flexible Gestaltung der Softwarearchitektur ermöglicht werden.

Durch die beschriebene Heterogenität im Gegenstandsbereich werden verschiedene Softwarekomponenten die sich ggf. sogar auf verschiedenen Computersystemen befinden in einem verteilten System zusammengeführt. Daher erscheint ein modulares Architekturkonzept absolut notwendig, um der Heterogenität adäquat begegnen zu können. Es muss die Option bestehen, Komponenten dynamisch dem System hinzuzufügen oder wegzulassen, je nachdem wie sich die Bedürfnisse des Nutzers darstellen. Damit ergibt sich die Anforderung einer losen Kopplung der einzelnen Systemkomponenten, um sinnvoll auf die Rahmenbedingungen reagieren zu können. Dadurch werden Abhängigkeiten im System reduziert und „je weniger Abhängigkeiten es gibt, desto geringer sind die Auswirkungen von Änderungen [...]“ [Jo09].

Für die Umsetzung eines Systems mit loser Kopplung muss eine Kommunikationskomponente geplant werden, die die anderen Komponenten miteinander verbindet und ein zuverlässiges, intelligentes Routing von Anfragen, Ergebnissen und auftretenden Events leisten kann. Dabei muss klar definiert sein, welche Ereignisse von dem System erkennbar sind und einen signifikanten Einfluss auf den Informationsbedarf des Nutzers nehmen können. In dem Projekt OrGaMIR<sup>1</sup> kommt daher ein Enterprise Service Bus zum Einsatz, der eine lose Kopplung der Komponenten durch Web Service Schnittstellen sinnvoll ermöglicht. Die Frage nach der Adressierung bzw. Benachrichtigung bei Events kann auf unterschiedliche Weise gelöst werden. Zum einen wird ein Publish/Subscribe - Verfahren realisiert, damit sich Komponenten auf gewünschte Nachrichten und Informationen registrieren können.

Darüber hinaus werden weitere Routing – Konzepte umgesetzt, die eine inhaltsbezogene Zuweisung von Nachrichten vorsieht. In [TCVX04] wurde „Content as Address“ als ein Design Prinzip vorgestellt, wobei die Ausrichtung an dieser Stelle auf die Kommunikation zwischen beteiligten Nutzern in der Krise ausgelegt wurde. Eine Übertragung auf die Kommunikation von Systemkomponenten ist folgerichtig, da jeder Teil des Systems eine bestimmte Aufgabe zu erfüllen hat und eine bestimmte Art von Information benötigt.

Nicht nur innerhalb der Gesamtsystemarchitektur ist die Anforderung nach einer losen Kopplung festzustellen. Dies lässt sich auch auf anderen Ebenen der Architektur ableiten.

Wenn ein Informationssystem mehr leisten soll als ein Nachschlagewerk wird eine Komponente, die eine Informationsaggregation durchführt, benötigt. Auf diese Weise werden zum Beispiel aus Sensordaten Informationen generiert, die den Nutzwert dieser Daten erhöhen.

---

<sup>1</sup> Die im Folgenden genannten Projekte ORGAMIR und AirShield werden gefördert durch das BMBF im Programm „Forschung für die zivile Sicherheit“, einem Teil der HighTech-Strategie der Bundesregierung; das Projekt Mobis Pro wird durch das BMWi gefördert.

Da, wie zuvor schon erwähnt, der Informationsbedarf der Nutzer stark divergiert, ist ein Konzept, das durch leichte Anpassung die Herleitung unterschiedlicher Informationen für verschiedene an der Situation beteiligte Rollen sichern kann, erforderlich. Auch hier ist eine Realisierung durch die Verwendung des Konzepts von Service Orientierten Architekturen denkbar. Einzelne atomare Operationen können von Web Services erfüllt werden und durch eine folgerichtige Orchestrierung der Services werden für den Menschen nachvollziehbare und nutzbare Informationen aus Sensordaten generiert. Durch die Verwendung von Web Services in einer Informationsverdichtungskomponente ist gleichermaßen eine Schnittstelle geschaffen, um heterogene Daten in das System zu integrieren. Die Notwendigkeit von vielen unterschiedlichen Daten ist durch die Analyse der Rahmenbedingungen hervorgegangen.

### **3 Zusammenfassung und Ausblick**

Es konnte gezeigt werden, dass eine Vielzahl an Rahmenbedingungen sowohl im Gegenstandsbereich der nicht-polizeilichen Gefahrenabwehr, als auch auf Seiten des Entwicklers festzuhalten sind. Diese wirken sich maßgeblich auf die Konzipierung eines Informationssystems für diesen Anwendungsfall aus.

Als Anforderungen konnten eine gute Usability erkannt werden, da die Akteure unter hohem psychischem Druck stehen und durch eine intuitive und selbsterklärende Benutzungsschnittstelle entlastet werden können. Um das System im hohen Maße wirtschaftlich zu gestalten, wurden eine lose Kopplung der einzelnen Systemkomponenten und ein dynamisches Routing zwischen ihnen als Anforderungen an die Architektur ersichtlich.

Zurzeit werden die vorgestellten Konzepte und Ideen in unterschiedlichen Projekten umgesetzt. In ORGAMIR wird die Gesamtarchitektur von Informationsverdichtung, Kommunikationsmanager, Relevanzengine (Rollenkonzept) und GUI realisiert, während im Projekt AirShield das Konzept der modularen, informationsverdichtenden Komponente in einem anderen Zusammenhang verwendet werden. Sowohl das Projekt ORGAMIR als auch das Projekt Mobis Pro nutzen denselben Kommunikationsmanager, um die Kommunikation in einem lose gekoppelten System zu verwirklichen.

Im Rahmen einer Evaluation ist geplant, einige der getroffenen Aussagen bezüglich der Anforderungen an ein Informationssystem zu verifizieren. Dabei wird die Usability von Informationssystemen aus den genannten Projekten in Feldtests mit Feuerwehren untersucht. Besonderes Augenmerk wird auf die Relevanz und den angemessenen Umfang von Informationen gelegt. Für weitere Projekte muss ebenfalls die Übertragbarkeit auf die polizeiliche Gefahrenabwehr geprüft werden.

### **Literaturverzeichnis**

[AFW99] Ausschuss Feuerwehrangelegenheiten (AFW) des Arbeitskreises V der ständigen Konferenz der Innenminister und - Senatoren der Länder (1999): Feuerwehr Dienstvorschrift 100 (FwDV 100), vom 10.03.99.

- [BK07] Bächle, M; Kolb,A.: Einführung in die Wirtschaftswissenschaften, Oldenbourg, München, 2007
- [BGXR08] Becerra-Fernandez, I; Gudi, A.; Xia, Weidong; R., Jose (2008): Task Characteristics, Knowledge Sharing and Integration, and Emergency Management Performance: Research Agenda and Challenges. In: Fiedrich, F.; van der Walle, B. (Hg.): Proceeding of the 5th International ISCRAM Conference. Tilburg University, Netherlands, S. 88–92.
- [BL07] Badke-Schaub, P.; Lorei, C. (2007): Führung und Entscheiden. In: Strohschneider, S. (Hg.): Entscheiden in kritischen Situationen. 2. Aufl. Frankfurt a.M.: Verl. für Polizeiwissenschaft, S. 153–167.
- [Doer09] Dörner, D. (2009): Die Logik des Misslingens. Strategisches Denken in komplexen Situationen. 8. Aufl. Reinbek bei Hamburg: Rowohlt (Rororo).
- [Fran02] Franke, D. (2002): Krisenmanagement ist lernbar. In: Bevölkerungsschutz. Magazin für Zivil- und Katastrophenschutz, H. 4, S. 17–20.
- [FSHG97] Land Nordrhein-Westfalen: Gesetz über den Feuerschutz und die Hilfeleistung. FSHG, vom 10.2.1997.
- [Hof07] Hofinger, G. (2007): Fehler und Fallen beim Entscheiden in kritischen Situationen. In: Strohschneider, S. (Hg.): Entscheiden in kritischen Situationen. 2. Aufl. Frankfurt a.M.: Verl. für Polizeiwissenschaft, S. 115–136.
- [Jo09] Josuttis, N.: SOA in der Praxis – System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, Heidelberg, 2009.
- [Klei03] Klein, G. (2003): Natürliche Entscheidungsprozesse. Über die "Quellen der Macht", die unsere Entscheidungen lenken. Paderborn: Junfermann.
- [LFW573] Erlass betreffend Einführung bundeseinheitlicher Feuerwehr-Dienstvorschriften (FwDV), vom 20.12.1973
- [Li99] Linder, W.: Geo-Informationssysteme, Springer-Verlag, Berlin Heidelberg New York, 1999
- [NDSL04] Niedersächsischer Landtag: Drucksache 15/974, „Kompetenzzentrum Großschadenslage“ - Katastrophenschutz in Niedersachsen auf neuen Wegen oder nur unter neuem Namen?, 2004.
- [RKMPS07] Rohweder, J. P.; Kasten, G.; Malzahn, D.; Piro, A.; Schmid, J. (2007): Informationsqualität – Definitionen, Dimensionen und Begriffe. Herausgegeben von Projektgruppe „Normen und Standards“ Deutsche Ges. für Informationsqualität.
- [Stro02] Strohschneider, S. (2002): Die Aufrechterhaltung der Handlungsfähigkeit. In: Strohschneider, S. (Hg.): Ja, mach nur einen Plan. Pannen und Fehlschläge - Ursachen, Beispiele, Lösungen. 2., vollst. überarb., erw. und aktualisierte Aufl. Bern: Huber (Psychologie Forschung), S. 36–50.
- [SBH05] St.Pierre, M.; Buerschaper, C.; Hofinger, G. (2005): Notfallmanagement. Human Factors in der Akutmedizin. Berlin, Heidelberg: Springer Medizin Verlag Heidelberg (Springer-11773 /Dig. Serial)].
- [TCVX04] Turoff, M.; Chumer, M.; van der Walle, B.; Yao, X. (2004): The Design of a Dynamic Emergency Response Management Information System (DERMIS). In: The Journal of Information Technology Theory and Application (JITTA), Jg. 5, H. 4, S. 1–35.
- [UM01] Ungerer, D.; Morgenroth, U. (2001): Analyse des menschlichen Fehlverhaltens in Gefahrensituationen. Empfehlungen für die Ausbildung. ISSN 0343-5164. In: Schriftenreihe der Schutzkommission beim Bundesminister des Innern herausgegeben vom Bundesverwaltungsamt – Zentralstelle für Zivilschutz – im Auftrag des BMI, H. 43.
- [Weth02] Weth, R. von der (2002): Risikoabwägung und Prozesssteuerung in kritischen Situationen. In: Strohschneider, Stefan (Hg.): Ja, mach nur einen Plan. Pannen und Fehlschläge - Ursachen, Beispiele, Lösungen. 2., vollst. überarb., erw. und aktualisierte Aufl. Bern: Huber (Psychologie Forschung), S. 41–54.



# Architektur eines prozessunterstützenden Softwaresystems für den Rettungsdiensteinsatz bei einem Massenansturz von Verletzten

Carsten Wirth, Dirk Roscher,  
Paul Zernicke  
TU-Berlin, DAI-Labor  
Sekretariat TEL 14  
Ernst-Reuter-Platz 7  
10587 Berlin  
{carsten.wirth, paul.zernicke,  
dirk.roscher}@dai-labor.de

Dr. med. Martin Schultz, Christine  
Carius-Düssel  
Charité – Universitätsmedizin Berlin  
Telemedizinzentrum Charité TMCC  
Charitéplatz 1  
10117 Berlin  
{martin.schultz, christine.carius-  
duessel}@charite.de

**Abstract:** Das Einbringen von Innovationen aus dem Bereich der Informationstechnologie in die Prozesse des Rettungsdienstes, mit dem Ziel der Verbesserung des Einsatzserfolges, stellt eine aktuelle Herausforderung dar. Dieser Herausforderungen nimmt sich das Forschungsprojekt ALARM an, welches als Zielsetzung hat, einen Lösungsansatz für den Gesamtprozess bei einem Massenansturz von Verletzten zu konzipieren und zu evaluieren. Es werden die grundlegenden Anforderungen dieses Systems beschrieben und der bisherige Architektorentwurf für das prozessunterstützende Softwaresystem vorgestellt. Im Mittelpunkt stehen dabei der verteilte modulare Plattformansatz, welcher sich auf die Bereiche des strategischen und lokalen Einsatzmanagements erstreckt, sowie die technischen Anforderungen, die sich hinsichtlich der Kommunikation der Komponenten ergeben.

## 1 Einleitung

In der Forschung und Entwicklung im Bereich Rettungswesen und Katastrophenschutz wurden gemessen an der Anzahl abgeschlossener Forschungsprojekte und deren Innovationen im letzten Jahrzehnt wesentliche Fortschritte erzielt. Die Realisierung der von Herrmann skizzierten Vision eines durchgängig und umfassend IT-gestützten Notfalleinsatzes [HE97] rückt damit näher. Die reale Integration der Innovationen in den alltäglichen Rettungsdienst beschränkt sich aber bisher auf vereinzelte kleinere Systeme sowie auf Feldtests mit neuartiger Technologie, so dass sich die Prozesse und Abläufe im heutigen Rettungswesen bisher nur unwesentlich verändert haben. Weiterhin zeigt die Vielzahl an neuen bewilligten Forschungsprojekten, im Themenfeld „Schutz und

Rettung von Menschen“ des Bundesministeriums für Bildung und Forschung<sup>1</sup> sind dieses alleine 15 Projekte, einen hohen Bedarf an weiteren Innovationen im Forschungsgebiet. Eines der in diesem Rahmen bewilligten Forschungsprojekte ist das Projekt ALARM<sup>2</sup>, welches als Zielsetzung hat, einen Lösungsansatz für den Gesamtprozess (siehe Abbildung 1) bei einem Massenansturm von Verletzten (MANV) zu konzipieren und zu evaluieren. Konträr zur Herangehensweise für einzelne Einsatzaspekte Lösungen zu erforschen, bietet der Alarm-Ansatz, eine umfassende Lösung für den Gesamtprozess zu finden, wesentliche Vorteile. So wird einer Entwicklung von Insellösungen, welche aufgrund von fehlenden Standards nicht kompatibel sind, vorgebeugt. Weiterhin ist eine Verbesserung des Einsatzerfolges nur durch Erfolgsindikatoren messbar, welche nicht nur gesondert auf einzelne Prozessschritte anzuwenden sind, sondern vielmehr im Kontext des Gesamtprozesses erhoben und betrachtet werden müssen. Ein Gesamtsystem ermöglicht die Forschung an den verschiedensten Einflussfaktoren und lässt eine Gewichtung hinsichtlich des Gesamteinsatzerfolges zu. In diesem Beitrag stellen wir den aktuellen Stand der Konzeption des ALARM-Systems (Abschnitt 4) vor, der einerseits auf den Erkenntnissen des aktuellen Stands der Technik (Abschnitt 2) und andererseits aus den erhobenen bzw. abgeleiteten Systemanforderungen (Abschnitt 3) begründet ist. Abschließend erfolgen eine kritische Betrachtung und die Darstellung des weiteren Vorgehens im Projekt (Abschnitt 5).

## 2 Stand der Technik

Eine Recherche im Bereich IT für das Rettungswesen offenbart eine Vielzahl an Forschungsprojekten, die sich mit unterschiedlichen Aspekten der Einsatzunterstützung bei Notfällen mit Relevanz für MANV-Lagen auseinandersetzen bzw. gesetzt haben. Wesentliche Aspekte sind die Unterstützung des Triageprozesses[GA06], welcher auch durch neuartige Sensorik und Sensornetzwerktechnologien[MA10] optimiert werden kann, sowie die generelle Unterstützung der Einsatzkräfte durch mobile Dienste, wie sie z.B. xMotion[XM10] zeigt. Ein zweiter großer Bereich sind Assistenzsysteme wie NOAH[HE97] und das Rettungsassistenzsystem TU Clausthal[RE10], welche die Erfassung, Weiterleitung, und Auswertung von Patientendaten optimieren. Den dritten Schwerpunkt bilden Telemedizinssysteme, die dem Rettungsdienstpersonal vor Ort[SK09][ZI06] oder beim Transport im Rettungswagen[ST08] zur Verfügung stehen und damit den Behandlungserfolg bei einem MANV verbessern. Zusätzliche Unterstützung bieten Systeme, die gezielt Informationen zum Einsatzgeschehen und – vorgehen bereitstellen und dabei innovativ vorhandene Kommunikationstechnologien benutzen[LO06][SA10]. Andere Projekte demonstrieren die Möglichkeiten neuartiger Darstellungen von Lageinformationen und darauf basierender Entscheidungsunterstützung, z.B. beim Katastrophenmanagement[ER10] oder bei

---

<sup>1</sup>Bewilligte Projekte aus dem Themenfeld "Schutz und Rettung von Menschen":

<http://www.bmbf.de/de/13091.php>

<sup>2</sup>gefördert durch das Bundesministeriums für Bildung und Forschung; Förderkennzeichen 13N10106 bis 13N10112, ALARM Projektseite: <http://www.alarm-projekt.de/>

Zugunfällen in Tunneln[OR10]. Bei großen Schadenslagen, dieses können auch MANV mit einer entsprechend hohen Anzahl an Verletzten sein, sind mehrere Organisationen an der Bewältigung des Vorfalls beteiligt. Daher wird in Projekten wie Oasis[OA10] und Orchestra[OR08] eine Kollaborationsunterstützung für die Akteure fokussiert, um damit den Einsatzerfolg entscheidend zu erhöhen.

Die vorgestellten Projekte bzw. Aspekte sind sehr hilfreich und richtungweisend für die Konzeptionen des ALARM-Systems. Es gibt Parallelen zu den zu konzipierenden Diensten unseres Systems wie z.B. in den Bereichen Triage, medizinische Versorgungsunterstützung durch Telemedizin, Lageinformation, Entscheidungs- und Kommunikationsunterstützung. Allerdings sind die bestehenden Lösungen durch ihren spezifischen Projektkontext einerseits nicht zwingend für die speziellen Anforderungen in einem MANV-Szenario geeignet und betrachten andererseits durch den begrenzten Rahmen nur Teilaspekte des Gesamtprozesses. Eine wesentliche Aufgabe in ALARM ist es daher bestehenden Ansätze in einen einheitlichen Rahmen bezüglich des Gesamtprozesses zu integrieren und hinsichtlich der Anforderungen eines MANV-Szenarios zu adaptieren – dieses umfasst auch Erarbeitung neuer problemspezifischer Lösungsansätze. Die allgemeinen und spezifischen Anforderungen an das System und dessen Komponenten werden im nachfolgenden Abschnitt präsentiert.

### **3 Technische Anforderungen an das System**

Das zu entwickelnde System hat als oberstes Ziel den Gesamtprozess bei der Bewältigung eines MANV zu unterstützen. Daraus ergeben sich die ersten Anforderungen aus der direkten Betrachtung der Prozesse sowie den Orten an denen diese ablaufen. Der Gesamtprozess wurde mit Hilfe von Ereignisgesteuerten Prozessketten (EPK) formal erfasst. Die Eignung dieser Methodik für Prozessabläufe der Feuerwehr wurde bereits im Forschungsprojekt MOBIDAT[SO08] dargelegt. Zum besseren Verständnis der nachfolgenden Ausführungen sind der vereinfachte Gesamtprozess als EPK in Abbildung 1 und die örtliche Strukturen in Abbildung 2 dargestellt. Den Anfang der Prozesskette bildet die Entgegennahme von Notrufen mit der darauf folgenden Alarmierung der Einsatzkräfte durch die Feuerwehrleitstelle. Weiterführende Maßnahmen wie der Aufbau von Führungsstrukturen vor Ort werden direkt an der Schadenstelle durchgeführt. Die Prozessschritte sind demnach auf verschiedene Orte verteilt, woraus sich die Anforderung ergibt: (1) das System hat die Prozessschritte unter Berücksichtigung der örtlichen Verteilung zu unterstützen. Weiterhin muss das System derart konzipiert sein, dass (2) zentrale Komponenten eine Ausfallsicherheit gewährleisten, störungsresistent sind (Infrastrukturstörung, Angriff von außen) und Rückfallebenen offenhalten.

Ein weiterer Anforderungspunkt, der sich durch alle Systemebenen zieht, sind (3) Datensicherheitsaspekte, bedingt durch das Einsatzgebiet des Systems, die den Schutz von persönlichen Daten, Einsatzplänen und –dokumentationen sowie von Kommunikationskanälen gewährleisten. Außerdem soll das System, nicht nur in MANV-Lagen eingesetzt werden können, sondern auch alltägliche Notfallszenarien unterstützen, da dadurch die Effizienz der Investition in ein solches umfassendes System



maßgeblich gesteigert werden kann sowie ein automatischer Trainingseffekt durch die alltägliche Benutzung erzielt werden kann. Daraus ergeben sich weitere Anforderungen, insbesondere hinsichtlich der Flexibilität, Skalierbarkeit, Erweiterbarkeit, Adaptierbarkeit und Bedienbarkeit, auf die im Rahmen dieses Beitrages nur am Rande eingegangen werden soll. Ausgehend von diesen Anforderungen wird im nächsten Abschnitt das Konzept für die Systemarchitektur vorgestellt.

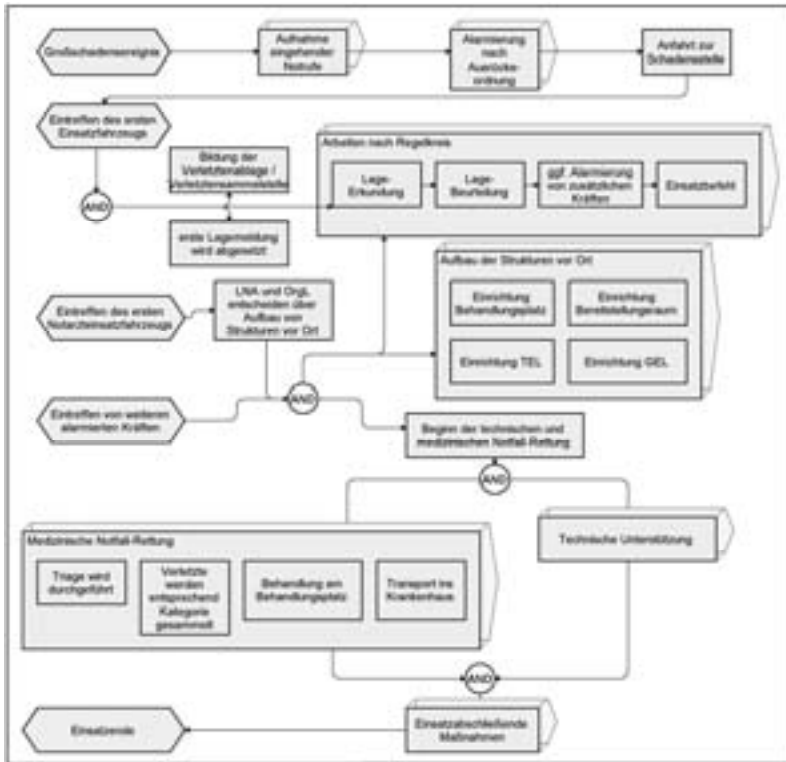


Abbildung 1: Darstellung für den Gesamtprozess bei der Bewältigung eines MANV

#### 4 Technisches Konzept für die Systemarchitektur

Nachdem der aktuelle Forschungsstand dargestellt wurde und die technischen Anforderung an das System erhoben wurden, wird in diesem Abschnitt das bisher entwickelte Konzept für die Systemarchitektur präsentiert, welches in der weiteren Projektarbeit angepasst und verfeinert wird. Das Konzept baut auf den erhobenen Anforderungen, den Implikationen aus anderen Forschungsarbeiten sowie einem vordefiniertem aber modifizierbarem Dienstportfolio auf. In diesem Dienstportfolio sind sowohl grundlegende als auch erweiterte Dienste enthalten, von denen sich ein positiver Einfluss auf den Gesamteinsatzernfolg versprochen wird. Diese Dienste befassen sich z.B. mit Lageinformation und -bewertung, Triage, Ressourcenverwaltung und -steuerung,

Telemedizinische Dienste und semiautomatischem Monitoring, Panikvermeidung, Informationsdiensten sowie einer durchgängige Einsatzdokumentation.

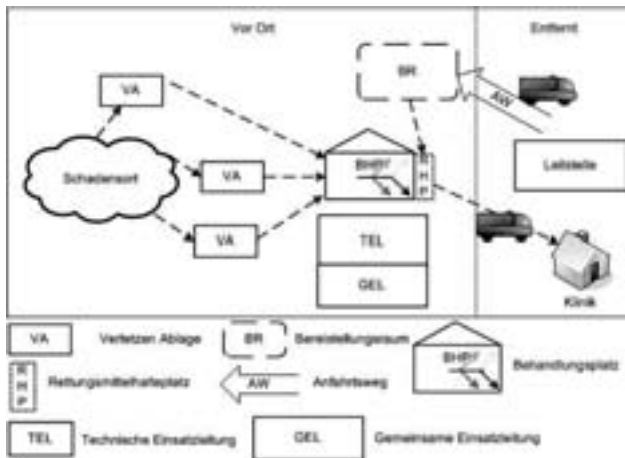


Abbildung 2: Örtliche Strukturen in einem MANV

Diese Dienste werden – bedingt durch die örtlichen Strukturen in einer MANV-Lage – in einer verteilten Umgebung benötigt (vergleiche (1) Abschnitt 3). Ein wichtiger Aspekt in der Konzeption ist daher die Dienstnutzung in dieser verteilten Umgebung. Ein klassischer Client-Server-Ansatz ist unzureichend, da in einem solchen Szenario nicht davon auszugehen ist, dass eine funktionierende Infrastruktur für einen breitbandigen Zugriff auf Serverfunktionen zur Verfügung steht. Ein Konzept in Anlehnung an die Führung der Feuerwehr beim MANV-Einsatzfall erscheint dagegen vielversprechender. Standardmäßig ist die entfernte Leitstelle die zentrale Koordinierungsstelle bei Notfalleinsätzen, bei einem MANV werden am Schadensort lokale Führungsstellen eingerichtet, die *Technische Einsatzleitung* und die *Gemeinsame Einsatzleitung*. Weiterhin kann abhängig von der Einsatzgröße ein Stab gebildet werden, welche von der Leitstelle oder anderen entfernten Örtlichkeiten wie z.B. eine Feuerwache Führungsaufgaben übernimmt. Diese Aufteilung in strategisches (entfernte Führung) und lokales (örtliche Führung) Management lässt sich auf das ALARM-System übertragen. Daher werden zwei Plattfortmtypen konzipiert, welche im Einsatz ein Dienstportfolio anbieten, das auf die zu unterstützenden Einsatzkräfte angepasst ist, die örtlich von einander getrennt sind und sich synchronisieren müssen (siehe Abbildung 3). Die beiden Plattformen sollen aber nicht unabhängig von einander entwickelt werden, vielmehr soll eine generische Basisplattform realisiert werden, welche durch eine dynamische Konfiguration von Komponenten und Diensten bzw. Dienstmodulen für den vorgesehenen Anwendungsbereich eingerichtet werden kann. Dieser Ansatz kann durch Konzepte aus dem Bereich *Service-Oriented Architectures* (SOA) realisiert werden. So wäre eine fest installierte strategische Plattform einer bestimmten Feuerwehr entsprechend ihrer Bedürfnisse angepasst. Die lokale(n) Plattform(en) für den MANV-Einsatzfall könnten einerseits in den entsprechenden Einsatzfahrzeugen dieser Feuerwehr ebenso konfiguriert auf den Bordrechnern installiert sein oder andererseits

erst im konkreten Einsatzfall und speziell auf diesen konfiguriert auf die Bordrechner migriert werden, das heißt die strategische Plattform würde die Instanziierung bzw. Replikation der lokalen Plattform steuern und realisieren. Ein Umgang mit mehreren gleichzeitig eintretenden Großschadenslagen mit einem MANV wäre damit systemseitig gesichert, da für jede MANV-Lage eine separate Lokale Plattform instanziiert werden kann.

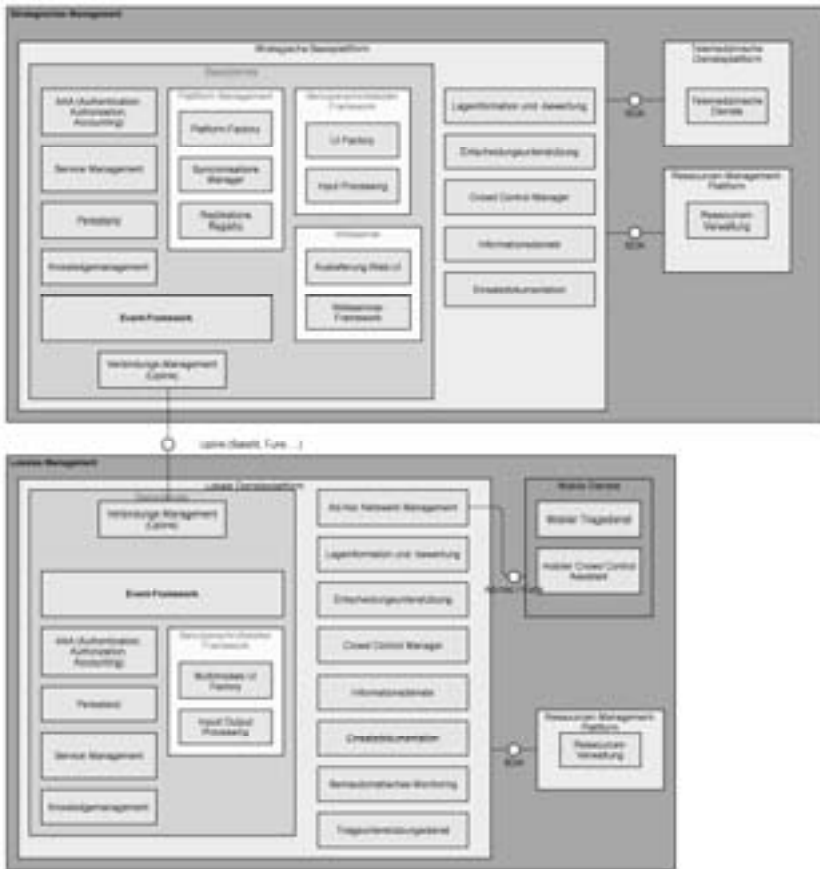


Abbildung 3: Überblick ALARM-Dienstplattformen

Weiterhin wird durch die getrennte Bereitstellung der Dienste über die beiden Plattformen eine Grundverfügbarkeit von Systemfunktionalität sichergestellt, so dass bei einer Störung bzw. Beeinträchtigung der Kommunikation zwischen strategischem und lokalem Management entscheidende Dienste weiterhin nutzbar sind (vergleiche (2) Abschnitt 3). Deshalb bietet die Plattform einerseits einen Synchronisationsbus über den verteilte Dienstinstanzen ihren Zustand synchronisieren können. So kann im Falle eines Verbindungsausfalls sofort auf der lokalen Dienstinstanz weitergearbeitet werden. Andererseits sollte die Zuverlässigkeit der Plattformverbindung maximiert werden,

was über die redundante Nutzung von Kommunikationstechnologien wie z.B. UMTS, GPRS/EDGE und Satellitendiensten<sup>3</sup> mit *Seamless Handover* [ST08] ermöglicht werden soll. Die Nutzung von Kommunikationstechnologien mit unterschiedlichen Bandbreiten erfordert auf Dienstebene eine dynamische Anpassung der Dienste selbst sowie der kommunizierten Daten (Vorfilterung, Bündelung, Komprimierung). Eine Klassifizierung von Daten mit einer Priorisierungszuordnung einerseits soll die Steuerung des Kommunikationsflusses befähigen, so dass die wichtigsten Informationen vorangestellt übertragen werden. Andererseits sollen die Dienste *situation aware* bezüglich der verfügbaren Kommunikationsbandbreite sein, das heißt sie sind in der Lage sich eigenständig hinsichtlich der aktuellen als auch der zukünftigen Situation zu adaptieren.

Die Sicherheitsanforderungen (vergleiche (3) Abschnitt 3) werden im Rahmen der Systemarchitektur nebenläufig umgesetzt. Hierbei ist aus unserer Sicht insbesondere die vorgesehene Pseudonymisierung von hoher Relevanz, da hierdurch aufbauend auf einer grundlegenden AAA-Architektur<sup>4</sup> eine weitere Absicherung der Daten gegen Missbrauch erreicht werden kann. Ein solcher Pseudonymisierungsdienst ist ausserdem ein gutes Beispiel für die genannte Duplikation und Synchronisation von Systemdiensten, da dieser in jedem Fall auf beiden Plattformebenen vorgehalten werden muss.

Zusammenfassend für die präsentierte Konzeption der ALARM-System-Architektur läßt sich sagen, dass ein zentraler Punkt in der Aufteilung der Basisplattform in zwei auf den jeweiligen Nutzungskontext adaptierten Teilplattformen liegt. Diese bieten Möglichkeiten zur Synchronisation zwischen Dienstinstanzen und zur automatischen Adaption an die Parameter der Plattformverbindung, so dass die gesamte Prozesskette im Rettungswesen über die Grenze zwischen lokalem und strategischen Management unterstützt werden kann.

## 5 Abschließende Betrachtung und weiteres Vorgehen

In dem Beitrag wurde die Konzeption für das ALARM-System zur Unterstützung von Rettungskräften in einer MANV-Lage mit dem Fokus auf eine verteilte Systemarchitektur und den sich daraus ergebenden Kommunikationsaspekten vorgestellt. Bedingt durch die fortlaufende Projektarbeit kann in diesem Beitrag kein Anspruch auf Vollständigkeit hinsichtlich des präsentierten Architekturkonzepts erhoben werden. Allerdings sind die bisherigen Ansätze aus unserer Sicht vielversprechend, um der Herausforderung der Umsetzung dieses komplexen Systems zu begegnen und dienen als Grundstein für die weiterführende Konzeptionsarbeit. Diese wird sich mit der Schärfung des vorgestellten Konzepts befassen sowie näher auf die Anforderungen und das Design der grundlegenden und erweiterten Dienste eingehen.

---

<sup>3</sup> Beispiele für Satellitendienste sind: Inmarsat, DSL via Satellit, Iridium OpenPort und Thuraya

<sup>4</sup> AAA steht für Authentication, Authorization, Accounting

Darauf aufbauend wird im ALARM-Projekt eine Referenzimplementierung des Gesamtsystems erfolgen, die in mehreren praxisnahen Übungen, darunter auch eine Großübung von einem MANV-Szenario, evaluiert wird. Bei der Evaluierung werden zur Bewertung der optimierten Prozesse die im Projekt erhobenen Erfolgsindikatoren für den Gesamtprozess angewendet und ermöglichen so eine Auswertung und Bewertung des Einflusses des Systems auf die Prozesse.

## Literaturverzeichnis

- [ER10] ERMA Projektseite, <http://erma-project.org/>, zuletzt besucht am 06.01.2010.
- [GA06] Gao T.; White D.: A next generation electronic triage to aid mass casualty emergency medical response. In: Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, Volume Supplement, S. 6501–6504, Sept. 2006.
- [HE97] Herrmann, P.: High-Tech im Notfall. Der Anaesthetist, 46, S.727-730, 1997.
- [LO06] Löffler, J; Schon, J; Köhler, J.: SHARE: supporting large-scale rescue operations with communication and information services over mobile networks. In(ACM): MobiMedia '06: Proceedings of the 2nd international conference on Mobile multimedia communications, S. 1–6, New York, NY, USA, 2006
- [MA10] MANET Projektseite, [www.manet-projekt.de](http://www.manet-projekt.de), zuletzt besucht am 08.02.2010
- [OA10] Oasis Projektseite, <http://www.oasis-fp6.org>, zuletzt besucht am 06.01.2010.
- [OR10] Orgamir Projektseite, <http://www.orgamir.de/>, zuletzt besucht am 06.01.2010.
- [OR08] ORCHESTRA Consortium. ORCHESTRA – An Open Service Architecture for Risk Management, <http://www.eu-orchestra.org/docs/ORCHESTRA-Book.pdf>, 2008
- [RE10] Rettungsassistenzsystem TU Clausthal Projektseite, <http://www2.in.tu-clausthal.de/~Rettungsassistenzsystem>, zuletzt besucht am 06.01.2010.
- [SA10] SAFeR Projektseite, <http://www-cik.uni-paderborn.de/Forschung/SAFeR>, zuletzt besucht am 06.01.2010.
- [SK09] Skorning, M.; Bergrath, M.; Rörtgen, M.; Brokmann, J.C.; Beckers, S.K. Protogerakis, M.; Brodziak, M.; Rossaint. R.: "E-Health" in der Notfallmedizin - das Forschungsprojekt Med-on-@ix. Der Anaesthetist, 58(3) S.285–292, 2009.
- [SO08] Soboll, M.; Binder, B.; Quix, C., Geisler, S.: Prozessmodellierung der mobilen Datenerfassung für den Rettungsdienst bei einer Großschadenslage. In (Höhn, R; Linssen, O.). Vorgehensmodelle und Implementierungsfragen – Akquisition – Lokalisierung – soziale Maßnahmen – Werkzeuge. Tagungsband zum 16. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e.V. (GI), S. 109-125, Düsseldorf, 09.04.2009.
- [ST08] Steuer, F.; Geithner, T.; Kaschwig, T.; Bür, K.; Albayrak, S.: A connectivity management system for vehicular telemedicine applications in heterogeneous networks. In (ICST): TridentCom '08: Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities, S. 1–10, Brussels, Belgium, 2008.
- [XM10] xMotion Projektwebsite, [http://www.kilab.uni-bremen.de/index.php?option=com\\_content&task=view&id=28&Itemid=52](http://www.kilab.uni-bremen.de/index.php?option=com_content&task=view&id=28&Itemid=52), zuletzt besucht am 06.01.2010.
- [ZI06] Ziegler, V.; Hiermann, E.; Kippnich, U.; Freibott, E.; Theiss, S.; Weidenhaupt, K. ; Rose, G.; Siebler, M.; Griewing. B.: Das Stroke Angel-Projekt - Computerunterstützte Optimierung der Schlaganfall-Versorgungskette In: Tagungsband des 79. Kongresses der Deutschen Gesellschaft für Neurologie (DGN) mit Fortbildungsakademie, S. 20-24., Mannheim, 2006.

# Systemkonzept für ein Mobiles Informationssystem zur Prozessoptimierung in Feuerwehren und öffentlichen Verwaltungen<sup>1</sup>

Anna Maria Japs<sup>2</sup>, Stephan Prödel<sup>3</sup>

Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung (C.I.K)  
Universität Paderborn, Fakultät Maschinenbau

Warburger Str. 100  
33098 Paderborn  
japs@cik.uni-paderborn.de  
s.proedel@cik.uni-paderborn.de

**Abstract:** In zeitkritischen Situationen, wie z. B. dem abwehrenden Brandschutz, bleibt in den seltensten Fällen ausreichend Zeit um langwierige Recherchen durchzuführen und somit für den Einsatz notwendige Informationen zu finden. Vor allem mit dem hohen Datenaufkommen der letzten Jahre wird es immer schwieriger die relevanten Dokumente zu finden und zeitnah einzusetzen. Die vorliegende Arbeit stellt ein Systemkonzept für ein IT-System, mit welchem es ermöglicht wird relevante Informationen aus einer großen Menge unstrukturierter und dezentraler Datenquellen dem Endanwender zu Verfügung zu stellen.

## 1 Modulares Systemkonzept für zeitkritische Situationen

Die Umsetzung eines IT-Systems zur Entscheidungsunterstützung durch situationsbezogene Informationen erfordert einerseits genaue Kenntnis über die spezifischen Informationsbedürfnisse und andererseits die besondere Anforderung an das IT-System. [Hi08]

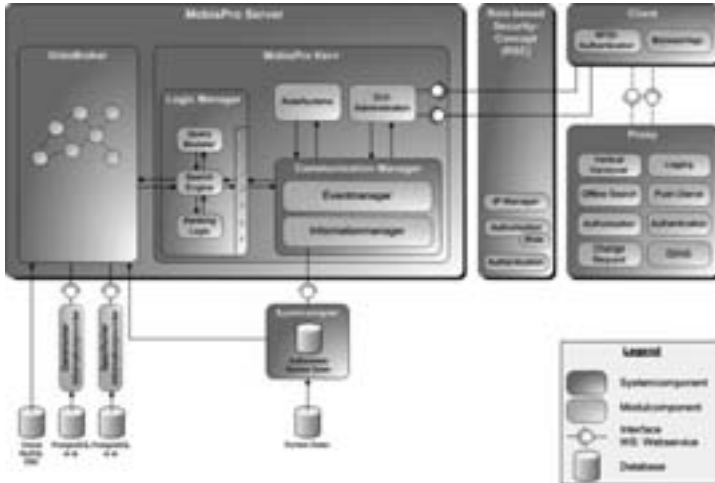
In zeitkritischen Situationen muss ein IT-System vor allem dem Endnutzer Informationen anbieten, welche erst zur Laufzeit aus heterogenen Datenquellen abgerufen und verknüpft werden können. Um diesen Ansatz zu ermöglichen fiel die Entscheidung auf ein modulares IT-System und einem webbasierten Client. [Ga09] Die Systemkomponenten (s. Abb. 1) verfügen dadurch jeweils über einen Webservice, der als Schnittstelle zwischen den einzelnen Komponenten fungieren kann. In den jeweiligen Systemkomponenten selbst werden können die parallel eingehenden Anfragen über die einzelnen Modulkomponenten verarbeitet und weitergeleitet werden.

---

<sup>1</sup> Dieses Paper basiert auf der Arbeit der Universität Paderborn im Projekt Mobis Pro im Rahmen des Förderprogramms SimoBIT des BMWi ([www.simobit.de](http://www.simobit.de)).

<sup>2</sup> Dipl.-Inf. Anna Maria Japs ist Mitarbeiterin am Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung.

<sup>3</sup> Dipl.-Ing. Stephan Prödel ist Mitarbeiter am Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung und Projektleiter im Forschungsvorhaben Mobis Pro.



**Abbildung 1:** Systemkonzept MobisPro

Die Kommunikation zwischen den Komponenten beginnt beim Client, der sich beim RSC anmeldet. Nach erfolgter Authentifizierung wird der Client beim IP-Manager registriert, um die Konsistenz einer Session im Falle eines Verbindungsabbruches zu gewährleisten. Daraufhin übermittelt die GUI-Administration eine Liste mit offenen Einsätzen aus denen der Benutzer auswählen kann. Die in der GUI-Administration integrierte Benutzerverwaltung liefert die Information, welche Rollen im ausgewählten Einsatz zur Verfügung stehen und für welche der angemeldete Benutzer autorisiert ist. Der User kann nun aus den ihm angebotenen Rollen auswählen. Nach Auswahl einer Rolle ruft die GUI-Administration über den Communication Manager das zugehörige Prozessmodell aus der RoleRuntime ab und übermittelt es an den Client.

Eine Anfrage des Clients geht bei der GUI-Administration ein. Die GUI-Administration verwaltet die Anfragen unterschiedlicher Clients und leitet die Anfrage, erweitert um den globalen Kontext aus dem Einsatz und der Client-ID, an den Communication Manager weiter. Dieser ermittelt aus dem in der RoleRuntime hinterlegten Prozessmodell den Informationsbedarf und übergibt Anfrage, Kontext und Informationsbedarf an die SearchEngine im LogicManager. Mit Hilfe des QueryModelers wird die Suchanfrage in eine F-Logic-Anfrage umformuliert und an OntoBroker gestellt. Die zurückgelieferten Antworten werden in der RankingLogic bewertet, sortiert und über den Communication Manager an die GUI-Administration zurückgegeben. Diese sendet die Antworten an den entsprechenden Client zurück.

## Literaturverzeichnis

- [Ga09] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software. Addison-Wesley, München, 2009.
- [Hi08] Hildebrand, H.; Mielke, M.; Gebauer, M.; Hinrichs, H.: Daten- und Informationsqualität: Auf dem Weg zur Information Excellence. Vieweg+Teubner, 2008.

# Das Paradigma der Event-Driven Architecture als Grundlage für ein entscheidungsunterstützendes IT-System im Zivil- und Katastrophenschutz

Robin Marterer, Benedikt Birkhäuser, Rainer Koch

Universität Paderborn

Computeranwendung und Integration in Konstruktion und Planung (CIK)

Pohlweg 47-49

33098 Paderborn

{marterer, b.birkhaeuser, r.koch}@cik.uni-paderborn.de

**Abstract:** Entscheidungsprobleme, denen sich die strategischen Führungsebenen der Feuerwehr im Krisenmanagement stellen müssen, sind als hoch komplexe, stark vernetzte, intransparente und dynamische Entscheidungsprobleme anzusehen. Im Zuge vorangegangener Projekte hat sich herausgestellt, dass insbesondere im Bereich der strategischen Ressourcenkoordination Unterstützungspotential liegt. Aktueller technologischer Fortschritt im Bereich der Informations- und Kommunikationstechnologie in der Domäne der zivilen Sicherheit ermöglichen die Weiterentwicklung dieses Ansatzes. Das Projekt PRONTO verfolgt einen ereignisbasierten Ansatz um die skizzierten Herausforderungen durch ein echtzeitfähiges IT-System zu lösen. Aufbauend auf diesem Projekt werden in diesem Paper Aspekte der Anwendung einer Event-Driven Architecture in der Domäne der zivilen Sicherheit erörtert und diskutiert.

## 1 Einleitung

Entscheidungsträger der Feuerwehr stehen im Kontext der Einsatzführung vor komplexen Herausforderungen. Vor diesem Hintergrund stellt sich die Frage, wie eine angemessene Unterstützungsfunktionalität umgesetzt werden kann. Dieses Paper erörtert und diskutiert, inwieweit der Ansatz der Event-Driven Architecture einen Beitrag zur Realisierung liefern kann. In Abschnitt 2 werden dazu Forschungsergebnisse und Grundlagen beschrieben, die eine Einordnung der auftretenden Herausforderungen ermöglichen. Abschnitt 3 erläutert die derzeitigen, technologischen Strömungen, auf denen das Konzept der Event-Driven-Architecture aufbaut. Die Umsetzung der skizzierten Ansätze im Projekt PRONTO wird in Abschnitt 4 erläutert. Abschliessend wird versucht auf Grundlage der ersten Projektergebnisse ein Fazit zu ziehen.



## 2 Grundlagen und Herausforderungen

Das Projekt PRONTO betrachtet dazu unter anderem die Fallstudie 'Emergency Rescue Operation' (ERO), in der die Ansätze für den Bereich der Feuerwehr aufgegriffen und implementiert werden. Forschungsschwerpunkt des CIK ist der Bereich der zivilen Sicherheit, der somit im Rahmen dieses Papers fokussiert wird.

Im Folgenden werden die wesentlichen Grundrisse der Einsatzstrukturen im Zivil- und Katastrophenschutz am Beispiel der Feuerwehr Dortmund erläutert, die auf unterschiedlichen gesetzlichen Regelungen basieren. [FWD03, NW, Dor] Der Gesamtzuständigkeitsbereich der Feuerwehr ist in Einsatzbezirke aufgeteilt, für die jeweils eine Feuerwache zuständig ist. Wie viele und welche taktischen Einheiten und Einsatzfahrzeuge alarmiert werden, wird durch eine Alarm- und Ausrückeordnung (AAO) definiert. Im Einsatzfall ist die Feuerwehr hierarchisch organisiert, folglich existiert zu jedem Einsatz genau eine Person als Einsatzleiter, welche die Verantwortung trägt. In einer einfachen Betrachtungsweise können operative, taktische und strategische Führungsebenen unterschieden werden. Auf der strategischen Entscheidungsebene kommt in großen Schadenslagen ein strategischer Einsatzstab der Feuerwehr zum Einsatz, der im administrativen Bereich mit dem Krisenstab zusammenarbeitet.

Ziel von PRONTO ist es, Entscheidungsträgern der beschriebenen Domäne eine IT-basierte, strategische Einsatzunterstützung zur Verfügung zu stellen. Einsatzlagen, in denen strategische Entscheidungsträger und Stäbe zum Einsatz kommen, zeichnen sich durch hohe Komplexität, starke Vernetztheit, Intransparenz und große Dynamik aus. [PO92, BPK09] Eine Grundannahme von PRONTO ist es, die wesentlichen Informationen aus der realen Welt als Ereignisse aufzufassen und diese als Software-Objekte in ein IT-System zu überführen, um sie sammeln und miteinander in Bezug setzen zu können. Viele der für Entscheidungsträger bedeutungsvollen Informationen sind oft prinzipiell vorhanden, stehen ihnen aber in den durch hohen Entscheidungsdruck geprägten Situationen nicht (schnell genug) zur Verfügung oder sind unerreichbar in unterschiedlichen Systemen verborgen. Die Verwendung eines ereignisorientierten Ansatzes bringt eine Entlastung der Entscheidungsträger durch die überwiegend interaktionslose Bereitstellung von Informationen mittels Push-Ansatz mit sich.

## 3 Stand der Technik

Zu den Zwecken der Anwendungsintegration und der Abbildung von Geschäftsprozessen auf IT-Systeme hat sich in den letzten Jahren für viele Anwendungsbereiche das Paradigma der *Service-Oriented Architecture* (SOA) als zielführend erwiesen. Sprachen wie die XML-basierte *Business Process Execution Language for Web Services*<sup>1</sup> (WS-BPEL) dienen dazu, Geschäftsprozesse flexibel zu modellieren und durch die Komposition von *Web Services* zu realisieren. Auf der Implementierungsseite stehen SOAP, WSDL und UDDI als Mittel der Wahl zur Verfügung.

---

<sup>1</sup><http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (Zugriff: 2010-01-09)

Das Konzept der *Real-Time Enterprise* (RTE) [TYPM09] verfolgt das Ziel, Geschäftsprozesse permanent zu überwachen und kontinuierlich zu verbessern, um Zeit und Kosten einzusparen. Aus einer solchen Unternehmenskultur ergeben sich erweiterte Anforderungen an eine Systemlandschaft. Neben der Anwendungsintegration und der Beschreibung von Geschäftsprozessen erfordert die RTE Echtzeiteigenschaften, die durch herkömmliche SOAs nicht erfüllt werden können. Genau diese Echtzeiteigenschaften sind es, die neben der Anwendungsintegration für die Fallstudie ERO im Projekt PRONTO von besonderer Bedeutung sind.

Das Paradigma der *Event-Driven Architecture*<sup>2</sup> (EDA) [TYPM09] stellt einen aktuellen Ansatz dar, um diesen erweiterten Anforderungen zu begegnen. Eine Referenzarchitektur [DEF<sup>+</sup>08, S. 133] soll nachfolgend der Beschreibung der zentralen Konzepte einer EDA dienen, deren zentralen Artefakte erwartungsgemäß Ereignisse darstellen. Diese können aus beliebigen an das System angeschlossenen Ereignisquellen stammen und werden über einen Ereigniskanal, realisiert durch eine *Message-Oriented Middleware* (MOM), an eine Softwarekomponente für das *Complex Event Processing* (CEP) [Luc02] weitergeleitet. Die MOM aus dem Bereich *Enterprise Application Integration* (EAI) dient neben ihrer Funktion als Transportkanal für Ereignisse auch als Bindeglied zwischen allen an das System angeschlossenen Softwarekomponenten. Hierzu zählen nicht nur verschiedene Systeme, die Ereignisse erzeugen, sondern auch Softwarekomponenten aus Vorgängerprojekten. Zentrales Konzept ist das *Publish-Subscribe*-Entwurfsmuster, das eine flexible Weiterleitung von Ereignissen, basierend auf verschiedenen Ereignistypen, an verschiedene Softwarekomponenten ermöglicht. Das CEP realisiert eine regelbasierte Verarbeitung eingehender Ereignisse. Dies geschieht anhand von durch eine *Event Processing Language* (EPL) formulierten Regeln (auch: Event Pattern), die meist logische, zeitliche oder auch räumliche Zusammenhänge ausdrücken. Die Anwendung dieser Regeln hat die Erzeugung von Ereignissen zur Folge, die einen höheren Informationsgehalt aufweisen. Ein Beispiel für eine EPL stellt *Rapide*<sup>3</sup> [Luc02, S. 145ff] dar. Eingehende oder durch das CEP erzeugte Ereignisse können dazu verwendet werden, Entscheidungsträger in einem Unternehmen über beliebig komplexe Zustände und Vorgänge zu informieren oder Geschäftsprozesse anzustoßen. Die aggregierte Darstellung von Ereignissen und ihren Zusammenhängen lässt sich über Dashboards realisieren und ist dem *Business Activity Monitoring* (BAM) zuzuordnen. Mit [LS08] stellt die *Event Processing Technical Society*<sup>4</sup> (EPTS) eine aktuelle Zusammenfassung der Begrifflichkeiten im CEP-Bereich zur Verfügung. Die Tatsache, dass alle großen – wie auch kleinere Anbieter – von Middleware-Produkten EDA- und CEP-Konzepte in ihre Produkte integrieren, lässt auf das Vorhandensein einer entsprechenden Nachfrage nach solchen Technologien auf dem Markt schließen. Im Open-Source-Bereich steht *N/Eesper*<sup>5</sup> zur Verfügung.

Eine andere Bezeichnung für eine Ereignisquelle innerhalb eines EDA-Systems ist auch 'Sensor'. Dieser Begriff spiegelt den Charakter wider, der die meisten Ereignisquellen beschreibt. Oft handelt es sich um Hardware mit assoziierten Softwarekomponenten, die im Zusammenspiel Ereignisse der realen Welt (physikalische Phänomene) erkennen und in

---

<sup>2</sup>auch: *Event-Driven Service-Oriented Architecture* (ED-SOA), *SOA 2.0* oder *advanced SOA*

<sup>3</sup><http://complexevents.com/rapide> (Zugriff: 2009-01-09)

<sup>4</sup><http://www.ep-ts.com> (Zugriff: 2010-01-09)

<sup>5</sup><http://esper.codehaus.org> (Zugriff: 2010-01-09)

durch Computer verarbeitbare Formate (digitale Werte) überführen. Beispiele hierfür sind Temperatur-, Beschleunigungs- und Wasserstandsensoren – aber auch Mikrofone, Videokameras, RFID-Transponder und GPS-Geräte.

Eine Initiative, die sich der standardisierten Verknüpfung von Sensoren und Web Services verschrieben hat, ist das *Sensor Web Enablement*<sup>6</sup> (SWE). Hier bedient man sich des Konzepts *Data Fusion* [LHL09] zum Zweck der „Kombination von Daten (oder Informationen), um den Zustand eines Aspekts der Welt zu beschreiben oder zu prognostizieren“<sup>7</sup>. Die Parallelen zu EDA und CEP sind offensichtlich. Die Anwendbarkeit des SWE auf die Domäne des *Zivil- und Katastrophenschutzes* wurde bereits im EU-Projekt OSIRIS<sup>8</sup> erforscht.

## 4 Adaption auf PRONTO

Die Anforderungsanalyse im Projekt PRONTO ergab zahlreiche Herausforderungen, die an dieser Stelle nur grob skizziert werden können und im Rahmen des Systemdesigns Berücksichtigung finden mussten: Ein wesentlicher Aspekt ist die Notwendigkeit der Anbindung zahlreicher heterogener Softwarekomponenten. Diese werden von unterschiedlichen Partnern auf zum Teil unterschiedlichen Plattformen entwickelt. Zusätzlich sind diese über unterschiedlich robuste und performante Netzwerke an das System angebunden. Berücksichtigt werden müssen zusätzlich Legacy-Module, deren Anbindung ebenfalls sichergestellt werden muss. Der breite Ansatz von PRONTO mündet im Bedarf einer hohen Flexibilität in Bezug auf die kurzfristige Anpassung des Systems an sich geänderte Anforderungen. Dies wird im Rahmen der Softwareentwicklung durch einen agilen Ansatz aufgegriffen, muss sich allerdings ebenso in den Designentscheidungen bezüglich der Systemarchitektur widerspiegeln.

Zentrale Aspekte der beschriebenen Referenzarchitektur ließen sich direkt auf die Softwarearchitektur von PRONTO übertragen. Abbildung 1 beschreibt das PRONTO-Gesamtsystem bestehend aus Subsystemen und im Kontext externer an das Demonstrator-System angeschlossener Systeme. Die Architektur unterteilt sich grob in zwei Teile: die verteilte Seite, welche mobile Einheiten umfasst, die Ereignisse über drahtlose Netzwerke an das System übermitteln und die Serverseite, die ebenfalls Komponenten umfasst, welche Ereignisse in das System einspeisen. Zusätzlich dient die Serverseite der Verarbeitung aller eingehenden Ereignisse. Eine zentrale Annahme ist, dass jede Softwarekomponente sowohl in der Rolle des Ereignisproduzenten als auch des Ereigniskonsumenten auftreten kann.

Es wird davon ausgegangen, dass externe Systeme sowohl auf der verteilten als auch auf der Serverseite in der Rolle von Ereignisproduzenten auftreten. Die von ihnen erzeugten Ereignisrohdaten (1a, 1b) werden von den mit ihnen assoziierten Komponenten

<sup>6</sup><http://www.opengeospatial.org/ogc/markets-technologies/swe> (Zugriff: 2010-01-09)

<sup>7</sup>[http://52north.org/joomla/components/com\\_publications/publications/2007/52N\\_EinfuehrungSensorWeb.pdf](http://52north.org/joomla/components/com_publications/publications/2007/52N_EinfuehrungSensorWeb.pdf) (Zugriff: 2010-01-09)

<sup>8</sup><http://osiris-fp6.eu> (Zugriff: 2010-01-09)

*Event Recognition* in den Subsystemen *Event Creation* in ein vom System akzeptiertes Ereignisformat transformiert und an die Komponente *Ereigniskanal* (MOM) im Subsystem *Integration* übermittelt (2a, 2b).

Zu den typischen Sensoren auf der verteilten Seite zählen u. a. GPS-Geräte, Mikrofone, Videokameras, Beschleunigungs-, Treibstofftank- und Wassertanksensoren, die allesamt für gewöhnlich auf verschiedenen Feuerwehrfahrzeugen untergebracht sind. Typische Sensoren auf der Serverseite greifen auf zentrale Systeme wie das Einsatzleitsystem, den Sprech- und Datenfunk sowie externe Services (Wetter, Hochwasserpegel) zu.

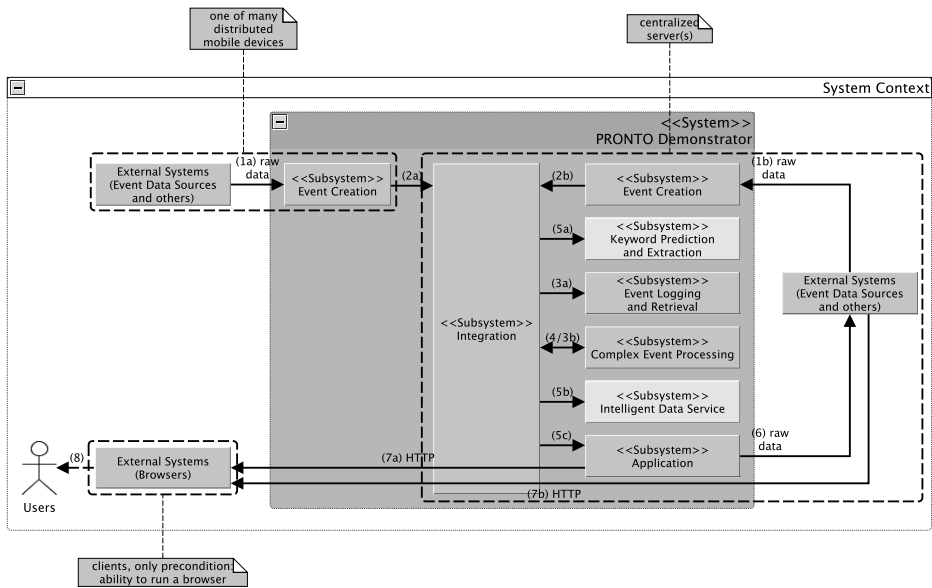


Abbildung 1: Gesamtarchitektur des PRONTO-Systems

Das Subsystem *Event Logging and Retrieval* abonniert über den *Publish-Subscribe*-Mechanismus des Ereigniskanals Ereignisse, die archiviert werden sollen und speichert diese in einer Datenbank, um sie zu einem späteren Zeitpunkt, beispielsweise zu Test- oder Trainingszwecken, wieder abrufen – und ggf. erneut in das System einspielen – zu können (3a). Die dem Subsystem *Complex Event Processing* zugeführten Ereignisse werden als *Low-Level Events* (3b) bezeichnet und innerhalb dieses Subsystems zu *High-Level Events* (4) kombiniert. Sowohl die Software zur Ereignisverarbeitung als auch die zugehörige *Event Processing Language* werden von Projektpartner NCSR entwickelt. Die resultierenden Ereignisse werden in erster Linie vom Subsystem *Application* verarbeitet (5c), welches die Komponenten *Intelligent Resource Management* (IRM), MAP, *Decision Support* (DS) und CHAT enthält. Die Subsysteme *Keyword Prediction and Extraction* (KPX) sowie *Intelligent Data Service* (IDS) werden ebenfalls von Projektpartner NCSR entwickelt. KPX dient der Unterstützung der Spracherkennung durch entsprechende Komponenten in den Sub-

systemen *Event Creation*, die durch Projektpartner FhG geliefert werden. IDS enthält eine Ontologie der Anwendungsdomäne *Zivil- und Katastrophenschutz*, welche für die Berechnungen in den Subsystemen CEP sowie KPX dient. Zunächst wird davon ausgegangen, dass die Benutzer des Systems webbasiert auf das System zugreifen. Hierzu werden die Komponenten im Subsystem *Application* in einer einheitlichen Anwendung integriert und über die Komponente *Web Server* im Subsystem *Integration* zur Verfügung gestellt (7, 8).

Während der gesamten Konzeption des Systems wurde die bundesweite Einführung des TETRA-Digitalfunks für *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) [Lin08] berücksichtigt. TETRA wird Datenübertragungsmöglichkeiten mit sich bringen, die in der bisherigen, analogen Welt des BOS-Funks so nicht verfügbar waren. Das TETRA zu Grunde liegende TCP/IP-Protokoll ermöglicht eine leichtere Anbindung verteilter Ereignisproduzenten an die auf dem selben Protokoll aufbauenden und bereits etablierten Webtechnologien, die im EDA-Bereich Anwendung finden. Somit können Ansätze der Event-Driven Architecture nun prototypisch auf die Domäne der Feuerwehr übertragen werden. Die erste Generation des BOS-Funks bringt Datenübertragungsmöglichkeiten mit sich, die anfangs noch sehr begrenzt sein werden. Die Übertragungsraten der zweiten Generation werden mit den *TETRA-enhanced Data Services* (TEDS) in etwa im Bereich von *Enhanced Data Rates for GSM Evolution* (EDGE) liegen und somit deutlich performanter sein.

Derzeit befindet sich PRONTO in der Phase der ersten Prototypentwicklung. Für den Ereigniskanal wird derzeit der von JBoss abgeleitete Message Channel HornetQ verwendet. Die aktuellen Prototypen basieren auf HornetQ und verwenden JMS-Nachrichten, um Ereignisse im JSON-Objekt-Format zu versenden.

## 5 Fazit

In weiteren Teilprojekten müssen Fragen der konkreten Ereignisdefinition, der Ereignismodellierung, der Ereigniserkennung sowie der Darstellung der Ereignisse auf der GUI gelöst werden. Zusätzlich wird versucht, die erreichten Ergebnisse auf eine zweite Fallstudie 'City Transport Management' zu übertragen.

Dazu werden in den nächsten Projektschritten die einzelnen Komponenten durch die beteiligten Partner entwickelt werden und in einem ersten, lauffähigen Demonstrator integriert werden. In den weiteren Phasen des Projektes werden die ereigniserkennenden Komponenten sukzessive durch Testläufe auf der Basis von Realdaten verbessert werden. Zusätzliche Herausforderungen stellen sich im Bereich des GUI-Designs, der Systemperformance sowie der Sensoranbindung. Das Gesamtergebnis des Projektes wird im Rahmen einer Übung mit den Endanwenderpartnern evaluiert werden.

Nimmt man die aufgenommenen Anforderungen als ein Bewertungsmaßstab, so konnten bisher alle wesentlichen Anforderungen durch die Entscheidung für eine Event-Driven Architecture erfüllt werden. Eine endgültige Evaluation der getroffenen Designentscheidung kann zu diesem Zeitpunkt nicht erfolgen. Diese kann erst im Zuge des weiteren Projektfortschrittes vor dem Hintergrund der gestellten Anforderungen und zum Beispiel

in Anlehnung an gängige Kriterien für Softwarequalität [ISO01] erfolgen.

## Literatur

- [BPK09] Benedikt Birkhäuser, Jens Pottebaum und Rainer Koch. Unterstützung von Einsatzentscheidungen der Feuerwehr auf Basis IT-unterstützter Kräftekoordination. In Stefan Fischer, Hrsg., *Informatik 2009: Im Focus das Leben: Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 28.9. - 2.10.2009 in Lübeck*, Jgg. 154 of *GI-EditionProceedings*, Seiten 1393–1406. Gesellschaft für Informatik, Bonn, 2009.
- [DEF<sup>+</sup>08] Jürgen Dunkel, Andreas Eberhart, Stefan Fischer, Carsten Kleiner und Arne Koschel. *Systemarchitekturen für verteilte Anwendungen*. Carl Hanser Verlag, München, 2008.
- [Dor] Feuerwehr Dortmund. Brandschutzbedarfsplan der Stadt Dortmund.
- [FWD03] *Feuerwehr-Dienstvorschrift 100: Führung und Leitung im Einsatz - Führungssystem*. Deutscher Gemeindeverlag, Stuttgart, 2003.
- [ISO01] ISO/IEC. ISO/IEC 9126-1:2001, Software Engineering - Product Quality - Part 1: Quality model, 2001.
- [LHL09] Martin E. Liggins, David L. Hall und James Llinas. *Handbook of Multisensor Data Fusion*. CRC Press, Boca Raton, second edition. Auflage, 2009.
- [Lin08] Christof Linde. *Aufbau und Technik des digitalen BOS-Funks*. Franzis Verlag, Poing, 2008.
- [LS08] David Luckham und Roy Schulte. Event Processing Glossary - Version 1.1, 2008.
- [Luc02] David Luckham. *The Power of Events*. Addison-Wesley, Pearson Education, Boston, 2002.
- [NW] Landtag Nordrhein-Westfalen. Gesetz über den Feuerschutz und die Hilfeleistung Nordrhein-Westfalen: FSHG: 10.02.1998.
- [PO92] Wiebke Putz-Osterloh. Entscheidungsverhalten. In Erich Frese, Hrsg., *Handwörterbuch der Organisation*, Jgg. 2 of *Enzyklopädie der Betriebswirtschaftslehre*, Seiten 585–599. Poeschel, Stuttgart, 1992.
- [TYPM09] Hugh Taylor, Angela Yochem, Les Phillips und Frank Martinez. *Event-Driven Architecture*. Addison-Wesley, Pearson Education, Boston, 2009.



# Die praxisorientierte Perspektive der Anforderungsanalyse aus Sicht der Endanwender

Jörg Rhode

Business Development  
DB Kommunikationstechnik GmbH  
Caroline-Michaelis-Strasse 5-11  
10115 Berlin  
E-Mail: joerg.rhode@deutschebahn.com

**Abstract:** Die fortwährende Weiterentwicklung der Instrumente des Requirements Engineering in Wissenschaft und Praxis hat zu einer deutlich gestiegenen Erfolgsquote hinsichtlich der Zielerreichung von Software- und IT-Projekten geführt, wie neuere Untersuchungen zeigen.<sup>1</sup> Doch eine weiterhin bestehende Abbruchquote von 32%<sup>2</sup> verdeutlicht auch, dass für die Zukunft weiterhin Handlungsbedarf besteht. Mit einem erheblichen Anteil von 87%, wird die Ursache im Bereich der Anforderungsanalyse gesehen.<sup>3</sup> Hier zeigt sich sehr deutlich der Handlungsbedarf aus Anwender- und Entwicklersicht. Mit der vorliegenden Arbeit soll ein praxisorientierter Diskussionsbeitrag, im Sinne einer kritischen Würdigung des aktuell vorliegenden Methodenansatzes im Bereich der Anforderungsanalyse aus Endanwendersicht, zu dieser umfassenden Thematik gegeben werden. Dafür abgestellt wird auf die Erfahrungen aus der Praxis des Forschungsprojektes *Sicherheit in offenen Verkehrssystemen Eisenbahn Management* [SinoVE Management].

## 1. Motivation

Der Aspekt der zivilen Sicherheit in kritischen Verkehrsinfrastrukturen ist gerade auch vor dem Hintergrund der aktuellen Gefährdungslage eine wachsende gesellschaftliche Herausforderung. Dabei stehen – neben den politischen Akteuren – speziell auch die in diesen Bereichen agierenden Sicherheitsbehörden sowie die Verkehrsunternehmen und Infrastrukturbetreiber zunehmend in der Pflicht, präventive Sicherheitsvorsorge zu betreiben. Dieses Ziel wird ohne maßgebliche Unterstützung durch Software- und IT-Innovationen nicht leistbar sein, was die Praxisrelevanz dieses Themas verdeutlicht.

---

<sup>1</sup> Vgl. [Eb2008], S. 3-4.

<sup>2</sup> Vgl. [St2009], Website.

<sup>3</sup> Vgl. [Eb2008], S. 3-4.



Die Politik hat auf diesen Bedarf unter anderem mit der Initiierung des Forschungsprogramms *Forschung für zivile Sicherheit* aktiv reagiert. Abgestellt wird dabei auf die Entwicklung innovativer Lösungsansätze, die nachhaltig die Sicherheit der Bürgerinnen und Bürger in kritischen Verkehrsinfrastrukturen stärken, ohne Einschränkung von Freiheit und Persönlichkeitsrechten.<sup>4</sup>

Der Herausforderung zur szenarioorientierten Entwicklung von Innovationen zur Unterstützung der zivilen Sicherheit, unter Verbindung technologischer und gesellschaftlicher Fragestellungen, stellen sich mit namhaften Vertretern aus Endanwendern, Forschung und Industrie auch die Partner des Projektes SinoVE Management.<sup>5</sup>

Daraus ergibt sich unmittelbar die Frage nach dem Verhältnis von ingenieurwissenschaftlicher und wirtschaftswissenschaftlicher Sicht auf die Entwicklung von Innovationen: Dichotomie oder iterativ-inkrementelles Vorgehensmodell?

Zielstellung des vorliegenden Diskussionsbeitrages ist eine kritische Würdigung des aktuell vorliegenden Methodenansatzes im Bereich der Anforderungsanalyse, in einem Vergleich bezogen auf die bisherigen Erfahrungen aus der Praxis des Forschungsprojektes SinoVE Management.

Dafür wird im nachfolgenden *zweiten Kapitel* eine vergleichende Darstellung des Innovationsansatzes aus wirtschafts- und ingenieurwissenschaftlicher Sicht gegeben. Danach schließt sich im *dritten Kapitel* eine kurze Einführung in das Projekt SinoVE Management, die Darstellung der projektbezogenen Entwicklung des Kataloges der Endanforderungen sowie ein Fazit mit kritischer Würdigung der methodischen Ansätze in den zu Grunde liegenden Wissenschaftsdisziplinen, unter Einschluss von Hinweisen auf Innovationshemmnisse und mögliche Lösungsansätze an. Im abschließenden *vierten Kapitel* werden eine Zusammenfassung und ein Ausblick auf mögliche künftige Entwicklungen gegeben.

## **2. Innovationen: Vergleich ingenieur- und wirtschaftswissenschaftliche Perspektive**

Da in der Literatur eine vielfach differenzierte Begriffsverwendung vorliegt, soll zunächst der Versuch einer Systematisierung der Anforderungsanalyse unternommen werden. Dazu werden nach einer Einführung in das Verhältnis von Anforderung und Innovation Definitionen für die Begriffe Anforderung und Anforderungsanalyse vorgestellt.

---

<sup>4</sup> Vgl. [Bu2010a], Website.

<sup>5</sup> Vgl. [Bu2010b], Website.

Handelt es sich bei der Anforderung um eine anwenderorientierte Zieldefinition aus ingenieurwissenschaftlicher Perspektive<sup>6</sup>, zielt die Innovation auf das wirtschaftswissenschaftliche Verständnis ab; durch eine neuartige Art und Weise der Kombination von Produktionsfaktoren soll die Aufstellung einer neuen Produktionsfunktion erreicht werden.<sup>7</sup>

### 1. *Ingenieurwissenschaftliche Sicht*

Unter einer Anforderung soll die Definition nach dem IEEE Standard 610.12.-1990 verstanden werden. Danach handelt es sich bei Anforderungen um "...

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1) or (2)."<sup>8</sup>

Die Definition von Anforderungsanalyse (oder des im Bereich der Softwareentwicklung synonym verwendeten Begriffes des Requirements Engineering) nach Boehm lautet wie folgt:

„Software requirements engineering is the discipline for developing a complete, consistent unambiguous specification – which can serve as a basis for common agreement among all parties concerned – describing what the software product will do (but not how it will do it; this is to be done in the design specification).“<sup>9</sup>

Insgesamt wird durch beide Begriffsdefinitionen eine klare Abgrenzung der Anforderung gegenüber Merkmalen und Lösungsansätzen deutlich. Handelt es sich bei Merkmalen um eine realisierte (Produkt-)Eigenschaft, sind Lösungen Bestandteil des Realisierungsprozesses.<sup>10</sup>

---

<sup>6</sup> Vgl. [Eb2008], S. 4ff.

<sup>7</sup> Vgl. [Ha1993], S. 7.

<sup>8</sup> Siehe [Ie1990], Glossary.

<sup>9</sup> Siehe [Hu2007], Boehm zitiert nach van Husen, S. 33.

<sup>10</sup> Vgl. [Hu2007], S. 32.

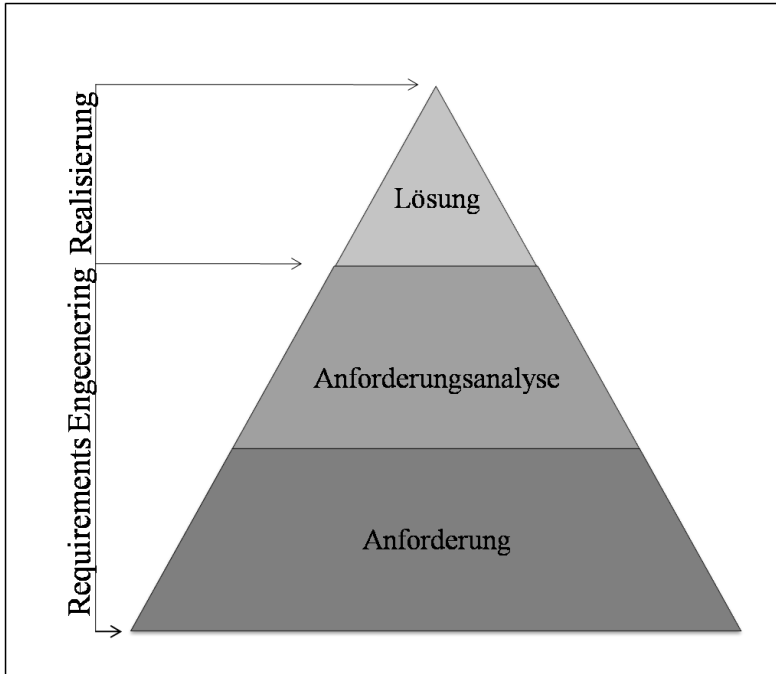


Abbildung 1: Funktionsumfang des Requirements Engineering

## 2. Wirtschaftswissenschaftliche Sicht

Die wirtschaftswissenschaftliche Sicht unterscheidet für die Definition von Innovationen drei Kategorien: a) Ziel, b) Mittel und c) Zweck.<sup>11</sup>

- a) **Ziel:** Das Ziel einer Innovation wird formuliert in einer Kundennutzen stiftenden Perspektive. Dabei kann es sich sowohl um Themen vorrangiger Einzelinteressen handeln, wie zum Beispiel die Verbesserung des Produktionsprozesses einer Unternehmung, als auch um gesamtgesellschaftliche Fragestellungen, wie die Stärkung der zivilen Sicherheit. Es handelt sich dabei um übergeordnete global gültige Zielformulierungen auf Metaebene.
- b) **Mittel:** Erreicht werden die vorangestellten Ziele mittels einer materiell-technischen Lösung, z. B. die Einführung einer neuen Hardware, oder eines immateriellen Lösungsansatzes, so zum Beispiel eines neuen Maßnahmenplanes zur Qualifizierung von Personal, an die im Rahmen der Mittelbeschreibung Lösungsansätze formuliert werden.

---

<sup>11</sup> Vgl. [Ha1993], S. 1-3.

- c) **Zweck:** Der Mitteleinsatz dient der funktionalen Zweckerfüllung. Damit werden konkrete mittelbezogene Endanwenderanforderungen des Lösungsansatzes beschrieben, sei es eine neuartige leistungsfähigere Hardware oder etwa eine vertiefte Stärkung der Sicherheit durch gezielte Schulungsmaßnahmen für Mitarbeiter im Service-Bereich im Themenfeld zur Stärkung der zivilen Sicherheit.

Daraus abgeleitet lässt sich folgende Systematisierung vornehmen:

		<b>wirtschaftsw. Sicht</b>		
ingenieurw. Sicht	<b>Kategorie</b>	<b>Ziel</b>	<b>Mittel</b>	<b>Zweck</b>
	<b>Anforderung</b>	-	-	<b>x</b>
	<b>Anforderungs-analyse</b>	-	-	<b>x</b>
	<b>Lösung</b>	-	<b>x</b>	-

**Legende:**  
**x** Übereinstimmung  
**-** keine Übereinstimmung

Abbildung 2: Vergleichende Darstellung Anforderung/ Innovation

Wie vorstehende Abbildung 2 verdeutlicht, folgt das ingenieurwissenschaftliche Vorgehensmodell nur bedingt der wirtschaftswissenschaftlichen Perspektive. Während die Wirtschaftswissenschaften dem Ziel eine eigenständige Bedeutung im Sinne eines gesellschaftlichen Telos zuweisen, wird es durch die Ingenieurwissenschaften rein kundenutzenorientiert hermeneutisch im Rahmen der Anforderungsanalyse abgeleitet; die Anforderung ordnet sich danach unmittelbar dem Ziel als abstrakte Vision unter, siehe Abbildung 3.

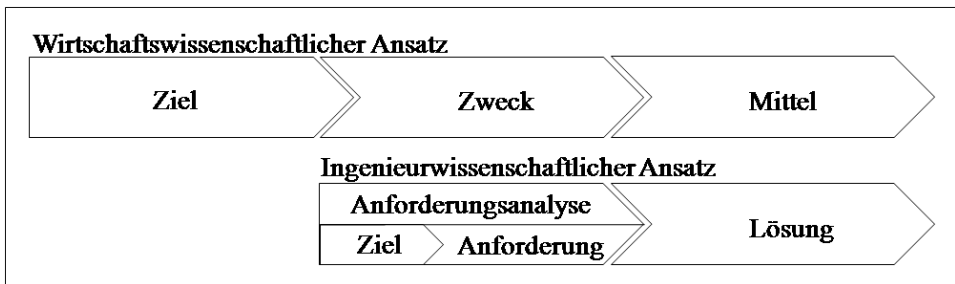


Abbildung 3: Vorgehensmodelle der Anforderungsanalyse

### **3. Anwenderforderungsanalyse im Rahmen des Forschungsprojektes SinoVE Management**

#### **3.1 Gegenstand des Forschungsprojektes**

Die Herausforderung im Projekt SinoVE Management besteht aus Sicht der Endanwender in der Stärkung der Sicherheitszentralen und Lagezentren durch intelligente Sicherheitssysteme für den Schutz von Personen und Vermögen, z. B. Lenkung von Menschenströmen in hoch frequenten Verkehrsstationen auch in Ausnahmesituationen, wie Großereignissen (Demonstration, Sonderzüge, WM etc.), unter Wahrung der Freiheits- und Persönlichkeitsrechte. Gleichzeitig ist auf die Möglichkeit einer wirtschaftlichen Verwertung der Ergebnisse zu achten.

Der Lösungsansatz des Projektes kann als Entwicklung eines vernetzten Sicherheits-Management-Systems mit integriertem Video- und Sensor-System nach neuestem Stand der Technik formuliert werden. Um eine größtmögliche Akzeptanz des Lösungsansatzes zu erreichen, ist auf das Verhältnis von Datenschutz gegenüber dem Sicherheitsbedürfnis der Infrastrukturnutzer zu deren Gunsten zu achten.

Neben der Bereitstellung der Verkehrsinfrastruktur und dem fachspezifischen Know-how, besteht der wesentliche Projektbeitrag der Endanwender in der Definition des Kataloges Anforderungen der Endanwender auf Szenario-Basis.

#### **3.2 Die Anforderungsanalyse in der praktischen Umsetzung**

Dass innerhalb des Projektes vorliegende Innovationsverständnis schließt einen über die Projektlaufzeit fortwährenden Wissenstransfer zwischen Anwendern und Entwicklern ein. Mit dieser sozialökonomischen Orientierung<sup>12</sup> wurde für die Anforderungsaufnahme und –analyse eine iterativ-inkrementelle Vorgehensweise gewählt.

Daraus folgend wurde zunächst eine übergreifende Projektbeschreibung erstellt, in der die Forschungsziele des Projektes als visionäre offene Sicht aller Beteiligten formuliert wurden. Gleichzeitig wurde damit eine greifbare, verbindliche Ausgangslage für die individuelle Antragstellung je Projektpartner geschaffen.

---

<sup>12</sup> Vgl. [Hi2005], S. 194-195.

Eine Konkretisierung der Endanwenderanforderungen auf Basis der Projektbeschreibung erfolgte mittels standardisierter, im Projekt abgestimmter Formulare. Auf diese Weise wurden 28 Einzelszenarios erfasst und für die Anforderungsanalyse innerhalb des Projektes zur Verfügung gestellt. Während in einem nachfolgenden Schritt fachliche Anforderungen, die zu technisch redundanten Lösungsansätzen führen, zusammengefasst wurden, ergab die Analyse der verbleibenden Szenarios zwei verschiedene Bewertungskategorien, die – obwohl auf gleicher Projektbeschreibung basierend – unterschiedliche Erwartungshaltungen zwischen Endanwendern und Forschungs- und Industriepartnern verdeutlichten: „Szenarios im Rahmen des Antrages“ der Forschungs- und Industriepartner und „Über den Antrag hinausgehende Forschungsleistungen“. Im Rahmen gemeinsamer Analyseworkshops wurde dabei deutlich, dass der Grund für diese nicht erwartete Abweichung darin lag, dass a) die Endanwender Anforderungen nicht auf Basis ange-dachter technischer Lösungsansätze formulierten, sondern als Grundlage einzig die übergreifende Projektbeschreibung heranzogen, und somit b) bestimmte Szenarios erst mit der fachlichen Expertise der Forschungs- und Industriepartner dahingehend identifiziert werden konnten. Kriterien für die Beurteilung waren Technikeignung – Ist die Zielerreichung mit einem Video-Managementsystem und Videosensorik möglich oder gibt es vorteilhaftere Lösungsansätze außerhalb des Forschungsprojektes? - und der forschungs-basierten Umsetzbarkeit – Wie realistisch ist das Erreichen des mit der Anforderung verbundenen Forschungszieles?

Die verbliebenen Szenarios wurden zunächst auf ihre übergreifende Zielkonformität geprüft und danach unterschieden in „Für den Meilenstein verfügbar“ und „Fertigstellung vsl. zum Projektende“, um eine inkrementelle Umsetzung zu ermöglichen. Insgesamt sind damit rund 14 Szenarios verblieben, die im Rahmen des Forschungsprojektes weiterhin betrachtet werden. Der ergebnisoffene Charakter eines Forschungsprojektes erfordert auch weiterhin die aufgeschlossene Sicht der Projektpartner auf die verbliebenen Szenarios. Daher ist die Anforderungsanalyse auch weiterhin Bestandteil der Arbeitspakete.

### **3.3 Fazit und kritische Würdigung**

Die bisherigen Erfahrungen aus Endanwendersicht verdeutlichen, dass ein iterativ-inkrementelles Vorgehensmodell einer rein dichotomischen Systematik vorzuziehen ist. Die aus der (Forschungs-)Praxis heraus anzuführenden Gründe, wie der Grad der Zielerreichung und die den Zweck erfüllenden Mittel, also die (Software-)Lösung, lassen sich nicht ohne ganzheitliche Kenntnis der gesellschaftlichen Aspekte des Projektzieles und der immanenten Innovationshemmnisse erkennen. Zur Problemvermeidung sollte daher ein fortwährender, kreislaufartiger Wissenstransfer für alle Beteiligte den Blick für die komplexen Herausforderungen schärfen, wie nachfolgende Abbildung zeigt.

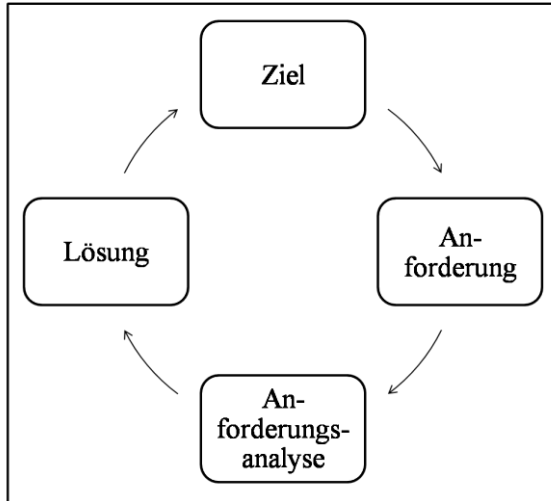


Abbildung 4: Innovationsprozess im Forschungsprojekt SinoVE Management

Gerade die aus dichotomischer Sicht für den Entwickler vernachlässigbaren gesellschaftlichen Implikationen können zu Wirkungen führen, die eigentlich funktionsfähige, einsetzbare Innovationen (z. B. Bodyscanner) ihrer Marktfähigkeit mit fatalen wirtschaftlichen Folgen berauben. Denn gesellschaftliches Handeln, also die Frage nach den Akteuren und deren (zum Teil gegenläufigen) Handlungsmustern, bieten einen Bedarf an Koordination, um weitgehend Konflikte zu vermeiden.<sup>13</sup> Daher wird zunächst die gemeinsame, übergreifende Projektzieldefinition für alle Beteiligten aus den positiven Erfahrungen des Projektes heraus unerlässlich. Die zunehmende Relevanz dieses Themas spiegelt sich in Ansätzen auch in der neueren (wissenschaftlichen) Literatur wieder.<sup>14</sup>

Und auch aus der Sicht der unmittelbaren (Produkt-)Innovation lohnt sich diese übergreifende Sicht für Anwender und Entwickler. Denn über die Spezifizierung der funktionalen Anforderungen einer (Software-)Lösung durch die Endanwender werden in der Regel Innovationshemmnisse sichtbar, für deren Auflösung ein iteratives Vorgehen zwischen Anwender und Entwickler unabdingbar ist. Wesentlich im Rahmen der Projektdurchführung identifizierte Innovationshemmnisse sind dabei subjektiv formulierte Anforderungen (Menschen mit unlauteren Absichten) sowie im Bereich der technischen Leistungsfähigkeit (das wird erst in ein paar Jahren in der Forschungspraxis lösbar sein).

<sup>13</sup> Vgl. [Hi2005], S. 13-17.

<sup>14</sup> Vgl. [Hu2007], S. 43-46.

Als Lösungsansatz für die Konfliktvermeidung im gesellschaftswissenschaftlichen Bereich der Zieldefinition ist ein verstärkter Rückgriff auf die sozialwissenschaftlichen Methoden (zum Beispiel Netzwerkanalyse, empirische Sozialforschung) und Erkenntnisse empfehlenswert, um (technischen) Innovationen eine breite Akzeptanz bei den Anwendern zu ermöglichen. Dieser interdisziplinäre Ansatz empfiehlt sich auch für den Bereich der technisch geprägten Innovationshemmnisse. So konnte eine nicht umsetzbare Produktinnovation in eine Prozessinnovation gewandelt werden. Zur Realisierung dieses Lösungsansatzes empfiehlt sich aus funktionaler praxisorientierter Sicht die strukturierte Vorgehensweise im Rahmen sogenannter *Mixed Projects*. Darunter versteht man die interdisziplinäre Zusammensetzung des Projektteams in fachlich-operativer und wissenschaftsübergreifender Hinsicht.

## Zusammenfassung und Ausblick

Der Einsatz innovativer Softwarelösungen im Kontext der gesamtgesellschaftlichen Entwicklungen ist nicht mehr wegzudenken. Daher sollten die gemeinsamen Anstrengungen von Endanwendern und Entwicklern verstärkt werden um beginnend mit einem gemeinsamen (Ziel-)Verständnis, zu einem einheitlichen Vorgehensmodell des Requirements Engineering zu gelangen. Die hier aufgezeigte Vorgehensweise in der Praxis bemüht sich, dabei auch darauf über Grenzen hinweg zu denken und Methoden und Erkenntnissen anderer wissenschaftlicher Disziplinen für eine nachhaltige Stärkung der Innovationsfähigkeit einzubeziehen. Nicht zuletzt ein aktiver Dialog zwischen den Endanwendern und (Software-)Entwicklern soll beiderseitig Vorteile eröffnen. So ist es auch künftige, bleibende Herausforderung, Methoden und Standards für eine verbesserte Innovationsfähigkeit zu entwickeln.

## Literaturverzeichnis

- [BH2009] Beaty, J; Hulgan, J.: Experiences with a Requirements Object Modell. In (Glinz, M; Heymans, P. (Eds.): Requirements Engineering: Foundation for Software Quality, Springer-Verlag, Berlin Heidelberg, 2009.
- [Bu2010a] BMBF. Staatliche Beihilfe Nr. N 330/2007 - Deutschland Forschung für die zivile Sicherheit: <http://www.bmbf.de/pub/n330-07.pdf>. Zitiert am 07.01.2010.
- [Bu2010b] BMBF. Forschung für zivile Sicherheit: [http://www.bmbf.de/pub/BMBF\\_Verkehrssicherheit.pdf](http://www.bmbf.de/pub/BMBF_Verkehrssicherheit.pdf). Zitiert am 07.01.2010.
- [Eb2008] Ebert, C.: Systematisches Requirements Engineering und Management. Dpunkt.verlag, Heidelberg, 2. aktualisierte und erweiterte Auflage, 2008.
- [Ha1993] Hauschild, J.: Innovationsmanagement. Verlag Franz Vahlen GmbH, München, 1993.
- [Hi2005] Hirsch-Kreinsen, H.: Wirtschaftssoziologie und Industriesoziologie. Juventa Verlag, Weinheim und München, 2005.
- [Hu2007] Husen, C. v.: Anforderungsanalyse für produktionsbegleitende Dienstleistungen. Jost-Jetter Verlag, Nr. 458, Heimsheim, 2007.
- [Ie1990] IEEE Standard 610.12.1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE, New York, NY, USA, 1990.
- [St2009] Standish Group, Chaos Reports: [http://www.standishgroup.com/newsroom/chaos\\_2009.php](http://www.standishgroup.com/newsroom/chaos_2009.php). Zitiert am 07.01.2010.





# Mobile Dienste für die Einbindung von Bürgern und Hilfeinsatzkräften<sup>1</sup>

Heiko Roßnagel

CT Informationsmanagement  
Fraunhofer Institut für Arbeitswirtschaft und Organisation (IAO)  
Nobelstr.12  
70569 Stuttgart  
heiko.rossnagel@iao.fraunhofer.de

**Abstract:** In vielen Metropolregionen finden immer häufiger Großveranstaltungen wie Straßenfeste, Sportveranstaltungen und Konzerte statt. Dabei müssen zum Teil enorme Besucherströme bewältigt werden. Aufgrund der hohen Personendichte können auch kleinere ungeplante Ereignisse enorme Auswirkungen haben und sich zu Krisen oder Notfällen entwickeln. Weiterhin macht dies Veranstaltungen auch zu einem attraktiven Ziel von Terroristen. Mobile Dienste können einen wertvollen Beitrag leisten, um diese Probleme zu reduzieren. Sie können sowohl bei der Durchführung von Veranstaltungen als auch im Bereich des Notfallmanagements eingesetzt werden. Um vernünftig auf Warnsignale ohne Verzögerungen reagieren können, ist es aber notwendig, dass die betroffenen Personen mit dem Notfallsystem vertraut sind. Um dies zu ermöglichen wird in diesem Beitrag eine einheitliche Plattform vorgestellt, die eine Integration von mobilen Mehrwertdiensten und Notfalldienstleistungen ermöglicht. Dafür werden zunächst Anforderungen an Notfallmanagementsysteme aus der Literatur abgeleitet und anschließend ein Konzept für die Integration von Notfall- und Mehrwertdiensten präsentiert und bewertet.

## 1 Einleitung

Durch die steigende Anzahl an Großveranstaltungen und kürzer werdende Vorlaufzeiten werden die Organisation und Durchführung immer komplexer und zeitkritischer. Beispielsweise können Schwierigkeiten durch unzureichenden Informationsaustausch zwischen den Verantwortlichen, fehlende Informationsweitergabe an Teilnehmer und Fahrgäste, mangelhafte Schulungen des angeworbenen Sicherheitspersonals, knappe Finanzmittel der beteiligten Institutionen, uneinheitliches Datenmanagement der einzelnen Einsatzzentralen, eingeschränkter Informationsaustausch im Krisenfall sowie einer späten Erkennung von Krisenfällen entstehen. Aufgrund der hohen Personendichte bei Großveranstaltungen können auch kleinere ungeplante Ereignisse enorme Auswirkungen haben und sich zu Krisen oder Notfällen entwickeln. Weiterhin macht dies Veranstaltungen auch zu einem attraktiven Ziel von Terroristen.

---

<sup>1</sup> Die Erarbeitung erfolgte innerhalb des durch das BMBF im Rahmen des Programms "Forschung für die zivile Sicherheit" als Teil der High-Tech-Strategie geförderten Forschungsprojekts VeRSiert

Notfallmanagementsysteme bieten eine Möglichkeit, dieses Dilemma zu adressieren. Sie ermöglichen eine Vorfallerkennung und -analyse und unterstützen die Einsatzkräfte beim Management der Folgen eines Katastrophenvorfalles. Dabei können die verantwortlichen Personen bei der Vorbereitung von Evakuierungen, Instruktion und Unterstützung von Einsatzkräften sowie bei der Lokalisierung von Opfern unterstützt werden [CT07]. Mobile Dienste können hierbei einen wertvollen Beitrag leisten. Mobilfunkinfrastrukturen bieten standardisierte drahtlose Kommunikationsdienste in nahezu allen Ländern an und ermöglichen eine schnelle Verbreitung von Informationen [GS08b]. So können Notfallmanager Warnungen in betroffenen Gebieten mittels Cell Broadcast verteilen und so potenzielle Opfer rechtzeitig benachrichtigen [FS05]. Der Erfolg eines Notfallmanagementsystems hängt aber stark von geübten Nutzern ab, die mit den Funktionalitäten der Dienste vertraut sind [TU04a]. Um zeitnah und zielgerichtet auf die Warnsignale reagieren können ist es notwendig, dass die betroffenen Personen mit dem Notfallsystem vertraut sind [GH89]. Dies ist nur sehr schwer zu erfüllen, wenn das System ausschließlich in Notfällen verwendet wird. Bei einem selten genutzten Notfallmanagementsystem können auch nur eingeschränkte praktische Erfahrungen erwartet werden [MH07]. Viele Dienste, die für Notfalldienstleistungen verwendet werden, unterscheiden sich technisch nicht wesentlich von Diensten, die für alltägliche Anwendungsfälle angeboten werden. So kann ein Foto-Upload sowohl dazu dienen, die aktuelle Situation effizient an einen Notfallmanager zu kommunizieren als auch dazu, Veranstaltungsfotos einer Online-Community zur Verfügung zu stellen. Ziel und Herausforderung ist es, solche grundlegenden Funktionalitäten so in ein System zu integrieren, dass Anwender mit dem Gebrauch solcher Funktionalitäten in unterschiedlichen Situationen vertraut sind. Eine Integration von Notfalldiensten und häufiger genutzten Mehrwertdiensten erscheint deshalb vorteilhaft. Sie würde einerseits die Vertrautheit der Nutzer mit dem System erhöhen und andererseits einen zusätzlichen wahrgenommenen Mehrwert für diese Kunden schaffen. Darüber hinaus könnte durch kommerzielle Mehrwertdienste auch ein Umsatz erzielt werden, der wiederum zur Refinanzierung der Notfallmanagementinfrastruktur genutzt werden könnte.

Der weitere Verlauf der Arbeit gliedert sich wie folgt. Zunächst werden Anforderungen an Notfallmanagementsysteme aus der Literatur abgeleitet. Anschließend werden verwandte Arbeiten darauf untersucht, inwieweit sie diese Anforderungen erfüllen und ein Systemdesign vorgestellt mit dem diese Anforderungen besser adressiert werden können, indem die Plattform auch für Mehrwertdienste geöffnet wird. Im folgenden Abschnitt wird die technische Umsetzung diskutiert bevor dann die Ergebnisse zusammengefasst werden.

## **2 Anforderungen**

Aus der Fachliteratur lassen sich die folgenden Anforderungen an Notfallmanagementsysteme ableiten, die eine zusätzliche Nutzung für andere Zwecke erlauben: (1) Effektivität, (2) Zuverlässigkeit, (3) Kosteneffizienz, (4) problemlose Dienstintegration, (5) mehrseitige Nutzerinteraktion, (6) Verfügbarkeit und (7) Sicherheit. Im Folgenden werden diese identifizierten Anforderungen genauer betrachtet:

**Effektivität:** Effektive Frühwarnsysteme sollten auf intensiv genutzten Kommunikationsinfrastrukturen aufgebaut werden und die betroffenen Personen sollten in der Lage sein, die gesendeten Warnsignale zu verstehen, um angemessen reagieren zu können [JO07]. Um diese Anforderung zu erfüllen, bietet sich der Einsatz bereits breit genutzter Infrastrukturen wie GSM an. GSM bietet eine standardisierte Kommunikationsmöglichkeit für Einsatzkräfte und Opfer, die weltweit von bereits mehr als 2 Milliarden Kunden verwendet wird [GS08a].

**Zuverlässigkeit:** Ein weiteres entscheidendes Erfolgskriterium ist die Zuverlässigkeit des Systems, was eine regelmäßige Wartung erfordert. Dies wiederum setzt eine regelmäßige Nutzung des Systems durch Einsatzkräfte voraus [KT02]. Darüber hinaus sollten auch potenziell betroffene Personen in vorbereitende Maßnahmen eingebunden werden. Warnsysteme, die nicht regelmäßig genutzt werden, werden häufig weder gewartet noch an sich ändernde Anforderungen angepasst [GH89].

**Kosteneffizienz:** Das System sollte über geeignete Schnittstellen verfügen, um existierende Informationsquellen nutzbar zu machen. Darüber hinaus sollte es die Nachrichten möglichst kostengünstig übermitteln [ZH05]. Größere Veränderungen an der Mobilfunkinfrastruktur und den mobilen Endgeräten der Nutzer sollten daher vermieden werden. Weiterhin sollte auch eine Sekundärnutzung des Systems durch zusätzliche kommerzielle Dienste unterstützt werden [ZE96].

**Problemlose Dienstintegration und Geräteunabhängigkeit:** Nutzer sollten in der Lage sein, das System benutzen zu können, ohne in neue Endgeräte investieren zu müssen. Hohe Anschaffungskosten führen häufig zu einer geringeren Nutzerakzeptanz [CH02] [RI04].

**Wechselseitige Interaktion:** Das System sollte eine wechselseitige Punkt-zu-Punkt Kommunikation erlauben, um beispielsweise Rettungskräfte vor Ort bei der Bildung dynamischer Teams zu unterstützen. Dies setzt voraus, dass das System eine dynamische bedarfsorientierte Allokation von Zugangskontrollrechten erlaubt [TU04b].

**Verfügbarkeit:** Die Gewährleistung der Verfügbarkeit ist ein bekanntes Problem von Systemen, die sich mit Katastrophen oder ähnlichen Ereignissen beschäftigen [FA01]. Katastrophen können direkte Auswirkungen auf die Kommunikationsinfrastruktur innerhalb des betroffenen Gebietes haben [ME07]. Wenn man aber auf der anderen Seite die Menge an vorhersagbaren Ereignissen, wie Stürme, Unwetter und zu einem gewissen Grade auch die Auswirkungen von Erdbeben, betrachtet, so zeigt sich, dass das Problem der Verfügbarkeit während eines Ereignisses weit weniger bedeutsam ist, da die rechtzeitige Warnung den zerstörenden Auswirkungen möglichst vorseilt. Zudem ist die Redundanz von Notfallsystemen wichtig, es sollte immer Alternativen geben.

**Sicherheit:** Neben der Verfügbarkeit müssen weitere sicherheitsrelevante Anforderungen erfüllt werden. Nutzer sollten in der Lage sein, die Integrität und Authentizität von Notfallwarnungen jederzeit zu verifizieren [VA04]. Weiterhin muss das System sicherstellen, dass keine Partei einen erfolgten Versand von Warnungen nachträglich bestreiten kann. Eine detaillierte Analyse, wie diese Sicherheitsanforderungen mittels mobiler elektronischer Signaturen erfüllt werden können, findet sich in [RS06].

### 3 Verwandte Arbeiten

Zahlreiche internationale Initiativen und Forschungsprojekte beschäftigen sich mit den Möglichkeiten, Mobilfunknetzwerke für Notfallmanagementsysteme einzusetzen. Die meisten dieser Initiativen konzentrieren sich dabei auf bestimmte Phasen einer Katastrophe, wie die Notfallfrüherkennung beziehungsweise die Vorhersage resultierender Schadensauswirkungen [MC02]. Andere Ansätze stellen die Frühwarnung in den Mittelpunkt, in dem sie Mechanismen für die Zustellung von Warnungen und Instruktionen an potenzielle Betroffene zur Verfügung stellen. Das vom Joint Research Center (JRC) der Europäischen Kommission ins Leben gerufene Global Disaster Alert and Coordination System (GDACS) hat zum Ziel, zeitnah Informationen über aktuelle, aus Naturgewalten resultierende Gefahren bereitzustellen [EC08]. Die von der Überwachung eingeschlossenen Ereignisse umfassen gegenwärtig Erdbeben, tropische Wirbelstürme, Vulkaneruptionen und Überschwemmungen, wie sie beispielsweise durch Tsunamis entstehen. Das System wertet Informationen aus unterschiedlichen Quellen aus, bewertet das resultierende Gefahrenpotenzial und stellt die Ergebnisse registrierten Nutzern, beispielsweise per SMS oder Email, zur Verfügung. Dieser Ansatz der Informationsübermittlung ist kosteneffizient, da bestehende Kommunikationsinfrastrukturen als Kommunikationsbasis dienen. Problematisch ist jedoch die Effektivität des Ansatzes zu bewerten: Warnungen werden ausschließlich an einen Personenkreis adressiert, der sich explizit für entsprechende Warnmeldungen angemeldet hat. Zudem enthalten zugestellte Warnungen keine Handlungsempfehlungen für Personen, die sich innerhalb der betroffenen Region aufhalten.

Eine niederländische Initiative verfolgt einen ähnlichen Ansatz und zielt darauf ab, Einwohner mittels SMS über kritische lokale Geschehnisse über die zuständige Polizeidienststelle zu informieren [KB07]. Wie bei GDACS werden keine Zusatzinformationen bereitgestellt. Es findet lediglich eine Filterung in Bezug auf bestimmte, für den Anwender relevante Regionen statt. In Bezug auf seine Effektivität lassen sich die bei GDACS genannten Schwächen ebenfalls aufführen. Die beiden aufgezeigten Ansätze nutzen bestehende IKT-Infrastrukturen und konnten somit kosteneffizient implementiert werden. Problematisch zu bewerten ist die einseitige Kommunikationsbeziehung, d.h. der fehlende Rückkanal über den aus den betroffenen Regionen nützliche Informationen zurückfließen könnten. Wie Studien zeigen, können entsprechende Mechanismen dazu beitragen, dass das Notfallmanagement zu besseren Entscheidungen gelangt [SU08] bzw. die Auswirkungen von Falschwarnungen reduziert werden können [RO91]. Die Anforderungen in Bezug auf Effektivität und Zuverlässigkeit erscheinen auf Basis der obigen Ausführungen verletzt, insbesondere weil die notwendige Nutzerregistrierung diesen Anforderungen entgegen steht. Die Sicherstellung der Verfügbarkeit stellt eine der großen Herausforderungen kritischer Systeminfrastrukturen dar und kein spezifisches Problem der aufgeführten Systeme. Im Zuge einer zugleich kommerziellen Nutzung der verwendeten Infrastrukturen ließen sich gleichwohl Systemredundanzen leichter realisieren.

## 4 Konzeption zur Integration von Notfall- und Mehrwertdiensten

Die vorgeschlagene Systemarchitektur baut auf dem Artikel von [FS05] auf, in dem die Frühwarnfunktionalität eines Notfallmanagementsystems beschrieben wird. Um kommerziellen Anbietern einen Zugang zu ermöglichen, wurden zusätzlich die Rollen Notfallmanager und Dienste-Anbieter definiert. Beide Entitäten agieren mit einer Untermenge von Verantwortlichkeiten und Funktionen des Gesamtsystems. Sie verfügen über einen Zugang zu der zugrunde liegenden Kommunikationsinfrastruktur und bieten darauf aufbauend spezifische Dienste an. Abbildung 1 zeigt, wie die einzelnen Parteien innerhalb des Systems interagieren und welche Dienste zur Verfügung gestellt werden.

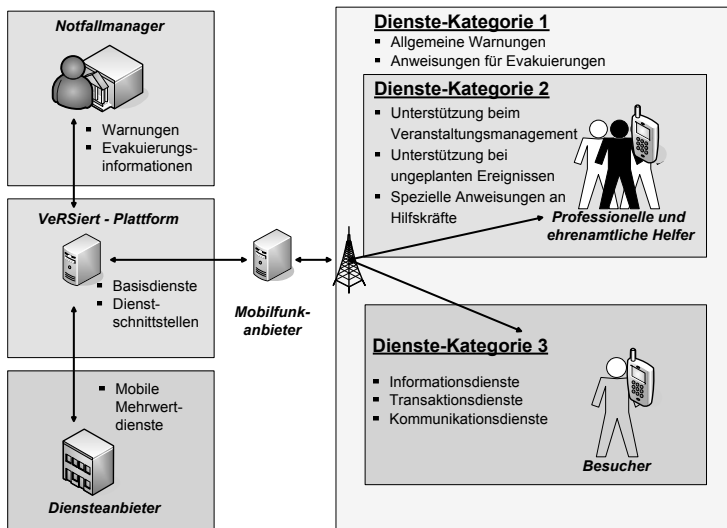


Abbildung 1: Illustration der Referenzarchitektur

Die zentrale Komponente ist eine Service-Plattform, die von einem Plattformanbieter betrieben wird. Diese Plattform kommuniziert mit den Mobilfunkanbietern und stellt Basisdienste mittels standardisierter Dienstschnittstellen zur Verfügung, die von Anbietern von Veranstaltungsdienstleistungen und Notfallmanagern gleichermaßen genutzt werden können. Der Plattformbetreiber kann eine öffentliche oder private Institution sein. Seine Hauptaufgabe ist es, die Systeminfrastruktur zu betreiben und Basisdienste über standardisierte Dienstschnittstellen den beteiligten Parteien zur Verfügung zu stellen. Mehrere unterschiedliche Institutionen kommen als Betreiber einer solchen Plattform in Frage. So könnte dies beispielsweise durch die Stadt Köln übernommen werden. Dadurch könnte es gelingen, das Veranstaltungserlebnis für die Besucher durch neue Mehrwertdienste aufzuwerten und gleichzeitig durch die Etablierung eines umfangreichen Notfallmanagementsystems potenziellen Besuchern ein Gefühl von Sicherheit zu vermitteln. Dabei muss die Stadt diese Plattform nicht zwingend selbst betreiben, sondern könnte sie an kommerzielle IT-Dienstleister auslagern. Darüber hinaus könnten auch Unternehmen mit einer starken lokalen Bindung, wie Nahverkehrsgesellschaften, diese Rolle übernehmen.

Selbstverständlich kommen auch kommerzielle Dienstleister als Betreiber in Frage, die die Nutzung der Basisdienste, die über die Plattform angeboten werden, mit einer Gebühr versehen [RO08a]. Der Notfallmanager ist verantwortlich für alle Notfalldienstleistungen, wie beispielsweise allgemeine Warnungen oder Evakuierungsanweisungen (Dienstkategorie 1). Darüber hinaus stellt er den Dienstleistern und ihren Angestellten Dienste zur Verfügung, die ihnen helfen, sich auf Notfallsituationen vorzubereiten und die sie während einer Notfallsituation unterstützen (Dienstkategorie 2). Für die Implementierung dieser Notfalldienstleistungen nutzt der Notfallmanager die Basisdienste, die von der Plattform bereitgestellt werden. Die an der Veranstaltung beteiligten Parteien wiederum bieten kommerzielle Dienste für Besucher (Dienstkategorie 3) an, auf die im folgenden Abschnitt genauer eingegangen wird. Auch diese Dienste werden aufbauend auf den Basisdiensten realisiert, was eine schnelle Entwicklung ermöglicht. Diese Dienste werden in aggregierter Form durch den Plattformbetreiber angeboten, der dadurch die Rolle eines Informationsintermediärs einnimmt. Durch diesen Ansatz kann die hier beschriebene Referenzarchitektur die Anforderungen an Effektivität, Zuverlässigkeit und Kosteneffizienz erfüllen, in dem die gleiche Infrastruktur für unterschiedliche Aufgaben eingesetzt werden kann. Das System ist damit kein Stand-Alone-System, das separat gewartet und angepasst werden muss.

## **5 Umsetzung Mobiler Mehrwert-, Notfall-, und Basisdienste**

Für die Umsetzung der mobilen Basis-, Notfall- und Mehrwertdienste wurde eine Serverinfrastruktur aufgebaut, die das Backend für die mobilen Dienste bildet. Diese besteht aus einem SMS-Gateway und einem Applikationsserver. Aufgabe des SMS-Gateways ist der Versand von SMS-Nachrichten. Das SMS-Gateway wurde unter Verwendung der Software „NowSMS/MMS-Gateway 2007 Edition“ umgesetzt. Diese Software bietet ein Web-Interface, über das Nachrichten mittels eines an den Rechner angeschlossenen Mobiltelefons versendet werden können. Die zweite Komponente der Serverinfrastruktur ist der Applikationsserver, auf dem mehrere Services wie z.B. ein XMPP-Chat Server laufen. Für die Darstellung ortsbezogener Dienste wurde eine Anbindung an OpenStreet-Map [OS10] implementiert. Neben der Schnittstelle für ortsbezogene Dienste wurde ein weiterer web-basierter Dienst für Microblogging entwickelt. Dieser Dienst ermöglicht es, ähnlich wie auch bei dem bekanntesten Microblogging-Dienst Twitter [TW10], Kurznachrichten mit bis zu 140 Zeichen in Echtzeit ins Web zu stellen. Nutzer dieses Dienstes können sowohl eigene Nachrichten einstellen als auch die Nachrichten anderer Teilnehmer lesen. Um die Nachrichten anderer Nutzer zu lesen, können die einzelnen Teilnehmer bestimmte andere Nutzer „abonnieren“ und bekommen dann auf ihrer Seite auch die Nachrichten dieser Nutzer angezeigt. Neben den web-basierten Diensten werden auch Dienste entwickelt, die eine Softwarekomponente auf dem Client voraussetzen. Hierfür wird eine Client-Anwendung für die Android-Plattform implementiert. Diese beinhaltet einen Chat- und einen Twitter-Client um sowohl eine synchrone als auch eine asynchrone Kommunikation zu ermöglichen. Darüber hinaus verfügt die Anwendung auch über einen Friend-finder-Dienst. Aufbauend auf dieser Kommunikationsinfrastruktur können nun Mehrwertdienste implementiert werden wie z.B. Angaben zu veranstaltungsspezifischen ÖPNV-Fahrplänen und verfügbaren Parkplätzen oder auch Hinweise zu Sonderfahrten und Ausfällen.

Bei Veranstaltungen, die sich über einen größeren Teil des Stadtgebiets erstrecken und ein überregionales Einzugsgebiet aufweisen, können auch stadtbezogene Informationsdienste wie Touristeninformationen und Stadtpläne von großem Nutzen sein. Weiterhin verfügt die Client-Applikation auch über einen Notfallknopf, mit dem eine automatische Ortung ausgelöst wird und der dem Benutzer die Möglichkeit bietet, Informationen zu dem Notfall an einen Notfallmanager weiterzuleiten.

## 6 Zusammenfassung

Mobile Dienste können einen wertvollen Beitrag im Bereich des Notfallmanagements leisten. Eine zentrale Voraussetzung ist aber, dass die betroffenen Personen mit dem Notfallsystem vertraut sind und vernünftig auf die Warnsignale ohne Verzögerungen reagieren können. Daher wurde in diesem Beitrag eine einheitliche Plattform vorgestellt, die eine Integration von mobilen Mehrwertdiensten zur attraktiveren Gestaltung von Großveranstaltungen und Notfalldienstleistungen ermöglicht.

## Literatur

- [CT07] Carver, L., und Turoff, M. 2007 Human Computer Interaction: The Human and Computer as a Team in Emergency Management Information Systems, Communications of the ACM, 50, 3, 33-38.
- [CH02] Chen, L., Gillenson, M. L., und Sherrell, D. L. 2002 Enticing online consumers: An extended technology acceptance perspective, Information & Management, 39, 8, 705-719.
- [EC08] European Communities (2008) Global Disaster Alert and Coordination System (GDACS) Homepage, <http://www.gdacs.org/>, 2008-11-21.
- [FA01] Faulkner, B. 2001 Towards a Framework for Tourism Disaster Management, Tourism Management, 22, 135-147.
- [FS05] Fritsch, L., und Scherner, T. 2005 A Multilaterally Secure, Privacy-Friendly Location-based Service for Disaster Management and Civil Protection. In Proceedings of the AICED/ICN 2005, Springer, Berlin, Heidelberg, New York, 1130-1137.
- [GH89] Grunfest, E., und Huber, C. 1989 Status report on flood warning systems in the United States, Environmental Management, 13, 3, 279-286.
- [GS08a] GSM-Association (2008) About GSM Associaton, [www.gsmworld.com/about/index.shtml](http://www.gsmworld.com/about/index.shtml), accessed 2008-04-24.
- [GS08b] GSMworld (2008) GSM Operators, Coverage Maps and Roaming Information, [www.gsmworld.com/roaming/gsminfo/index.shtml](http://www.gsmworld.com/roaming/gsminfo/index.shtml), accessed 2008-02-07.
- [JO07] Johnston, D., Becker, J., Gregg, C., Houghton, B., Paton, D., Leonard, G., und Garside, R. 2007 Developing warning and disaster response capacity in the tourism sector in coastal Washington, USA, Disaster Prevention and Management, 16, 2, 210-216.
- [KB07] Korteland, E., und Bekkers, V. 2007 Diffusion of E-Government Innovations in the Dutch Public Sector: The Case of Digital Community Policing. In Proceedings of 6th International Conference on Electronic Government (EGOV), M. Wimmer, H. J. Scholl, und A. Gronlund, Eds., Regensburg, 252-264.
- [KT02] Kron, W., und Thumerer, T. 2002 Water-related disasters: Loss trends and possible countermeasures from a (re-)insurers viewpoint. In Proceedings of the 3rd MITCH Workshop, Dresden, Germany.



- [MH07] Manoj, B. S., und Hubenko Baker, A. 2007 Communication Challenges in Emergency Response, *Communications of the ACM*, 50, 3, 51-53.
- [MC02] McEntire, D. A., Fuller, C., Johnston, C. W., und Weber, R. 2002 A Comparison of Disaster Paradigms: The Search for a Holistic Policy Guide, *Public Administration Review*, 62, 3, 267-281.
- [ME07] Mendonca, D., Jefferson, T., und Harrald, J. 2007 Collaborative Adhocracies and Mix-and-Match Technologies in Emergency Management: Using the emergent interoperability approach to address unanticipated contingencies during emergency response, *Communications of the ACM*, 50, 3, 45-49.
- [OS10] OpenStreetMap: Die freie Wiki-Karte, [www.openstreetmap.de](http://www.openstreetmap.de), accessed 2010-02-05
- [RI04] Ritchie, B. W. 2004 Chaos, Crises and Disasters: A Strategic Approach to Crisis Management in the Tourism Industry, *Tourism Management*, 25, 669-683.
- [RO91] Rosenthal, U., Hart, P., und Kouzmin, A. 1991 The Bureau-Politics of Crisis Management, *Public Administration*, 69, 211-233.
- [RS06] Roßnagel, H., und Scherner, T. 2006 Secure Mobile Notifications of Civilians in Case of a Disaster. In *Proceedings of the 10th IFIP Open Conference on Communication and Multimedia Security (IFIP CMS 06)*, H. Leitold and E. Markatos, Eds., Springer, Berlin Heidelberg, 33-42.
- [RO08a] Roßnagel, H., Engelbach, W., Frings, S., und Weisbecker, A. 2008a Mobile Dienste zur Erhöhung der Sicherheit bei Großveranstaltungen. In *Stuttgarter Softwaretechnik Forum 2008: Science meets Business*, D. Spath, O. Höß, und A. Weisbecker, Eds., Fraunhofer IRB Verlag, Stuttgart, 91-102.
- [RO08b] Roßnagel, H., Engelbach, W., und Frings, S. 2008b Ortsbezogene mobile Dienste zur Verbesserung der Sicherheit bei Großveranstaltungen. In *Tagungsband 5. Fachgespräch Ortsbezogene Anwendungen und Dienste*, J. Roth, Eds., Nürnberg, 35-40.
- [SU08] Sutton, J., Palen, L., und Shklovski, I. 2008 Backchannels on the Front Lines: Emergent Uses of Social Media in the 2007 Southern California Wildfires. In *Proceedings of the 5th International ISCRAM Conference*, F. Fiedrich, und B. Van de Walle, Eds., Washington D.C, USA.
- [TU04a] Turoff, M., Chumer, M., Hiltz, R., Klashner, R., Alles, M., Vasarhelyi, M., und Kogan, A. 2004a Assuring Homeland Security: Continuous Monitoring, Control & Assurance of Emergency Preparedness, *Journal of Information Technology Theory and Application (JITTA)*, 6, 3, 1-24.
- [TU04b] Turoff, M., Chumer, M., Van de Walle, B., und Yao, X. 2004b The Design of a Dynamic Emergency Response Management Information Systems (DERMIS), *Journal of Information Technology Theory and Application (JITTA)*, 5, 4, 1-36.
- [TW10] Twitter, [www.twitter.com](http://www.twitter.com), accessed 2010-02-05
- [VA04] Valtonen, E., Addams-Moring, R., Virtanen, T., Jrvinen, A., und Moring, M. 2004 Emergency Announcements to Mobile User Devices in Geographically Defined Areas. In *Proceedings of Information Systems for Crisis Response and Management (IS-CRAM)*, Brussels, Belgium, 151-156.
- [ZE96] Zeckhauser, R. 1996 The Economics of Catastrophes, *Journal of Risk and Uncertainty*, 12, 113-140.
- [ZH05] Zhao, S., Addams-Moring, R., und Kekkonen, M. 2005 Building Mobile Emergency Announcement Systems in 3G Networks. In *Proceedings of Communications and Computer Networks*, Marina del Rey, CA, USA.

# Modellierung von Prozessen in der Feuerwehrdomäne zur Identifikation von Informationsbedarfen<sup>1</sup>

Christian Lindemann<sup>2</sup>, Stephan Prödel<sup>3</sup>, Rainer Koch<sup>4</sup>

Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung (C.I.K.)  
Universität Paderborn, Fakultät Maschinenbau  
Warburger Str. 100  
33098 Paderborn  
S.Proedel@cik.uni-Paderborn.de  
lindemann@cik.uni-paderborn.de

**Abstract:** In dem Forschungsvorhaben Mobis Pro wird beabsichtigt, die Feuerwehren durch ein IT-System mit einsatzrelevanten Informationen zu unterstützen. Dazu müssen die individuellen Informationsbedarfe in den jeweiligen Phasen des Einsatzablaufes bekannt sein. In dieser Arbeit wird gezeigt, dass die Prozessmodellierung unter Verwendung von erweiterten Ereignisgesteuerten Prozessketten (eEPK) eine geeignete Lösungsmöglichkeit darstellt.

## 1 Einleitung

Mit einer zunehmend komplexer werdenden Umwelt steigen auch die Anforderungen an das technische Wissen, das die Feuerwehren neben ihrem Fachwissen benötigen, um Einsätze erfolgreich zu bewältigen. So bieten die stabilen Fahrgastzellen heutiger Autos einerseits eine erhöhte Sicherheit für die Passagiere bei Unfällen, andererseits versagt selbst modernes Rettungsgerät wenn nicht bekannt ist, an welchen Stellen es konstruktionsbedingt anzusetzen ist.

Alle erdenklichen Einsatzfälle und die dafür relevanten Informationen in Papierform zu sammeln und auf den Einsatzfahrzeugen der Feuerwehren mitzuführen ist schon auf Grund der resultierenden Menge der Dokumente unmöglich. Schon eine vollständige Liste aller möglichen Einsätze ist wegen der theoretisch unendlichen Anzahl von Kombinationen und Variationen einzelner Ereignisse unrealistisch. Selbst wenn man sich auf die häufigsten Einsatzfälle beschränken würde und alle relevanten Dokumente in elektronischer Form auf einem modernen Datenspeicher mitführen würde, bliebe das Problem der Datenaktualität.

---

<sup>1</sup> Dieses Paper basiert auf der Arbeit der Universität Paderborn im Projekt Mobis Pro im Rahmen des Förderprogramms SimoBIT des BMWi ([www.simobit.de](http://www.simobit.de)).

<sup>2</sup> Dipl.-Wirt.-Ing. Christian Lindemann ist Mitarbeiter am Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung.

<sup>3</sup> Dipl.-Ing. Stephan Prödel ist Mitarbeiter am Lehrstuhl für Computeranwendung und Integration in Konstruktion und Planung und Projektleiter im Forschungsvorhaben Mobis Pro.

<sup>4</sup> Prof. Dr.-Ing. Rainer Koch ist Leiter des Lehrstuhls für Computeranwendung und Integration in Konstruktion und Planung der Universität Paderborn

Um die Einsatzkräfte dennoch in jeder erdenklichen Schadenslage mit einsatzrelevanten Informationen zu unterstützen entwickelt die Universität Paderborn im Rahmen des Forschungsvorhabens Mobis Pro eine Mobilfunk gestützte Lösung zum „On Demand“ Zugriff auf an unterschiedlichen Standorten vorliegende Informationsquellen. Dabei wird auf Originaldaten im Besitz der jeweils für die Datenbestände verantwortlichen Organisationen, Ämter, Behörden oder Unternehmen zugegriffen um Redundanzen zu vermeiden und über die aktuellsten Informationen zu verfügen

Um den konkreten Informationsbedarf zu ermitteln und relevante Informationen durch ein zu entwickelndes IT-System in einem Einsatz verfügbar zu machen, wurden im Rahmen der Anforderungsanalyse die Abläufe in einer Feuerwehr modelliert und analysiert. Zielsetzung des vorliegenden Artikels ist, die Gründe darzulegen, die zur Auswahl von eEPKs als Modellierungsmethode führten und die Erfahrungen der Prozessmodellierung in der Domäne Feuerwehr zu schildern.

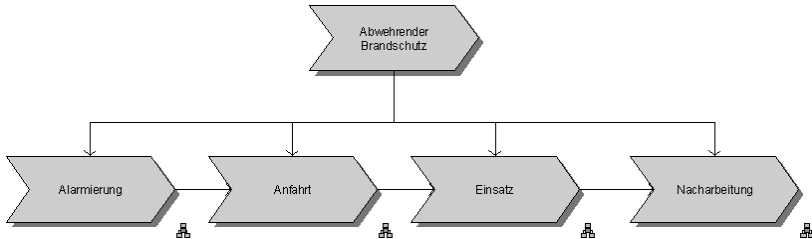
## 2 Informationsgenerierung aus Prozessmodellen

In der Einleitung wurde herausgestellt, dass die Einsatzkräfte kurzfristig auf einsatzrelevante Informationen angewiesen sind. Information wird hierbei nach Seifert definiert als: „[...] gegenwarts- und praxisbezogene Mitteilung über Dinge, die uns im Augenblick zu wissen wichtig sind“ [Se71]. Um eine spätere Informationsunterstützung der Einsatzkräfte zu ermöglichen, müssen die relevanten Informationen für verschiedene Arbeitsschritte im Vorfeld identifiziert werden. Im klassischen Wissensmanagement ist einer der ersten Schritte benötigtes Wissen zu identifizieren, relevante Prozesse zu verstehen und zu analysieren. Hierbei ist eine gewisse Fachkompetenz notwendig, um das Wissen bezüglich der Relevanz für die Geschäftsprozesse bewerten zu können [VDI09]. Um die relevanten Arbeitsabläufe zu verstehen und daraus die benötigten Informationen zu extrahieren und zu erfassen, bietet sich eine Modellierung der Prozesse der zu betrachtenden Unternehmung an.

Die Modellierung hat das Ziel die komplexen Abläufe in vereinfachter Form auf die wesentlichen Inhalte des Anwendungsbereiches zu reduzieren [GPW09]. Dies ermöglicht eine vereinfachte Sicht auf die verschiedenen Ebenen der Prozesse. Durch das gesammelte Verständnis bietet sich die Möglichkeit, Informationsbedarfe konkret in verschiedenen Arbeitsschritten zu betrachten. Ein weiterer Vorteil der Modellierung ist die Möglichkeit die Prozesse in standardisierter Form abzubilden. Hierbei wird der Spielraum für abweichende Interpretationen eingeschränkt. Somit wird ein gemeinsames Verständnis über die zu betrachtende Domäne ermöglicht [GPW09].

Ein Modell soll so einfach wie möglich, jedoch komplex genug sein, um die abzubilden Sachverhalte korrekt darzustellen. Um dies zu gewährleisten muss festgelegt werden auf welche Bereiche sich die Modellierung konzentriert, sowie der Zweck der mit dem Modell verfolgt wird. Im Rahmen von Mobis Pro soll eine Erhebung der Informationsbedarfe an spezifischen Prozessstellen erarbeitet werden. Ein besonderes Augenmerk fällt hierbei auf die Prozesse des abwehrenden Brandschutzes, der wiederum in vier Teilbereiche gegliedert wurde (vgl. Abb.1). In diesen Teilbereichen wurden wiederum

mehrere Systemebenen betrachtet. Zur Gliederung wurde hierfür ein Top down Ansatz verwendet. Um die Informationsbedarfe an verschiedenen Stellen zu identifizieren, bieten sich nun verschiedene Möglichkeiten an. Unterschieden wird hierbei zwischen subjektiven, objektiven und gemischten Verfahren [Kr05].



**Abbildung 1:** Prozessstruktur im abwehrenden Brandschutz

Um dem Modellgedanken weiterhin Rechnung zu tragen, bietet sich insbesondere die Methode der Prozessmodellierung als objektives Verfahren an [LS06].

Das Geschäftsprozessmanagement beschäftigt sich mit dem Herausfinden, Gestalten, Dokumentieren und Verbessern von Geschäftsprozessen. Ziel ist es, die in einem Unternehmen existierenden Informationen zu den eigenen Geschäftsprozessen zu nutzen, um sich auf den Kunden auszurichten und als Ergebnis die Unternehmensziele besser zu erreichen [Be08]. Auch im Requirements Engineering ist die Prozessmodellierung eine Methode zur Anforderungserhebung für Prozesse die mit IT-Systemen unterstützt werden sollen [GD06]. In der Literatur gibt es bis heute keine einheitliche Definition, was ein Geschäftsprozess ist. Daher wird folgende Definition verwendet:

*„Ein Geschäftsprozess ist eine zielgerichtete, zeitlich-logische Abfolge von Aufgaben, die arbeitsteilig von mehreren Organisationen oder Organisationseinheiten unter Nutzung von Informations- und Kommunikationstechnologien ausgeführt werden können. Er dient der Erstellung von Leistungen entsprechend den vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen. Ein Geschäftsprozess kann formal auf unterschiedlichen Detaillierungsebenen und aus mehreren Sichten beschrieben werden. [...]“* [Ga08]

Solche Prozesse finden auch im Einsatzfall der Feuerwehr statt. Hier grenzt sich die Domäne der Feuerwehr von der allgemeinen betriebswirtschaftlichen Organisation ab. In dem Bereich der BWL sind die Zielgrößen sehr vielfältig, allerdings hauptsächlich kostenorientiert, während diese im Rettungseinsatz eine eher untergeordnete Rolle spielen [Ha07]. Die Hauptziele der Feuerwehr-Domäne sind die Throughputtime (Zeit die zwischen der eingehenden Schadensmeldung bis zur Beseitigung der Gefahr benötigt wird), sowie die Response time (Zeit die zwischen einem eingehenden Notruf bis zum Eintreffen der Feuerwehren am Einsatzort vergeht). Die schnelle Bearbeitungszeit ist wichtig, um die Abwehr von Schäden an Personen, Tieren und der Umwelt zu gewährleisten. Diese Werte stehen vor dem Erhalt von Anlagewerten. Die zeitliche Variante ist mit Sicherheit der entscheidende Unterschied zwischen diesen beiden Domänen. Allgemein gültige Aussagen oder Empfehlungen zur Verwendbarkeit bestimmter Prozessmodellie-

rungstools können in der Regel nicht getroffen werden. Vielmehr hängt die Wahl der Methode vom Modellierungszweck und den gestellten Anforderungen hieran zusammen.

Der Markt bietet eine Vielzahl verschiedener Prozessmodellierungstools, was die Auswahl für die verschiedenen Anforderungen deutlich erschwert. Die Modellierungsmethoden gliedern sich auf in rein grafische Modelle zur Visualisierung (Visio, DIN 1966) und semi formale Modelle, wie die erweiterten ereignisgesteuerten Prozessketten. (eEPK). Des Weiteren gibt es Unterstützungswerkzeuge die stärker auf die informationstechnische Umsetzung abzielen, wie die unified modelling Language (UML) oder Modelle, die sich stärker darauf konzentrieren mathematische Ereignisse und Zustände zu modellieren, wie die Petri Netze [OL08].

### **3 Anforderungen an eine Modellierungsmethode für Feuerwehrprozesse**

Bevor mit einer Prozessmodellierung begonnen wird, muss der Betrachtungsgegenstand eingegrenzt werden. Dazu muss Klarheit darüber bestehen, was mit der Modellierung erreicht werden soll (Modellierungszweck). Das übergeordnete Ziel im Forschungsvorhaben Mobis Pro lautet: „Die erforderlichen Informationen zum relevanten Zeitpunkt an der richtigen Stelle“. Dazu müssen die handelnden Personen, ihre Tätigkeiten und die zur Erledigung ihrer Aufgaben notwendigen Informationen bekannt sein. Daher ergaben sich folgende Einzelziele:

- Darstellung von Aufbau- und Ablauforganisation einer Feuerwehr in standardisierter, bildlicher Form als Diskussionsgrundlage mit Domänenexperten und Aufbereitung für Domänenfremde
- Vereinheitlichung einer Mehrheit unterschiedlicher Einsatzfälle in einem Standardeinsatzablaufmodell
- Identifikation von Informationsbedarfen in den spezifischen Prozessschritten
- Verwendung des Standardmodells als Leitkomponente in einem IT-System

Im nächsten Schritt muss definiert werden, was modelliert werden soll (Modellierungsgegenstand). Die Feuerwehr hat Gefahren für die öffentliche Sicherheit und Ordnung abzuwehren, die durch Brände, Explosionen, Überschwemmungen, Unfälle und ähnliche Ereignisse entstehen (§ 1 Abs.1 Feuerwehrgesetz). Dazu gehören im Wesentlichen vorbeugende Maßnahmen (vorbeugender Brandschutz) und schadenabwehrende Feuerwehreinsätze (abwehrender Brandschutz und allgemeine Hilfeleistung) [Do03].

In diesen Bereichen tragen eventuelle Entscheidungen auch die weitreichendsten Konsequenzen, womit eine umfassende Informationslage von großem Nutzen ist. Daher wurden diese Bereiche für die Modellierung vorgesehen. Der letzte Schritt besteht in der Auswahl einer geeigneten Modellierungsmethode. Neben dem Modellierungszweck und den Anforderungen der Nutzer des späteren Modells müssen die spezifischen Eigenschaften des Modellierungsgegenstands berücksichtigt werden. Im Fall der Feuerwehr müssen dabei die Prozesse des vorbeugenden Brandschutzes von denen der Feuerwehreinsätze unterschieden werden. Die Prozesse des vorbeugenden Brandschutzes sind von

den gleichen Eigenschaften gekennzeichnet wie Verwaltungsprozesse. Die Eigenschaften dieser Prozesse und ihre Anforderungen an eine Modellierungsmethode haben **BECKER, ALGERMISSEN** und **NIEHAVES** in [BAN03] beschrieben. Die grundlegende Eignung von eEPK zur Modellierung von Verwaltungsprozessen wurde dort bestätigt.

Ein Feuerwehreinsatz unterliegt demgegenüber anderen Rahmenbedingungen. Die meisten Einsätze sind Notfalleinsätze, die direkt nach Eingehen des Notrufes beginnen und unverzüglich bearbeitet und koordiniert werden müssen. Der Entscheidungsträger ist einem hohen Zeitdruck ausgesetzt, was einen erheblichen Stressfaktor darstellt. Die Einsatzsituation wird im Wesentlichen von vier Merkmalen charakterisiert [Pu92]; [FK94]; [HW08], [BPK09]:

- Komplexität
- Vernetztheit
- Intransparenz
- Dynamik

Hieraus ergeben sich die folgenden Anforderungen an die Methode: (vgl. auch [SDL05] zum Einsatz von eEPK in der Krankenhausdomäne):

- Modellierbarkeit dynamischer Prozessabläufe, d. h. sequenzieller, paralleler und iterativer Abläufe.
- Darstellung von Entscheidungssituationen und des notwendigen Informationsbedarfs
- Berücksichtigung dynamischer Organisationsstrukturen im Falle von Änderungen der Alarmstufe

Weitere Anforderungen ergeben sich aus dem oben spezifizierten Modellierungszweck:

- Einfachheit (zur Diskussion mit Domänenexperten)
- Standardisiert (zur einheitlichen Modellierung, terminologische Konsistenz)
- Exportierbar, transformierbar (zur späteren Weiterverarbeitung in IT-Systemen)

## **4 Modellierung AB spezifischer Prozesse**

Für die Prozessmodellierung in Mobis Pro wurden die semi-formalen eEPK's eingesetzt, die ein wesentlicher Teil des ARIS Konzepts sind. Ereignisgesteuerte Prozessketten (EPK) bestehen aus Ereignissen und dadurch ausgelöste Funktionen, die durch logische Operatoren miteinander in Beziehung gesetzt werden. Die Möglichkeit der Verwendung von Organisations-, Daten- und Informationsobjekten ist durch die erweiterten Ereignisgesteuerten Prozessketten (eEPK) gegeben. Im deutschsprachigen Raum haben die eEPK's sich inzwischen als Standard zur Modellierung von Geschäftsprozessen herauskristallisiert [OI08]. Die Möglichkeit Prozesse in E-Government Projekten abzubilden, wurde schon mehrfach erfolgreich belegt [BAN03]. Auch zur Prozessmodellierung im Emergency Management wurden die eEPK's z. B. im ERMA Projekt [RPA09] oder bei

der Modellierung eines ÜMANV<sup>5</sup> [PR09] erfolgreich eingesetzt. Somit zeigen die Voruntersuchungen, dass die eEPK's ein gutes Potenzial besitzen, die Prozesse des abwehrenden Brandschutzes (AB) abbilden zu können.

Die Literatur empfiehlt eine eher rudimentäre Modellierung der Ist-Prozesse, um sich stärker auf die Soll-Prozesse zu konzentrieren [Be08]. Im Projekt Mobis Pro wurde direkt mit der Modellierung der Soll-Prozesse begonnen, da die Ist-Prozesse teilweise stark variieren. Für die Soll-Prozesse der Feuerwehr wird z. B. durch die FWDV100 bereits ein grober Handlungsrahmen gegeben. Input für die Modellierung lieferten die Konsortialpartner IFR<sup>6</sup> und Kreis Paderborn sowie die Literaturrecherche<sup>7</sup>.

Die oben genannten Anforderungen Einfachheit, Standardisierbarkeit und Transformierbarkeit werden durch die semi-formale Modellierungsmethodik der eEPK abgedeckt. Die Einfachheit der Darstellung wurde hierbei schon unter Beweis gestellt [BAN03].



**Abbildung 2:** Führungskreislauf der Feuerwehr Quelle: (FWDV100)

Die geforderte Standardisierbarkeit ist durch eine einheitliche Symbolik innerhalb der eEPK's gewährleistet. Der Export der Prozesse in die ausführbare Business Process Execution Language (BPEL) ist gewährleistet. Zwar gibt es Modellierungssprachen die den eEPK's gegenüber hier Vorteile besitzen, allerdings fällt dies nicht so stark ins Gewicht, da hier das eingesetzte Transformationstool erheblich zur Qualität der Umwandlung beiträgt[Kr08].

Auch die Methodenanforderungen konnten zufriedenstellend von den eEPK's abgedeckt werden. Als problematisch hat sich allerdings die Modellierbarkeit dynamischer sowie iterativer Prozessabläufe dargestellt (vgl. auch Sarshar). Diesem Problem wurde im Projekt Mobis Pro konstruktiv begegnet. Im Zentrum stand hierbei der Führungskreislauf der Feuerwehr nach FwDV100 (vgl. Abb.2), welcher einen in sich abgeschlossenen iterativen Prozess darstellt. Die Bedingung Lage feststellen ist Ausgangspunkt für alle

<sup>5</sup> Überörtliche Unterstützung beim Massenansturm von Verletzten

<sup>6</sup> Institut für Feuerwehr- und Rettungstechnologie der Stadt Dortmund

<sup>7</sup> Gesetzestexte, Dienstvorschriften, Standardliteratur u. a. aus dem Kohlhammer Verlag

weiteren Prozesse im AB. Somit kann das Prozessmodell hierdurch iterativ an die dynamisch veränderten Umweltbedingungen angepasst werden. Dies ermöglicht die schrittweise Adaption der Prozesse.

Informationen, die für die Ausführung von Prozessschritten notwendig sind, wurden anhand des fertigen Prozessmodells mit Domänenexperten für jede Rolle und jeden einzelnen Prozessschritt aufgenommen. In einem zweiten Modellierungsdurchgang konnten diese dann im Nachgang als Informationsobjekte in die eEPK integriert werden. Um die Informationsbedarfe bestimmter Rollen in einem Prozess zu identifizieren, wurden die eEPK's in sogenannte „swimlanes“ transformiert. Wie bei UML-Diagrammen, werden in swimlanes nur die Prozessschritte und die dazugehörigen Informationsbedarfe einer bestimmten Rolle abgebildet. Somit kann das Prozessmodell darstellen, welche Informationen die jeweiligen Rollen zu bestimmten Zeitpunkten benötigen.

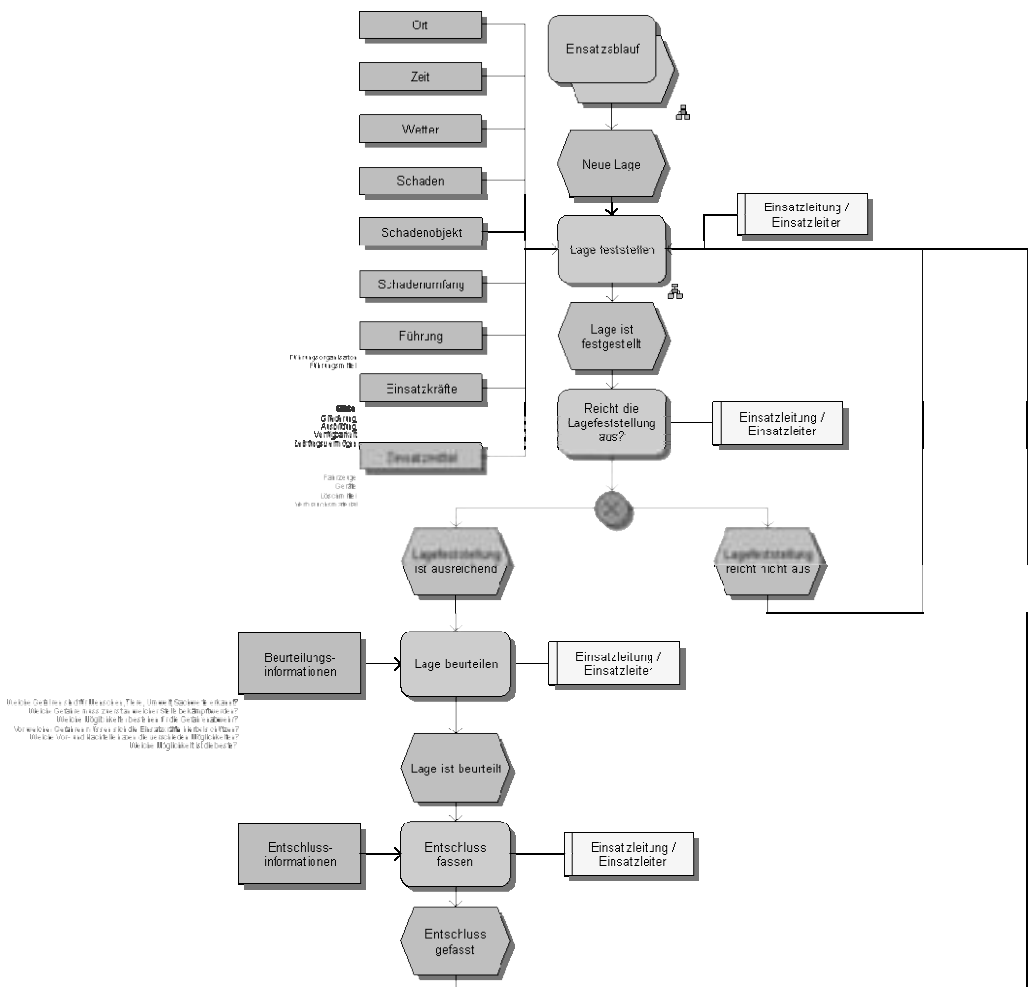


Abbildung 3: Ausschnitt Prozessmodell VB mit Soll-Informationen



## 5 Zusammenfassung und Ausblick

Die eEPK's haben sich in der Modellierung von Prozessmodellen für den abwehrenden Brandschutzes als tauglich erwiesen. Auch dynamische Prozesse konnten mit der Methode modelliert werden. Dies wurde jedoch durch konstruktive Maßnahmen erreicht. Hierbei wurde eine Schleife modelliert, die nach Ausführung bestimmter Unterprozesse immer wieder zu einer Lagefeststellung zurückkehrt. Somit kann dies als iterativer Prozess aufgefasst werden, der sich den geänderten Umweltsituationen anpasst. Zusätzlich hierzu besteht die Möglichkeit, wahlfrei sequenzielle Prozessabläufe zu modellieren. Dies wird durch einen Sequenz Konnektor (SEQ) realisierbar. Ähnlich der Funktion eines XOR-Konnektors ist die eEPK durch den SEQ in der Lage, Alternativen zwischen verschiedenen Handlungsalternativen abzubilden [SDL05]. Insgesamt wäre es für die gesamte Domäne der nichtpolizeilichen Gefahrenabwehr allerdings einfacher Prozessschritte zu modellieren, die im Sinn einer SOA die Darstellung frei konfigurierbarer Prozesse ermöglicht. Erste Ansätze, die sich mit dieser Problematik auseinandersetzen, werden bereits erforscht [LM08]. Nachdem die Modellierung der Prozesse des AB in der Theorie ihre Tauglichkeit bewiesen hat, steht nun noch die praktische Evaluation mit den Konsortialpartnern an, welche zugleich potenzielle Endanwender des Systems sind. Bei erfolgreichem Verlauf der Evaluation besteht die Möglichkeit, die modellierten Prozesse als Referenzmodell zu benutzen.

## Literaturverzeichnis

- [BAN03]Becker, J.; Algermissen, L.; Niehaves, B.: Prozessmodellierung in eGovernment-Projekten mit der eEPK. In (Nüttgens, M.; Rump, F. J. Hrsg.): EPK 2003. Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Proceedings, Hamburg, 2003;
- [Be08] Becker, Jörg (2008): Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 6., überarb. und erw. Aufl. Berlin: Springer.
- [BPK09] Birkhäuser, B.; Pottebaum, J.; Koch, R.: Unterstützung von Einsatzentscheidungen der Feuerwehr auf Basis IT-unterstützter Kräftekoordination. In (Fischer, S. Hrsg.): Informatik 2009. Im Focus das Leben. Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 28.9. - 2.10.2009 in Lübeck., Bonn, 2009; S. 1393-06.
- [Do03] Domke, J.: Rechtliche Grundlagen der Gefahrenabwehr. In (Rönnfeldt, J. Hrsg.): Feuerwehr-Handbuch der Organisation, Technik und Ausbildung. Mit 83 Tabellen. Kohlhammer, Stuttgart, 2003; S. 177–191.
- [FK94] Funke, J.; Kirk, M.: Schulung für den Katastrophenschutz: Psychologische Aspekte der Ausbildung und des Trainings. In Notfallvorsorge und Zivile Verteidigung - Internationale Zeitschrift für Gefahrenabwehr, 1994, 25; S. 22–25.
- [Ga08] Gadatsch, Andreas (2008): „Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker“; Wiesbaden: Vieweg & Sohn Verlag, 5. Auflage.
- [GD08] Groß, Anette; Dörr, Jörg (2006): Experimenteller Vergleich zweier Notationen zur Prozessmodellierung : Ereignisgesteuerte Prozessketten vs. UML-Aktivitätsdiagramme. Fraunhofer Institut für experimentelles Software Engineering. Kaiserslautern.
- [GPW09]Gausemeier, Jürgen; Plass, Christoph; Wenzelmann, Christoph (2009): Zukunftsorientierte Unternehmensgestaltung. Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. München, Wien: Hanser.

- [Ha07] Hajo A. Reijers, Monique H. Jansen-Vullers, Michael zur Muehlen, Winfried Appl (2007): Workflow Management Systems + Swarm Intelligence = Dynamic Task Assignment for Emergency Management Applications, Heidelberg
- [HW08] Hacker, W.; Weth, R. v. d.: Denken – Entscheiden – Handeln. In (Badke-Schaub, P.; Hofinger, G.; Lauche, K. Hrsg.): Human Factors. Psychologie sicheren Handelns in Risikobranchen ; mit 17 Tabellen. Springer Medizin Verl., Heidelberg, 2008.
- [Kr05] Krcmar, Helmut (2005): Informationsmanagement. Berlin: Springer.
- [Kr08] Kruczynski, Klaus: Prozessmodellierung im Wettbewerb : EPK vs. BPMN. In: Betriebswirtschaft Business Process Management, Jg. 2008, Ausgabe 06/2008, S. 30–35.
- [LM08] La Rosa, Marcellano; Mendling, Jan (2003): Domain-driven Process Adaption in Emergency Scenarios. In mecella, Massimo and Yang, Jida, Eds. Proceedings Business Process management Workshops 2008, Milan, Italy.
- [LS06] Lehner, Franz; Scholz, Michael (2006): Wissensmanagement. Grundlagen, Methoden und technische Unterstützung. München: Hanser.
- [OI08] Olbrich, Sebastian (2008): Modellierung gesetzlicher Rahmenbedingungen für Verwaltungsprozesse aus dem E-Government. Univ., Diss.--Marburg, 2007. 1. Aufl. Wiesbaden: Gabler
- [PR09] Peinel, G.; Rose, T.: Prozessmodellierung für das Notfallmanagement. In (König-Ries, B.; Erfurth, C. Hrsg.): Kurzbeiträge zum Workshop „IT-Unterstützung von Rettungskräften“ am 1.10.2009 (Lübeck). im Rahmen der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI) INFORMATIK 2009, Jena; S. 5–9.
- [Pu92] Putz-Osterloh, W.: Entscheidungsverhalten. In (Frese, E. Hrsg.): Handwörterbuch der Organisation. Schäffer-Poeschel, Stuttgart, 1992.
- [RPA08] Rose, Thomas; Peinel, Gertraud; Arsenova, Emilija: Process Management Support for Emergency Management Procedures, eChallenges e-2008 Conference, 2008, Stockholm, Sweden
- [SDL05] Sarshar, K.; Dominitzki, P.; Loos, P.: Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne. Eine empirische Methodenevaluation. In (Nüttgens, M.; Rump, F. J. Hrsg.): EPK 2005. Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Proceedings, Hamburg, 2005; S. 97–117.
- [Se71] Seiffert, H. (1971): Information über die Information: Verständigung im Alltag, Nachrichtentechnik, wissenschaftliches Verstehen, Informationssoziologie, das Wissen der Gelehrten. (3. Auflage). München: Beck
- [VDI09] Wissensmanagement im Ingenieurwesen. Grundlagen, Konzepte, Vorgehen = Knowledge management for engineering : Fundamentals, concepts, approach. März 2009. (2009). Berlin: Beuth (VDI-Handbuch Produktentwicklung und Konstruktion, 5610, 1).





**3rd International Workshop on Social Software  
Engineering  
(SSE 2010)**



# 3rd International Workshop on Social Software Engineering

Martin Ebner      Imed Hammouda      Hans-Jörg Happel      Walid Maalej  
Wolfgang Reinhardt

**Abstract:** Software is created by people, with people and for people. These people work in varying environments. They have their particular backgrounds and act under different conditions. Thus understanding the human and social aspects of software engineering is crucial to understanding how methods and tools are used, and thereby improving the creation and maintenance of software systems as well as the management of software projects.

“Social Software Engineering” (SSE) focusses on the development of systems in highly uncertain domains, with evolving goals, frequent changes and much user involvement. SSE systems can often (but not exclusively) be found on the web. Related technical concepts are user feedback, mashups, perpetual beta. However, beneath technical expertise developing social software systems requires competency from other disciplines as diverse as psychology, organizational science or economics. Besides the advent of social software applications, software engineering research recognized in recent years that effective collaboration and knowledge sharing are essential in order to guarantee successful software development and maintenance. Methods and tools that support development teams must be based on interdisciplinary research efforts that investigate technologies, tools, processes and human factors in a holistic manner.

While both directions - engineering social software and dealing with social aspects in the software engineering process - receive considerable attention, we think that ultimately, both might confluence into a new software engineering paradigm. This workshop provided a forum for discussing high quality research on the social aspects of software engineering and the engineering aspects of social software, as well as a meeting place for the community that is currently distributed over several research domains (software engineering, knowledge management, web 2.0, human computer interaction).

## 1 Motivation and Goals of the workshop

In this workshop we brought together researchers and practitioners working on different aspects of collaboration and knowledge sharing in software engineering as well as the engineering of social software to discussed new results and future research challenges. Major topics addressed at the workshop included:

- Social and human aspects of software engineering
  - Collaboration and knowledge sharing in development teams and (Open Source) communities
  - Impact of Social Software on development processes

- Empirical studies on collaboration and information behavior in social software engineering
- Engineering social software
  - Engineering of lightweight and unobtrusive tools, Web 2.0 and Social Semantic Web applications
  - Approaches and tools for context-aware and personalized assistance
  - Particularities in the development of Social Software
- Social Software Engineering
  - Concerns of individuals in collaboration settings, such as learning, usability and incentives
  - Usage of Social Software to teach software engineering, teaching social aspects of software engineering
  - Research methods and approaches for analyzing and designing successful collaboration support
  - Scientific analysis of the relation between methods/processes, tools and collaborative development practice

## 2 Programme

### Session I: Social Tools

- Jean-Marie Favre and Marc Quast: *Towards Social Information Systems*
- Swapneel Sheth, Nipun Arora, Christian Murphy and Gail Kaiser *weHelp: A Reference Architecture for Social Recommender Systems*
- Alexander Sahm and Walid Maalej *Switch! Recommending Artifacts Needed Next Based on Personal and Shared Context*

### Session II: Social Requirements Engineering

- Steffen Lohmann and Thomas Riechert: *Bringing Semantics into Social Software Engineering: Applying Ontologies to a Community-oriented Requirements Engineering Environment*
- Alexander Felfernig, Philipp Ghirardini, Monika Mandl and Monika Schubert *Diagnosing Inconsistent Requirements Preferences in Distributed Software Projects*
- Norbert Seyff and Florian Graf *User-Driven Requirements Engineering for Mobile Social Software*

## Keynote

- Bernd Brügge, Technische Universität München: *Opportunities for Social Software in Large-Scale Project Courses*

## Open Space

- Important topics were identified and discussed in small groups
- Final round table discussion

## 3 Program Committee

- Andreas Auinger, University of Applied Sciences, Austria
- Mohamed Amine Chatti, RWTH Aachen, Germany
- Jan Bosch, Intuit
- Björn Decker, Empolis GmbH, Germany
- Alexander Felfernig, TU Graz, Austria
- Tobias Hildenbrand, University of Mannheim and SAP AG, Germany
- Michael Koch, Bundeswehr University Munich, Germany
- Filippo Lanubile, University of Bari, Italy
- Francesco Lelli, University of Lugano, Switzerland
- Steffen Lohmann, University of Duisburg-Essen, Germany
- Johannes Magenheimer, University of Paderborn, Germany
- Michael Mlynarski, Software Quality Lab / University of Paderborn, Germany
- Dirk Riehle, University of Erlangen, Germany
- Uwe Riss, SAP AG, Germany
- Klaus Schmid, University of Hildesheim, Germany
- Sulayman K. Sowe, United Nations University, The Netherlands
- Markus Strohmeier, Institute of Knowledge Management, TU Graz, Austria





# Towards Social Information Systems

Marc Quast  
University of Grenoble  
France  
marc.quast@imag.fr

Jean-Marie Favre  
OneTree Technologies  
Luxembourg  
jean-marie.favre@megaplanet.org

**Abstract:** The development and usage of complex information systems leads to both technical and human challenges as large numbers of stakeholders with conflicting requirements are involved. Though global consistency must be preserved at the corporate level, software applications have to be adapted to particular needs. However, while individual users have the best knowledge about how to perform their job, they have little influence on the corporate information system. Our experience in industry shows that the “Knowledge/Influence Mismatch” leads to the “Information System Fragmentation” problem: departments, teams or individuals unsatisfied by corporate information systems tend to develop “parallel” ad-hoc applications. To cope with this problem this paper proposes core concepts for Social Information Systems in which much more power will be given to user communities, allowing them to extend existing applications for their particular needs, but also to share these extensions with colleagues; and this in a social way. This approach, based on the notion of “perspectives” linked to people, opens the possibility of a social adaptation and democratic evolution of the information system.

## 1. Introduction

After the “invention” of writing, which marks the move from prehistory to history and is the first information technology, the “invention” of software is perhaps one of the greatest achievements in the history of mankind [1]. Though we still lack a full understanding of what software actually is, let us consider a few facts.

*#1: Software is BY and FOR people.*

Software is clearly a creation of people for the benefit of people. To be more precise (1) software is *developed* BY people (a.k.a. *developers*) and (2) software is FOR the *usage* of people (a.k.a. *users*). As we will see both the development and usage of software can be considered from a social perspective.

*#2: Software runs ON computers.*

This comes at no surprise, but as suggested by the very terms “Computer Science“ and “Computer Languages“ this fact has been hiding fact #1 for long.

This paper is based on the experience gained during twenty years of development, integration, deployment and maintenance of *Information System Applications* (as defined below) for various multinational corporations, mostly in the high-tech industry.

- An **Information System** (IS) is a set of software applications running in a particular organization (a few applications to hundreds of them). We define IS as software. A broader definition would encompass people, documents, etc.
- An **Organization** is a corporation, an administration, or big department of either. The size of the organization may vary from a dozen of people to many thousands.
- An **Application** is any data-centric software system enforcing some process of the organization or helping it being more productive, or both.
- A **Stakeholder** is a person whose job is somehow affected by the IS. Beyond users this includes administrators, managers, developers, partners, etc.

This paper focuses on one particular challenge: while an IS should ideally capture all information and processes of an organization in a consistent way, the high number of stakeholders often produces conflicting requirements [2]. Although Applications typically provide adaptation mechanisms, we think these are limited and do not provide adequate support for serious conflicts, nor for business agility [3, 6]. We believe new technologies are needed to empower users with social adaptation of Applications and propose enabling concepts for what we call *Social Information Systems*.

A running example will be used throughout the paper. Its simplicity does not reflect the size and complexity of the problem we address, but it exhibits some of its main characteristics. It assumes an on-premise deployment model, but we consider that our proposal also applies to more recent multi-tenant Software-As-A-Service [13] models.

The Example Use Case
<ul style="list-style-type: none"><li>• Company ProjectSoft, a medium-sized software vendor, develops a product called ProjectTracker which provides enterprise project management. This product is installed in more than 100 companies world-wide.</li><li>• ACME Corporation is a large company which develops both electronics boards and companion software for industrial installations. ACME, a customer of ProjectSoft, uses ProjectTracker for all engineering projects, whether hardware assembly, software development or other activities.</li><li>• The hardware and software departments have different requirements. However, ACME chooses to share one installation of ProjectTracker (reasons are license costs, people working for both departments, common projects, consolidated dashboards, ease-of-administration, etc...).</li></ul>

The remainder of the paper is structured as following. Section 2 presents a conceptual view of a common adaptation approach based on decision layers. The limitations of this approach are described in Section 3 and Section 4 sketches the principles of an alternative approach more respectful of the social dimension. Section 5 presents the notion of perspective coupled with actors as a basis for implementing Social Information Systems. Related work is described in Section 6, further work in Section 7 and Section 8 concludes the paper.

## 2. Decision Layers in the Application lifecycle

This section, based on the observation of industry practices, provides a conceptual view of the application lifecycle (see Figure 1 for an overview). Our objective here is not to describe a particular technology but to illustrate the layers in the decision process and to define the vocabulary for the following sections.

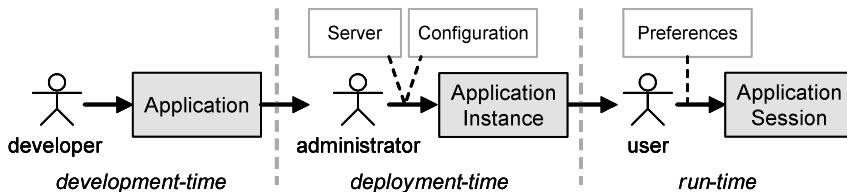


Figure 1. High-level overview of the Application lifecycle

### 2.1 Development-time – development of Applications

At development-time, an Application can be considered a collection of coarse-grained **ApplicationElements** like **Concepts** (representing entities of the problem domain and structural information, ensuring basic consistency), and **Features** (forms, reports...). Some ApplicationElements are configurable, i.e. designed to be further refined at deployment- and/or run-time. The level of configurability varies from very low (typical for in-house ad-hoc software developments) to fairly high for generic applications like Product Lifecycle Management systems.

Applications usually define **Permissions** restricting access to ApplicationElements, thus defining logical application subsets. The resulting application reflects the decisions of a (typically fairly small [2]) *software development group* (in the broad sense, including domain experts, ...).

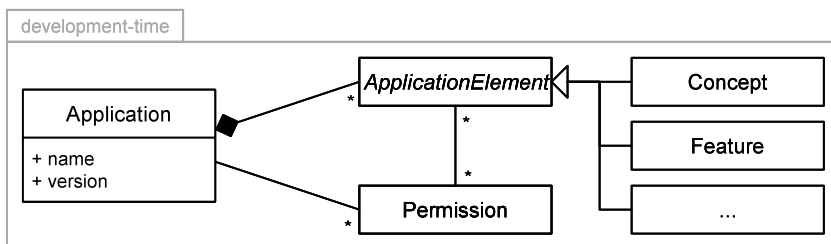


Figure 2. Applications at development-time

#### Example

- "ProjectTracker 1.2" is an *Application*.
- It provides *Concepts* like "Project", "Task", and "Resource". These can be extended with an ad-hoc mechanism ProjectSoft calls "custom-attributes"
- It provides various *Features* like "TaskScheduler", "GanttDiagramBrowser", etc.
- Its Permissions are {create,read,update,delete}-{Project,Task,...}

## 2.2 Deployment-time – configuration of ApplicationInstances

At deployment-time, the goal is to solve a specific problem for a given organization. An **ApplicationInstance** is created by physically installing an Application on a **Server** infrastructure and by using ApplicationElement configurability to adapt to the specific purpose. Actual **Actors** (i.e. Users or Groups) are granted Permissions. ApplicationInstances reflect the decisions of yet another small group of *software integrators* and *administrators* - constrained by the earlier decisions of developers.

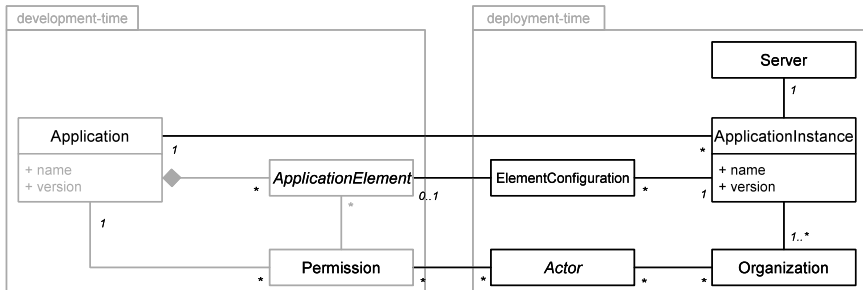


Figure 3. ApplicationInstance at deployment-time

### Example

- The “ACME Engineering Project Referential” installed on “http://project.eng.acme.com/” is an *ApplicationInstance* (using “ProjectTracker 1.2”)
- Several Concepts have been extended using the custom-attribute mechanism
  - A custom-attribute “customer” has been added to Concept “Project”
  - For software activities, a custom-attribute “type” with possible values {SPECIFICATION, DEVELOPMENT, TEST, OTHER} has been added to Concept “Task”
  - For hardware activities, it is required to track machine-time consumption, leading to several other custom-attributes and a dedicated usage report
- Several roles have been set up, for example Role “technical-staff” has Permission to see all “Project” and “Task” attributes except “\*Cost” and cannot see “Resources”

## 2.3. Run-time – execution of end-user Sessions

At run-time, a User connects to an ApplicationInstance establishing a **Session**. A Session filters available ApplicationElements according to Permissions, and can provide a final, typically shallow layer of **Preferences**.

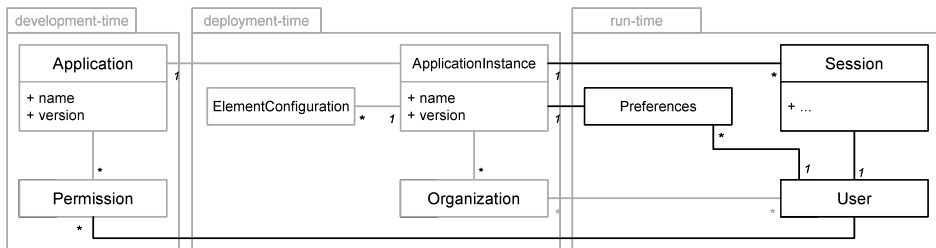


Figure 4. Sessions at run-time

### Example

- Fred is a software architect, with role “technical-staff”. His Preferences hide all “Task” attributes except “name” and “duration”
- Betty is the director of sales. Her Preferences show only “name” and “customer” attributes for “Projects”, plus all DATE attributes

## 2.4. Summary

Users reach the functionality of an Application through a number (possibly higher than in our example) of successive adaptation layers, the intent of which is to gradually fit a generic solution to a specific problem; all commercial-off-the-shelf applications we have encountered implement some variant of this approach. The main benefits are cost-reduction through factorization (re-use), and consistency (the adaptation at each level is constrained by what has been decided at the lower levels). However, the approach presents serious limitations as described in the next section.

## 3. The Knowledge/Influence Mismatch and IS Fragmentation

The problem we have observed with adaptation layers is that the most important decisions are taken by the people farthest away from the real-life problems the application needs to solve. Though end-users have the best *knowledge* of their information processing needs they have little *influence* on the software applications they work with. By contrast the developers have a tremendous influence on the application, although they usually have insufficient knowledge on the details about application usage in particular contexts. We refer to this phenomenon as the **Knowledge/Influence Mismatch**. Agile methodologies [5, 7] have alleviated this problem by reducing the cycle-time through the process in Figure 1, but have not solved it.

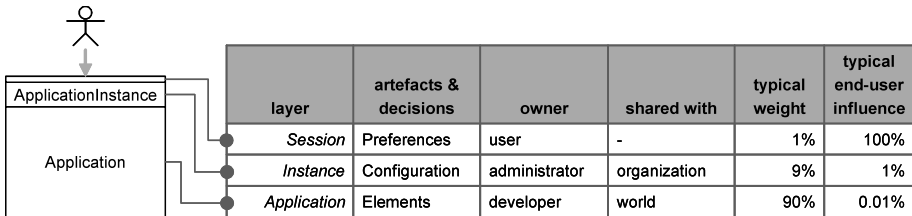


Figure 5. Typical weight of decision layers versus end-user influence<sup>1</sup>

The diminishing “end-user influence” figure is not only related to distance to key development roles but also to the fact that the lower layers are shared with a high number of other stakeholders. This sharing is the root cause for the difficulty of *requirements engineering*. Ideally, requirements analysis should be a highly *social* activity [2], bringing together all stakeholders, letting them exchange their requirements

<sup>1</sup> The weight and influence figures roughly represent our experience and are not backed up by empiric data

and expectations and converging on a common solution. Even with present communication technologies, this is an impossible feat, hence the common practice of expecting a subset of representatives to faithfully express requirements of other stakeholders, and as illustrated previously, trusting a few central roles to consolidate this into a single, consistent, satisfactory whole.

The results are that during each iteration, numerous requirements are excluded from both the Application and the Instance layers, that serious compromises on requirements are commonplace and that over time this leads to an overall state of disappointment and resignation of Users with their unsatisfactory information system [6, 7].

We think *Information System Fragmentation* is a direct consequence of this situation. High-end fragmentation happens when dissatisfied, resourceful actors develop their own ad-hoc systems as described in [8], where they have full control over the developer (i.e. influence on the Application) and a reduced number of stakeholders (i.e. fewer or no conflicts). At the low end of the spectrum, spreadsheets are commonly used to store critical data outside of mainstream applications.

Example
<ul style="list-style-type: none"><li>• Fred likes to complete tasks with highest risk first. He keeps a list of his tasks in a spreadsheet, with columns "Task.title", "Task.risk", "Task.notes"</li><li>• Betty needs a history of project delays to help her sales staff negotiate with customers. She has taken a summer intern to develop a small intranet system on sales.acme.com which extracts project data from "project.eng.acme.com" (via ProjectTracker web-services), adds some sales-specific data and computes nice indicators. This application has quickly become vital for the sales team</li></ul>

We believe that multiple adaptation layers are possible without sacrificing influence nor generating conflicts, by introducing a social dimension as described in the next section.

#### 4. An alternative distribution of responsibilities

Actors should be fully empowered to adapt an application to their specific problem, instead of having to rely on external factors like the availability and willingness of central roles and having to reach agreements and compromises with other Actors. As a related example of such a change in the social dynamics of groups, wikis have replaced the formerly powerful, central "webmaster" role with an infrastructure where anybody can freely contribute.

Beyond mere technical platforms, wikis represent a real shift of responsibilities. They imply *trusting the user community* to do what is right for the overall system. This trust should in theory be natural in an IS context because all users are authenticated (typically through an enterprise directory), and most of them are experienced professionals.

However, we think it is unrealistic to expect an organization (community) to agree and converge on all application elements. Divergence of points-of-view is unavoidable in complex information systems [2, 9], and we think that unlike both existing applications and wikis, actors should have a private space, following the sandbox principle.

This implies isolating Actors from each other by default, as opposed to implicit sharing. Sharing with other Actors, both in- and out-bound, should happen explicitly when desired. We can thus summarize the high-level requirements for a new approach as follows.

Actor Bill-Of-Rights		
	Any Actor can ...	... as opposed to having to ...
1	choose a subset of available ApplicationElements	daily wade through elements not relevant to his problem
2	extend / add ApplicationElements	appeal to central authorities, comply with more influent peers and/or build a parallel system
3	share whatever he owns with other selected Actors	
4	override any value	extract values and manipulate them outside the application

Independently of aforementioned usability, some end-users might not have the skills to exercise the rights above. However, we believe granting everyone the permission is far more interesting than the present opposite situation.

### 5. Towards Social Information Systems

We define **Social Information Systems** as ISs built with the *explicit* social dimension in mind with *dedicated* software technology. In this section we present one of the core concepts of our proposal, namely the introduction of the concept of Perspective as a central building block of Social Information Systems.

A **Perspective** defines a specific point of view for one Actor. It defines some ApplicationElements (for either private use or shared with other Perspectives), imports other relevant ApplicationElements and possibly extends them. In our proposal, the concepts of Application and ApplicationInstance are unified into a collection of ApplicationElements specific to a Perspective. Extensions provide a generalization of Configurations.

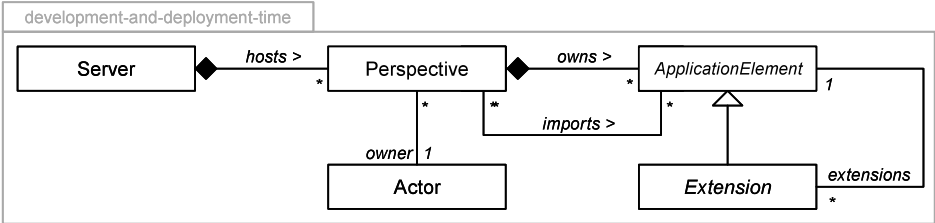


Figure 6. Perspective-centric application meta-model

Example
<ul style="list-style-type: none"> <li>• “http://project.eng.acme.com/” now hosts an “EngineeringProject” <i>Perspective</i>, importing several (but not all) elements from “ProjectTracker 1.2”</li> <li>• “SoftwareEngineering” is now a specific <i>Perspective</i>, which extends Concept “Task” as previously, clearly distinct from the “HardwareEngineering” concerns</li> </ul>



When considering social aspects, the notion of **Group** naturally appears. We therefore introduce it explicitly in our model as a refinement of the concept of Actor. In a social context Users typically belong to several Groups (Figure 7). As Actor is the generic term for both concepts, Perspectives are linked not only to Users but also to Groups (Fig. 6).

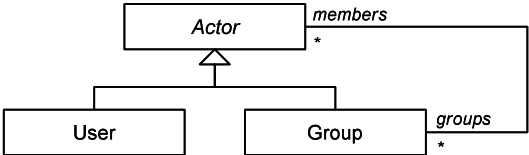


Figure 7. Actors, Groups and Users

It is important to understand that while both Groups and Users are involved at deployment time, at run-time only Users run Sessions. The process of building a session is not trivial, but below is a simplified view that may help in understanding the concept of Perspective. When a User, member of multiple Groups, connects and thus initiates a session, this involves the following steps.

1. fetch the User’s Perspective plus those associated to all of his Groups,
2. fetch all ApplicationElements and related Extensions from these Perspectives,
3. weave (consolidate) these elements into ComposedElements (figure 8).

The whole process results in a Session which constitutes indeed a *virtual private application* built on-the-fly.

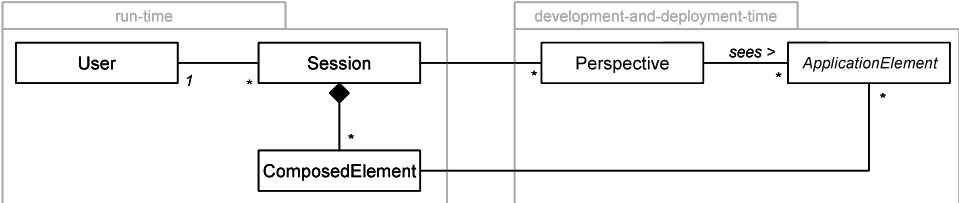


Figure 8. Session and Perspective at run-time

<b>Example</b>
<ul style="list-style-type: none"> <li>• Fred now defines his own Perspective               <ul style="list-style-type: none"> <li>○ It extends “Task” with columns “risk” and “notes”, replacing his former spreadsheet</li> <li>○ Other architects are interested in Fred’s risk-driven approach, and choose to adopt (i.e. import) his “risk” extension</li> </ul> </li> <li>• Betty’s department has its own “sales” Perspective too               <ul style="list-style-type: none"> <li>○ The summer intern has developed an ApplicationElement for the sales Perspective, which re-uses Project data directly</li> <li>○ Customer is now a full-blown concept owned by the sales Perspective</li> <li>○ The other fields and indicators are now mostly extensions of ProjectTracker concepts</li> </ul> </li> </ul>

Perspectives thus unify development-time artifacts, deployment-time configurations and run-time preferences, providing a successive refinement capability without Knowledge/Influence Mismatch or conflicts.

## 6. Discussion and related work

Our approach empowers users to *contribute* to the IS and to its evolution. Similarly to web 2.0 systems, users could interact through extending, tagging, rating, and sharing application elements. We believe that this would allow organizations to –continuously– leverage their *collective intelligence* and would enable *social software evolution*, opening new opportunities for social interactions and social schemes. For instance, in the context of open communities, one can imagine an *anarchic software evolution* where everybody can contribute to the same application without any supervision but with the opportunity to share elements through auto-organization. As in all social systems, some more structured organizations typically arise at some point leading for instance to a *meritocracy*. In even more structured contexts and in particular in business organizations, one can imagine *democratic software evolution* supported by the implementation within Social Information Systems of voting mechanisms [2]. We believe that many social schemes that characterize existing social systems could be applied in the software world, especially with appropriate supporting technologies.

The notion of perspective is clearly connected to the notion of viewpoints. However, the study of viewpoints [9] is mostly focused on *reconciling* divergent concerns at the specification and design level. Our proposal takes a complimentary bottom-up approach, allowing and even encouraging divergence in the implementation itself. Private elements allow *all* Actors to implement their specific requirements without complex mappings, lengthy negotiation or forking new ad-hoc Applications. We believe this encourages creativity and exploration, allowing new aspects to *emerge* from the bottom up.

Although our proposal in essence describes technical principles, the strong emphasis on people and groups ties it to the fields of collective intelligence [10] and complex adaptive systems [11].

The work presented here also relates to End-user development, a domain which has been studied intensively [8,12]. We believe that our proposed private elements could help by isolating Actors from one another's experiments. Also, the data-centric nature of ISs should make them less challenging to extend by end-users than more function-centric software; even without full end-user programming capability, we think end-user configuration and modeling would already provide huge benefits.

It is not clear how willing end-users would be to extend applications. We think this will essentially be related to ease-of-use, and are aware that end-user manipulation of the data, the underlying model and the views (forms, reports) represents a major usability challenge.

One can also wonder how *consistent* the resulting IS would be. We could argue that the consistency of most information systems is only superficial. In our experience, corporate-level applications, forming the tip of the iceberg, are usually consistent among themselves. This would not be different in our approach. The remainder however, hidden today in numerous parallel ad-hoc applications, is inconsistent by construction. Worse, it is out-of-reach for analysis. We think an infrastructure with a built-in divergence capability allows for processing to detect convergence opportunities [4,9] and notifying actors of similar-looking extensions, hence our evolutionary development [7] claim.

We also think our proposal fits within a natural trend, since the history of programming languages (PL) has been driven by social considerations [1]. For instance modular languages have been invented to support the development of software by *many people*, some of them using module interfaces, others providing their implementation. Similarly, the keywords “public” and “private” of object-oriented (OO) programming languages (and in the case of C++ the term “friend”) belong to terminology used in social contexts. In OOPL these terms refer to the relationships between ApplicationElements, and are mostly used at development-time. Perspectives go one step further by explicitly relating ApplicationElements to people (Actors, figure 6), and by utilizing these relationships at run-time as well.

## 7. Prototyping and Future work

Building upon the concepts presented in this paper, we are implementing a prototype of a distributed data-centric social infrastructure with the following high-level components.

- a *directory* with users and groups, annotated with references (URLs) to their associated Perspectives,
- a Perspective-centric *repository*, hosting class definitions (types) and their extensions as well as their associated instances (values),
- a *browser* communicating with both to compose simple virtual private applications allowing data and model manipulation.

Communication is asynchronous, standards-based, and resource-centric (REST architectural style [14]). The goal of this prototype is to validate the technical feasibility of our proposal, and to conduct field experiments in both academic and industrial organizations.

From a more theoretical point of view, we are extending the object-oriented paradigm with social concepts through the notion of Perspective. We think *Social Objects* are the building blocks of Social Information Systems.

## 8. Conclusion

In this paper we have introduced the notion of Social Information Systems. We have showed that the Knowledge/Influence Mismatch leads to users' frustration and thus to IS fragmentation. Our proposed solution is to empower users to contribute to the IS enabling *social software evolution*. We have provided a high-level description of an isolation mechanism called Perspectives to make this possible. Although more work is needed to assess the feasibility of our proposal, we are confident it is a track worth exploring and think that sharing these early ideas is a form of social software engineering in itself.

## 9. References

1. J.M. Favre, D. Gasevic, R. Lämmel, A. Winter, "Introduction to the Special Issue on Software Language Engineering", *IEEE Transactions On Software Engineering*, November/December 2009.
2. S. Lohmann, S. Dietzold, P. Heim, N. Heino, "A Web Platform for Social Requirements Engineering", *Software Engineering 2009 – Workshopband*, Köllen, Germany, 2009.
3. A. Wensley, E. van Stijn, "Enterprise Information Systems and the Preservation Of Agility", *Agile Information Systems*, 178-187, Elsevier, 2007.
4. N. Ahmadi et al, "A Survey of Social Software Engineering", *23rd IEEE/ACM International Conference on Automated Software Engineering - Workshops*, 1–12, 2008.
5. K. Beck et al, "The Agile Manifesto", <http://agilemanifesto.org/> - visited January 2010.
6. S. Newell, E. Wagner G. David, "Clumsy Information Systems : A Critical Review of Enterprise Systems", *Agile Information Systems*, Elsevier, 2007.
7. P. J. Denning, C. Gunderson, R. Hayes-Roth, "Evolutionary System Development", *Communications of the ACM*, vol. 51, 12, 29-31, December 2008.
8. W. Harrison, "The Dangers of End-User Programming", *IEEE Software*, vol. 21, 04, 5-7, July/August 2004.
9. M. Sabetzadeh, A. Finkelstein, M. Goedicke, "ViewPoints", *Encyclopedia of Software Engineering*, P. Laplante, Ed. New York: Taylor and Francis, 2010.
10. J. Surowiecki, "The Wisdom of Crowds", ISBN 978-0385503860, 2004.
11. [http://en.wikipedia.org/wiki/Complex\\_adaptive\\_system](http://en.wikipedia.org/wiki/Complex_adaptive_system) - visited February 2010.
12. A. Sutcliffe, "Evaluating the costs and benefits of end-user development", *ACM SIGSOFT Software Engineering Notes*, vol 30, July 2005.
13. K. Bennett , P. Layzell , D. Budgen , P. Brereton , L. Macaulay , M. Munro, "Service-based software: the future for flexible software", *Proceedings of the Seventh Asia-Pacific Software Engineering Conference*, December 2000.
14. Roy T. Fielding et Richard N. Taylor, "Principled design of the modern Web architecture," *ACM Transactions on Internet Technology*, no. 2, 115-150, 2002.



# weHelp: A Reference Architecture for Social Recommender Systems

Swapneel Sheth, Nipun Arora, Christian Murphy, Gail Kaiser

Department of Computer Science, Columbia University, New York, NY 10027

{swapneel, nipun, cmurphy, kaiser}@cs.columbia.edu

**Abstract:** Recommender systems have become increasingly popular. Most of the research on recommender systems has focused on recommendation algorithms. There has been relatively little research, however, in the area of generalized system architectures for recommendation systems. In this paper, we introduce *weHelp*: a reference architecture for social recommender systems - systems where recommendations are derived automatically from the aggregate of logged activities conducted by the system's users. Our architecture is designed to be application and domain agnostic. We feel that a good reference architecture will make designing a recommendation system easier; in particular, weHelp aims to provide a practical design template to help developers design their own well-modularized systems.

## 1 Introduction

In the past few years, recommender systems have become increasingly popular and ubiquitous. Recommendation systems are being used in a variety of different domains such as suggesting movies we might enjoy watching [Neta], things we might want to buy [Ama], music we may like [Pan], and people we may know [Fac], often based on community-driven “people like you . . .” paradigms. There has been keen public interest in improving such systems' recommendations, such as the Netflix Prize [Netb]. In the academic community, there has been much research on recommender systems, mostly focused on common issues of recommender systems such as recommendation algorithms [ZP07, GM08, PT08], implications of social networks in recommendation systems [ZC08, GDM<sup>+</sup>08] and security and privacy issues [BEKR07, MA07]. We have found relatively little research to date, however, in the area of system architectures for recommendation systems outside specific domains; work such as [GNOT92, RIS<sup>+</sup>94] has focused on their own architectures and implementations, without aiming to propose a general-purpose reference architecture, as we do here.

In this paper, we introduce *weHelp* - a reference architecture for social recommender systems. A reference architecture here is a collection of best practices and acts as a blueprint for building applications. Our architecture is designed to be independent of any specific application or domain. Our reference architecture is targeted towards community-driven recommender systems. weHelp systems are community-driven in the sense that the recommendations are derived automatically from the aggregate of logged activities conducted by the system's users and thus reflect community interests and/or behaviors. Commer-

cial examples of such recommender systems include Amazon [Ama] and Netflix [Neta], which use recorded user activities such as buying goods and renting movies, respectively, to recommend what might be a useful product to buy or an interesting movie to watch. We contrast social recommendation systems with what we call “knowledge driven recommendation systems”, such as ACE [Mei88] or [MFS09], which are essentially rule-based systems and do not take into account the activities of the users. Such systems – which might be referred to collectively by a term like “iHelp” – do not learn by observing the activities of the users, but instead rely on a hard-coded set of rules and are, hence, relatively static. *weHelp* social recommender systems, in contrast, are dynamic as they learn (and generally over time refine) their recommendations by observing their users.

The main contribution of this paper is our *weHelp* reference architecture. A good reference architecture should make designing a new recommendation system simpler. *weHelp* is intended to provide a template to help developers design a well-modularized system.

## 2 Background and Motivation

The authors are collectively working on three otherwise unrelated research projects that share a similar theme: providing suggestions aimed to help users employ particular software tools in some better way. From that ongoing body of work we reverse engineered the common components and structure that led to *weHelp*. Our goal is to present a generic reference architecture for recommender systems that will provide a good design template and act as a blueprint for building recommender system applications. The following paragraphs give some background on the three systems prototyped as part of our projects: *genSpace*, *Retina*, and *COMPASS*. These three systems were designed independently, and were not intended *a priori* to exhibit a common architecture. But we then realized that the architectures for these systems are very similar, and we have distilled a common architecture and the best practices that we discovered along the way into our reference architecture.

The **genSpace** project [MSKW08] is being conducted in collaboration with Columbia University’s Center for Computational Biology and Bioinformatics (C2B2). Researchers at C2B2 have developed *geWorkbench* [CFKW], which is an open-source Java-based system for integrated genomics targeted to biomedical researchers. As *geWorkbench* includes more than 50 plugin tools for genomics data analysis and visualization, it can be very daunting for a new user who does not know which tools to use with which data set, the order in which to use these tools, *etc.* A recommender system can be particularly beneficial for such users. At the same time, *genSpace* can also be very useful for experienced users, as it can provide insights about their peers’ *geWorkbench* usage that they may not be aware of. The main goal of *genSpace* is to provide collaborative filtering and knowledge sharing features to *geWorkbench* users through recommendations presented via social networking metaphors such as “people like you ...”. The *genSpace* module of *geWorkbench* records certain aspects of user activities and these are used to generate recommendations. Example recommendations include suggesting which analysis most people perform next given that they had started with the analysis most recently completed by this user, and listing the most commonly used workflows (series of analysis tools) and most popular tools.

We are separately investigating community-driven recommendation in the domain of computer science education. Our **Retina** system [MKLH09] is targeted towards CS1 (intro to programming) courses. Because students in these classes have little prior programming experience on which to draw, they may be unable to accurately estimate how long a programming assignment should take to complete, or how to address the often cryptic compiler and runtime error messages that they encounter. Thus, a recommendation system that is targeted to their individual needs could be very effective in helping them to be more efficient, and spend more time focused on algorithmic thinking and problem solving than on syntax and exceptions. Retina collects objective observational data about students' programming activities. Once data has been collected and aggregated, Retina makes real-time recommendations to students as they are working on their assignments. Example recommendations include warning the given student if she is encountering an especially high number of errors per compilation or if she is spending too long on the assignment. Retina also supports longer term "organizational memory", enabling it to make suggestions to students about what to expect from the assignment, for instance how much time to expect to spend on it or what errors to look out for. Last, Retina provides informative reports based on the aggregation of that data. These reports allow instructors to answer such critical questions as "how long are students taking to complete the programming assignments?", or "what sorts of errors are they making?" so that upcoming lectures can be revised accordingly.

Our third recommender system project, **COMPASS** (Community Driven Parallelization Advisor for Sequential Software) [SAG<sup>+</sup>09] proposes to leverage aspects of social networking to suggest multi-core (multi-threading) optimizations to programmers. COMPASS assumes a base of expert users who will parallelize existing sequential code using one or more of the numerous techniques for parallelization. The code changes are recorded, summarized, and stored in a central database. When an inexperienced user wishes to parallelize her code, COMPASS will first identify the regions of code most warranting performance improvement, and then show a list of potential optimizations mined from previous parallelizations, *i.e.*, matching previously observed "before" serial code to retrieve the corresponding "after" parallel code - thus propagating community knowledge.

### 3 weHelp Reference Architecture

A Reference Architecture [HH00] is a collection of best practices for a certain domain. It is a design template and acts as a blueprint for building applications. Reference Architectures also define a common vocabulary with the goal of standardizing different independent implementations. Reference Architectures usually define different modules and specify the functionality of each module along with the interaction and interfaces between the different modules. There have been many examples of reference architectures in varying domains such as scientific workflow management [LLF<sup>+</sup>09], and web servers [HH00]. In this paper, we define a reference architecture for social recommender systems.

The weHelp Reference Architecture is shown in Figure 1. It consists of three main components: the *Watcher*, the *Learner*, and the *Advisor*. All of these components have their



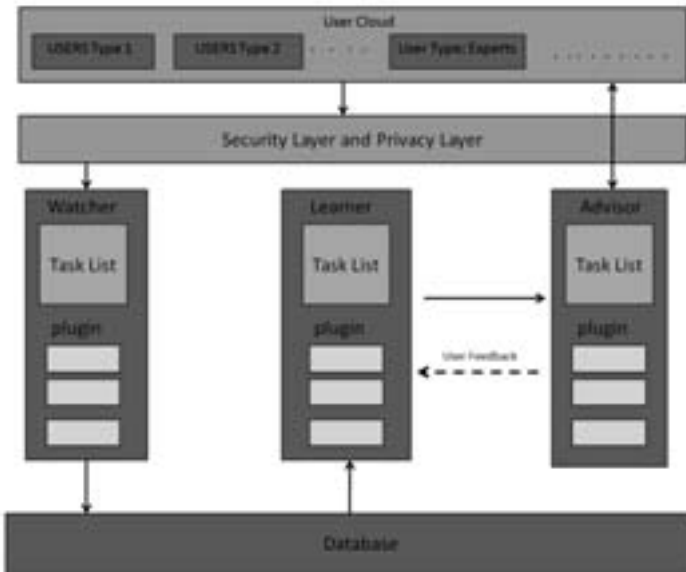


Figure 1: weHelp Reference Architecture

<p><b>Watcher</b></p> <ul style="list-style-type: none"> <li>● Monitor users' activities</li> <li>● Generate a representation of the user activities</li> <li>● Periodically send user activity logs to database</li> <li>● Optional Plugins: <ul style="list-style-type: none"> <li>◇ Allow users to provide extra domain specific information</li> <li>◇ Allow users to rate certain activities</li> </ul> </li> </ul> <p><b>Learner</b></p> <ul style="list-style-type: none"> <li>● Infer patterns and/or recommendations</li> <li>● Optional Plugins: <ul style="list-style-type: none"> <li>◇ Different ranking algorithms and constraint-based search</li> <li>◇ Generate overall statistics</li> <li>◇ Weigh user data</li> <li>◇ Optimizations such as caching</li> </ul> </li> </ul> <p><b>Advisor</b></p> <ul style="list-style-type: none"> <li>● Provide recommendations using different UI options</li> <li>● Optional Plugins: <ul style="list-style-type: none"> <li>◇ Allow the user to rate the recommendations</li> <li>◇ Allow users to provide feedback to other users</li> </ul> </li> </ul> <p><b>Other Components</b></p> <ul style="list-style-type: none"> <li>● Database</li> <li>● Security and Privacy Layer</li> <li>● User Cloud</li> </ul>
--

Figure 2: Task list for weHelp modules

own *task list*, which is a required set of tasks they must perform, and their own optional set of *plugins* for providing added functionality, shown in Figure 2. These plugins are conceptual, and in a given implementation might be hard-wired into the design/code or indeed be supplied in a replaceable plugin form. The following subsections describe the various components of the weHelp reference architecture. We use Amazon [Ama] as a running example to present how its recommender system behavior maps to weHelp<sup>1</sup>. We discuss how the architectures of our three systems map closely to weHelp.

### 3.1 Watcher

The Watcher module is an observer of user activities. It “watches” what a user does with the tools or software in question and logs this information. These observations may be explicit or implicit as far as the user is concerned. Explicit observations enable the user to annotate or tag the data and/or her activities. Other forms of explicit feedback include asking the user to fill out a form or answer a survey. Implicit observations, on the other hand, work in the background and monitor what the user is doing. Examples of implicit observations include counting the number of times a particular item was bought, or the number of times a link was clicked. Implicit observation is usually a better option, where possible, as it does not require any extra effort by the user and is also unobtrusive. The Watcher needs to generate an easily parseable representation of the user activities. Periodically, a log of users’ activities needs to be stored in a database or web repository. Optional plugin functionality provided by the Watcher may include allowing the user to provide extra domain specific information and allowing users to rate certain activities.

In the case of Amazon, we can imagine that the Watcher module keeps track of user activities by observing the web pages visited. Consider an example where a user wishes to buy an MP3 player. The Watcher module can, implicitly, keep track of information such as the different kinds of MP3 players looked at. If the user decides to buy a particular MP3 player, the Watcher can also keep track of details such as the color, the capacity, and the vendor. The user can also provide explicit feedback to the Watcher module by rating the product he bought and writing a review for it. Amazon’s Watcher module would conceivably log this information and it will later be used by the Learner module to infer recommendations for other users.

The following paragraphs describe how our three systems implement the Watcher module.

The Watcher module in genSpace transparently monitors the users’ activities as they use geWorkbench. Whenever any data analysis is executed, different kinds of information are captured such as the name of the analysis, the type of the data set, and the current time. The Watcher module also allows users to choose how their data is logged: with their user names, anonymously, or not at all. Finally, in genSpace, all users are treated equally; there is no differentiation between “novice users” and “expert users”. Thus, all the user logs are weighted equally when being considered as source data for the recommendations. This

---

<sup>1</sup>The authors have no affiliation with Amazon except as customers. The Amazon architecture has been inferred from the functionality provided by the website and may bear little or no relationship to the true architecture.

allows genSpace to work on the principle that if a particular tool is used in a certain way by a majority of the users, that is likely the “right” way to use that tool.

Similar to genSpace, the Watcher module in Retina transparently records students’ compilation attempts and compiler errors. Additionally, any compilation errors are reported to the student as normal, but for each error, the type of error, the file name and line number, and the associated error message are all recorded as well. This information is accumulated completely transparently from the students’ perspective without any manual intervention. Retina allows users to choose how their data is logged: with their user names, anonymously, or not at all. Finally, similar to genSpace, Retina treats all users equally and does not differentiate between “expert users” and “novices”.

On the other hand, COMPASS will distinguish between “novice users” and “expert users”. The Watcher module of COMPASS is meant specifically for experienced programmers, also known as “gurus”. COMPASS assumes that a guru is an expert at writing optimizations for the corresponding input program. In contrast to Retina and genSpace, recording observations is not transparent to the user in COMPASS. The COMPASS IDE will allow the user to select portions of the input program (hot-spots) for parallelization. The expert user will then write a corresponding parallel program in the IDE. The Watcher module parses and stores the input program and the corresponding parallel solution. The Watcher module will have an additional plugin that allows users to give additional domain information such as target architecture and miscellaneous comments.

### 3.2 Learner

The Learner module is responsible for inferring patterns and/or recommendations. It will use the data that the Watcher module stores in the database. Depending on the problem domain, the Learner module can either use a simple rule-based system, data aggregation, or complex data mining algorithms. Optional plugins to the Learner module include the ability to have different ranking algorithms. The Learner module may also weigh the user data in many different ways such as weighing data from experts more than the data from novice users, or weighing recent data more than older data. Finally, the Learner module may also generate overall statistics about system usage and include optimizations such as caching the results and data to improve the response time to user queries. Note that the name “Learner” is intended to refer to *any* means for distilling knowledge from the data accumulated by the Watcher, not just those using machine learning algorithms.

Continuing our Amazon example, we can imagine that the Learner module uses the user information recorded by the Watcher module to infer patterns and recommendations. An example of these recommendations is suggesting alternate products to buy after considering other similar products viewed by the user. The Learner module also generates statistics such as the highest rated products and the most popular products in different categories.

The following paragraphs describe how our three systems implement the Learner module.

In genSpace, the Learner module uses data aggregation to provide the recommendations. For example, one of the suggestions provided by genSpace is ‘what is the next tool to use’.

The Learner module uses the accumulated user data to find the most common workflows that are supersets of the user's current workflow. Looking at the most common workflows, it can suggest the best tool to use next. The genSpace Learner module uses an exponential time-decay formula [CS03] to weigh recent user data more heavily. This allows us to address the problem of concept drift [WK96], *i.e.*, workflows performed by users a long time ago may not be relevant today. The Learner module in genSpace also generates overall system statistics such as the most popular tools and most popular workflows.

In the case of Retina, although many of the recommendations are rule-based, the Learner module does perform some analysis of the data to determine the suggestions to make to individual students. For instance, one of Retina's suggestions is the amount of time that the student can expect to spend on the next programming assignment. This is done by considering the student's past performance on previous assignments with respect to the class average of time spent, and then finding the time that it took similarly-ranked students to complete the assignment in previous semesters. Another suggestion made by Retina involves the types of compiler errors that it feels the student is likely to make. This is achieved by noting any errors that the student has frequently made on previous assignments, especially those that fall outside the list of most common errors across all students in the class. The Learner module in Retina also has plugins for generating statistics that can be used in reports for the course instructor. The instructor can get an understanding of an individual student's efforts on a particular assignment, by seeing a list of all of the student's compilation and runtime errors, as well as aggregate data about the total number of compilation errors, the most common compilation error, and an approximation of how much time was spent on the assignment. For example, the instructor can select a single assignment and see an overview of how the class has performed as a whole, *e.g.*, the most common compilation and runtime errors. The instructor can also get a report of how much time each student has spent on the assignment, and the average time spent for all students in the class. This lets the instructor gauge the difficulty of a particular assignment.

In COMPASS, the Learner module will search for parallelizations performed by gurus so as to help inexperienced users. The workflow of the Learner module is as follows: the sequential code input by the user is first instrumented and its coverage data is analyzed to extract hot-spots (frequently executed parts of code). These hot-spots will now act as the input query to the database. The Learner module will use a code matching and ranking algorithm to extract various parallelizations for the user. The Learner module will also ask the user for some domain information, which it uses to get the best possible match.

### 3.3 Advisor

The Advisor module is responsible for providing recommendations to the users. These recommendations are given using the data inferred by the Learner module. The Advisor module can be implemented using a variety of different user interface options, depending on how exactly these recommendations should be provided, based on the problem domain. Furthermore, the recommendations can either be pushed to the user or pulled by the user. Pulled recommendations are very beneficial as users can ask for particular recommenda-

tions when they need them the most. Pushed recommendations can also be useful as they can be dynamic and take into account the user's current activities. Care needs to be taken, however, to avoid the "Clippy Effect" [Gal06], which would only serve to annoy the user. Optional plugins include allowing the user to rate the suggestions. Users may also provide feedback, *e.g.*, through comments, to the other users as well as to the recommender system. This form of feedback can be extremely useful as some recommendations might work better than others.

In the example of Amazon, the Advisor provides the recommendations to the user using HTML web pages as a user interface. *E.g.*, when a user is viewing a product, the Advisor module provides information such as other similar products viewed by users who also viewed this product, or other products purchased by users who also purchased this product.

The following paragraphs describe how our three systems implement the Advisor module.

In genSpace, the Advisor module is implemented as another component for geWorkbench. The user interface is Java Swing. The Advisor provides both pushed and pulled recommendations. Users can ask for recommendations such as finding workflows that include a certain tool or workflows that begin with a certain tool. Users can also view overall system usage statistics such as the top three most popular workflows, and the most popular tools. Users can also get pushed recommendations using "Real-Time Workflow Suggestions". As a user interacts with geWorkbench and runs different analyses, genSpace can provide real-time suggestions. Examples of the feedback provided include suggesting the most popular tool to use next and common superflows including the user's current workflow. Users can also rate the tools, write comments, and read other users' comments.

Similar to genSpace, the Advisor module in Retina supports two different models for providing advice and suggestions. Students can request suggestions by accessing a web application and getting reports about their own behavior, and what to expect from upcoming programming assignments. These suggestions include the amount of time that the student will likely spend on the assignment, and different errors to look out for. Additionally, Retina can produce immediate, real-time recommendations that are proactively sent to students based on their observed programming activities. These recommendations are sent to the students as they are programming and as their event logs are being captured. Retina uses Instant Messaging (IM) applications as the user interface for its recommendations.

The Advisor module in COMPASS will aid the user by providing suggestions in stylized templates known as 'sketches'. These sketches will be shown in the form of graphical overlays on the existing source code, which will be easily understood by the user. Users can accept the suggestions as is, or alter them according to their requirements. COMPASS will also allow users to give feedback to the system regarding the usefulness of the suggestion.

### 3.4 Other Components

The **Database** component is used as a data store for the user activities.

Depending on the problem domain, security and privacy issues can be critical as the rec-

ommendations provided by the system may be used to identify individual users or invade the privacy of users by inferring how a particular user uses the system. The **Security and Privacy Layer** can be used to provide the necessary functionality to ensure that the users' privacy is maintained and that security threats are mitigated.

Of course, a recommendation system is not useful without users in the so-called “user cloud”. Broadly speaking, users can be separated into two types: those from whom the recommendation system learns, and those who benefit from the aggregated knowledge. Sometimes those user groups can overlap. For instance, in genSpace all users are considered the same: anyone can contribute to the store of knowledge, and anyone can benefit from it. On the other hand, sometimes these user roles can be explicit. For instance, in COMPASS, users can either be “gurus” (those whose parallelization activities have been monitored) or “novices” (those who are trying to parallelize code and need assistance).

## 4 Related Work

There have been many examples of reference architectures in varying domains such as scientific workflow management [LLF<sup>+</sup>09], and web servers [HH00].

Previous research into community-driven recommender systems has investigated the rationale behind such tools, *e.g.*, the psychology of collaborative technical help and help-giving [COT<sup>+</sup>06, TR04] and using organizational memory for collaborative help [AM96]. The KnoSoS project [CKVDM06] applies social networking concepts to knowledge sharing by investigating how to create group boundaries and track content. Carroll *et al.* have looked at extending the idea of knowledge sharing to concept sharing, and eventually activity sharing and activity awareness [CRCG06]. We build on these important works by considering the challenges of designing a reference architecture that supports such techniques, and hope that our work facilitates the building of such systems in the future.

Fink and Kobsa in their work on “User Modeling Systems” [FK00] discuss some recommender systems but only focus on a generalized case study of these systems and compare each of their architectures. There has been other work [GNOT92, RIS<sup>+</sup>94] in which recommender system architectures are discussed. Most of the research has focused only on the existing architectures and implementations without aiming to propose a generic reference architecture. To the best of our knowledge, there is no definitive work that describes a generic end-to-end architecture for such systems.

Our reference architecture can be viewed as a design pattern. The weHelp reference architecture does not map to any of the well-known design patterns, to the best of our knowledge. The design pattern that is the closest to weHelp is the MVC design pattern [Ree79]. The Model in MVC corresponds to the Database and the Learner components in weHelp. The Controller in MVC is similar to the Learner in weHelp. The View in MVC corresponds to the Watcher and the Advisor components in weHelp. We did not use the MVC design pattern as is, as the functionality and the division of responsibility need to be different for social recommender systems.

## 5 Limitations and Future Work

Knowledge-based recommendation systems use domain knowledge to gather inferences about the requirements of the user and to understand how a particular item meets a user's need. These systems are very similar to expert systems and may not utilize any aspects of social networking. Examples of such systems include ACE [Mei88] or [MFS09]. A limitation of the weHelp architecture is that systems that do not use any knowledge gained from usage data cannot be mapped to the weHelp reference architecture.

Some of the future challenges would involve expanding on and providing more concrete modules for the Security and Privacy layer.

## 6 Conclusion

We have described a reference architecture for social recommender systems. Our reference architecture, *weHelp*, is designed to be generic and domain agnostic. The component structure of our reference architecture will provide a template for designing modularized recommender systems and could lead to standard interfaces and interoperability among recommender system components.

## 7 Acknowledgements

The authors would like to thank Aris Floratos and Kiran Keshav for their guidance and assistance with genSpace. We would also like to thank Simha Sethumadhavan for his direction on the COMPASS project, and Sahar Hasan for her work on Retina. We also thank Lauren Wilcox for sharing her insights into recommendation systems. The authors are members of the Programming Systems Lab, funded in part by NSF CNS-0905246, CNS-0717544, CNS-0627473 and CNS-0426623, and NIH 1 U54 CA121852-01A1.

## References

- [AM96] M. Ackerman and D. McDonald. Answer Garden 2: merging organizational memory with collaborative help. In *Proc. of the 1996 ACM conference on computer supported cooperative work (CSCW)*, pages 97–105, 1996.
- [Ama] Amazon.com. <http://www.amazon.com>.
- [BEKR07] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *RecSys '07: Proc. of the 2007 ACM conference on Recommender systems*, pages 9–16, 2007.
- [CFKW] A. Califano, A. Floratos, M. Kustagi, and J. Watkinson. geWorkbench: An Open-Source Platform for Integrated Genomics. <http://www.geworkbench.org>.

- [CKVDM06] T. Coenen, D. Kenis, C. Van Damme, and E. Matthys. Knowledge Sharing over Social Networking Systems: Architecture, Usage Patterns and Their Application. *LNCS 4277: On the Move to Meaningful Internet Systems 2006*, pages 189–198, 2006.
- [COT<sup>+</sup>06] A. Crabtree, J. O’Neill, P. Tolmie, S. Castellani, T. Colombino, and A. Grasso. The practical indispensability of articulation work to immediate and remote help-giving. In *Proc. of the 20th anniversary conference on computer supported cooperative work (CSCW)*, pages 219–228, 2006.
- [CRCG06] J. M. Carroll, M. B. Rosson, G. Convertino, and C. H. Ganoë. Awareness and teamwork in computer-supported collaborations. *Interacting with Computers*, 18(1):21–46, January 2006.
- [CS03] E. Cohen and M. Strauss. Maintaining time-decaying stream aggregates. In *Proc. of the 22nd ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems (PODS)*, pages 223–233, 2003.
- [Fac] Facebook. <http://www.facebook.com>.
- [FK00] J. Fink and A. Kobsa. A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. *User Modeling and User-Adapted Interaction*, 10(2-3):209–249, 2000.
- [Gal06] R. G. P. Galluccio. Humanizing CALL: The use of pedagogical agents as language tutors. New England Regional Association for Language Learning Technology, Oct. 2006.
- [GDM<sup>+</sup>08] W. Geyer, C. Dugan, D. R. Millen, M. Muller, and J. Freyne. Recommending topics for self-descriptions in online user profiles. In *RecSys ’08: Proc. of the 2008 ACM conference on Recommender systems*, pages 59–66, 2008.
- [GM08] A. Gunawardana and C. Meek. Tied boltzmann machines for cold start recommendations. In *RecSys ’08: Proc. of the 2008 ACM conference on Recommender systems*, pages 19–26, 2008.
- [GNOT92] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.
- [HH00] A.E. Hassan and R.C. Holt. A reference architecture for Web servers. *Proc. of Seventh Working Conference on Reverse Engineering*, pages 150–159, 2000.
- [LLF<sup>+</sup>09] C. Lin, S. Lu, X. Fei, A. Chebotko, D. Pai, Z. Lai, F. Fotouhi, and J. Hua. A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution. *IEEE Transaction on Services Computing*, 2(1):79–92, 2009.
- [MA07] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys ’07: Proc. of the 2007 ACM conference on Recommender systems*, pages 17–24, 2007.
- [Mei88] B. J. Meier. ACE: a color expert system for user interface design. In *UIST ’88: Proc. of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, pages 117–128, 1988.
- [MFS09] M. Mandl, A. Felfernig, and M. Schubert. Consumer Decision Making in Knowledge-Based Recommendation. *IEEE International Conference on Active Media Technology and Brain Informatics, Peking, China, Springer Lecture Notes on Computer Science, LNCS 5820*, pages 69–80, 2009.



- [MKLH09] C. Murphy, G. Kaiser, K. Loveland, and S. Hasan. Retina: Helping Students and Instructors Based on Observed Programming Activities. In *Proc. of the 40th ACM SIGCSE Technical Symposium on Computer Science Education*, pages 178–182, March 2009.
- [MSKW08] C. Murphy, S. Sheth, G. Kaiser, and L. Wilcox. genSpace: Exploring Social Networking Metaphors for Knowledge Sharing and Scientific Collaborative Work. In *1st Intl. Workshop on Social Software Engg. and Applications*, pages 29–36, September 2008.
- [Neta] Netflix. <http://www.netflix.com>.
- [Netb] Netflix Prize. <http://www.netflixprize.com>.
- [Pan] Pandora Radio. <http://www.pandora.com>.
- [PT08] Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. In *RecSys '08: Proc. of the 2008 ACM Conference on Recommender systems*, pages 11–18, 2008.
- [Ree79] T. M. H. Reenskaug. Models-Views-Controllers. *Xerox PARC technical note*, 1979.
- [RIS<sup>+</sup>94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proc. of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.
- [SAG<sup>+</sup>09] S. Sethumadhavan, N. Arora, R. B. Ganpathi, J. Demme, and G. Kaiser. COMPASS: Community Driven Parallelization Advisor for Sequential Software. In *2nd Intl. Workshop on Multicore Software Engg.*, 2009.
- [TR04] M. Twidale and K. Ruhleder. Where am I and who am I?: issues in collaborative technical help. In *Proc. of the 2004 ACM conference on computer supported cooperative work (CSCW)*, pages 378–387, 2004.
- [WK96] G. Widmer and M. Kubat. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23(1):69–101, 1996.
- [ZC08] V. Zanardi and L. Capra. Social ranking: uncovering relevant content using tag-based recommender systems. In *RecSys '08: Proc. of the 2008 ACM conference on Recommender systems*, pages 51–58, 2008.
- [ZP07] J. Zhang and P. Pu. A recursive prediction algorithm for collaborative filtering recommender systems. In *RecSys '07: Proc. of the 2007 ACM conference on Recommender systems*, pages 57–64, 2007.

# Switch! Recommending Artifacts Needed Next Based on Personal and Shared Context

Alexander Sahm and Walid Maalej

Technische Universität München  
Bolzmannstraße 3  
85748 Garching  
sahm@cs.tum.edu  
maalejw@cs.tum.edu

**Abstract:** While performing a certain task software developers use multiple tools, read different artifacts and change others. As software developers are often interrupted during a task, they end up simultaneously using a vast set of tools and artifacts. They need to switch between those artifacts many times until a task is completed. In sum a lot of time gets wasted due to locating, reopening or selecting the right artifact needed next. To address this problem we introduce *Switch!*, a context aware artifact recommendation and switching tool for software developers. *Switch!* recommends artifacts that are likely needed in the current situation, based on task semantics, interaction history and community profile.

## 1 Introduction

Software engineers work on several tasks in parallel [KM06], and their work is often interrupted [GM04]. Several studies have shown the fragmented nature of software engineering work. In a survey with 800 software engineers [Ma09] we asked subjects how frequent do they switch their work focus because of, e.g. a high priority request or remembering something. We found that about 60% of respondents change the focus at least hourly. Other observational studies found that engineers switch their focus even more frequently: every five minutes on average [MGH05, KDV07]. Mark, Gonzalez and Harris observed 57% of tasks were interrupted. As a result work was often fragmented into many small work sessions [MGH05] and engineers have to deal with many task switches.

In order to perform a certain task, engineers usually use various tools, in average 5 tools for a single task [Ma09] (such as source code editors, bug trackers or version control systems). Moreover engineers read and change different artifacts, like source code files, bug reports or diagrams. A case study done by Zou [ZG06] found that 8 code files are read and 6 are changed during a task. As engineers typically spend only 50% of their time for code creation [MH09], and as Zou only observed code files inside IDEs, it is most likely that the overall number of used artifacts is even higher. Multiple tasks in

conjunction with multiple artifacts per task increase the amount of artifacts the engineer has to deal with in parallel.

In this paper we introduce *Switch!*, a recommendation tool that supports software engineers to switch artifacts by employing task semantics, engineer’s interaction history and shared community profiles. The contribution of the paper is twofold. First, it introduces a conceptual model for a context-aware artifact switching to support software engineering (SE) work. Second, it presents a framework and user interface concept, which implements this model. In section 2 we describe the problems *Switch!* addresses. In section 3 we present our solution concept and the metrics required to recommend artifacts depending on engineer’s context. Section 4 introduces our tool and gives an overview of its user interface concept and architecture. In section 5 we present a paper-prototype based evaluation. Finally, section 6 discusses related work and section 7 summarizes the paper and gives an outlook on future directions.

## 2 Switching Problems

Modern desktop environments use windows as a main abstraction for artifact switching. Figure 1 shows the different implementations provided by *Microsoft Windows* and *Mac OS X*. We identified four drawbacks of this approach.



Figure 1: Window switching on Mac OS X (top) and Microsoft Windows (bottom)

1. *Mixture of task-specific windows*: Different tasks involve different artifacts and tools to use. While e.g. the debugger window is used for the bug fixing task, the text editor is used for the documentation refinement task. Window management systems are not aware of engineers' tasks and usually show windows of all tasks the engineer is working on. When switching artifacts, task irrelevant windows create a screen clutter.
2. *Window-centric instead of artifact-centric*: Desktop environments provide mechanisms to switch between open windows or running applications. However many applications display several artifacts in a single window. A web browser window, e.g., might contain several tabs. Thus engineers are not able to select one of these artifacts directly. Instead, first the window containing the artifact has to be identified, then the artifact inside the window.
3. *Task history is lost*: Information on artifacts and tools used during past tasks is not considered. If an engineer does not manually track the used artifacts in the current task, this information will be lost. When resuming a postponed task or working on a similar task in future the engineer will need to search and locate the same set of artifacts again.
4. *Open vs. closed windows*: Window management systems only consider open windows and running applications when switching artifacts. If an artifact is currently represented by a window, it is open. The engineer in this case can switch to it. If an artifact is not represented by a window, it is closed, and can not be a switch target. Engineers first need to find the artifact and open it manually. This interrupts the work-flow and can be time consuming.

### 3 Context-aware Artifact Recommendation

Our context model for a context-aware artifact recommendation consists of three dimensions: the semantics of the task being performed, the personal interaction history of the engineer and a shared community profile of co-workers. These dimensions respectively represent the “universal”, the “personal” and the “social” perspective on SE work. We first describe the two dimensions and then introduce the recommendation algorithm combining them.

#### 3.1 Task Semantics

SE tasks can be grouped based on the similarity of their goals and the software engineering activity they represent. Bug fixing tasks have a different goal than code-review or requirement analysis tasks. We distinguish between the following types of SE tasks (from [KDV07] and [MH09]): writing code, fixing bugs, reasoning about design, testing a program, maintaining requirements, integrating components and maintaining awareness. When working on one of these tasks (e.g. fixing a bug), engineers follow specific workflows (e.g. read report, reproduce bug, debug, commit), by using specific tools (e.g.

bug tracker, debugger) and specific types of artifacts (e.g. bug reports, source code). We annotate artifacts used during SE work with their semantic types. For example, a website rendering a bug report is of type *bug report*, a text file describing a test case is of type *test case*. The complete ontology describing the artifact and task types is available for download on the teamweaver site<sup>1</sup>. The ontology associates different artifact types to different task types. For example the engineer is likely to use a test case during testing, but not during implementation. The ontology enables to determine a fuzzy set of artifact candidates to be recommended, based on the type of current task.

### 3.2 Engineer's Interaction History

Even when working on similar tasks, software engineers differ in their work habits, work environments, work-flows, programming languages and tools. A need for a specific artifact highly depends on any of those factors. In order to address these needs we use the interaction history of the engineer. Analyzing engineer's interaction history gives insight on personal preferences. For example an engineer might use a specific folder for storing stable system builds. By analyzing her interaction history this preference can be detected and the folder can be recommended. We define three metrics to describe the engineers interaction with a single artifact.

- Usage Duration (*du*): The time in seconds the artifact was used by the engineer. Using includes reading as well as editing of the artifact.
- Usage Frequency (*fr*): The count how often an artifact has been used by the engineer.
- Switched To (*st*): The number of times this artifact was switched to from an other particular artifact. This value is relative to the current active artifact. The number is incremented if the user switches from the current active artifact to the given artifact.

These metrics are calculated for the artifact instances and the artifact types. While *du* and *fr* apply to a single artifact, the *st* value applies to two artifacts. The value of *st* describes the direct relationship between two artifacts (resp. artifact types). For example, a software engineer uses three different artifact types A, B and C. The engineer switches most of the time between A and C, and between C and B, but never directly between A and B. The *st* value is high for C when A or B is the active artifact type, or high for A and B if C is the active artifact type.

### 3.3 Shared Community Profile

Software engineers work in teams and share common characteristics with other engineers, i.e. members of particular communities. We argue that the context of co-workers

---

<sup>1</sup> <http://www.teamweaver.org/wiki/index.php/ontology>

is also of interest for recommendations. To reason about the context of co-workers their interaction history information is needed. To provide interaction history data, every developer of a project team or organization can share parts of their own interaction history. Before being shared, the interaction history is condensed to a profile, containing only abstract usage information. The data at this point cannot be traced to single tasks or interactions. This community profile contains the same numeric criteria as the personal interaction history described above. The client, depending on the profile provider's team role or experience level, aggregates the collected profiles.

By analyzing shared community profile team specific knowledge and best practices can be used for recommendation. The project or organization dependent work environment can also be detected. For example when the community profiles of the teammates are shared, the recommendation for a developer new to the project will also contain the project specific bug tracking system.

### 3.4 Recommendation Algorithm

The recommendation process is divided into two steps. First the number of possible artifacts to recommend is reduced by using the task semantic and focusing on a subset of the artifact types. In the second step artifacts of this types are ranked based on engineers interaction history. This reduces the number of rankings to be calculated and improves the performance of recommendation creation.

**I. Artifact Type Suitability** In order to identify the artifact type subset to use, we calculate a value representing the importance of each type. Only the artifact types with an importance rating higher than a given threshold are used. The used criteria refer to the interaction data:  $du$ ,  $fr$  and  $st$ . The values are all normalized to be between 0 and 1, with 0 being unimportant and 1 being very important. Since the  $st$  is the only value directly linked to the current active artifact type, it is considered to be a higher relevance indicator and hence has a factor 2. The suitability of an artifact type  $t$  is calculated as follows:

$$Suitability(t) = du(t) + fr(t) + st(t) * 2$$

**II. Artifact Ranking** All artifacts of suitable types are ranked according to their relevance. In contrast to the first step, here we calculate the relevance of concrete artifacts. The same interaction data is available for the artifacts themselves. Interaction data is available for the following time slots:

- Current Session ( $cSe$ ): The artifact interaction data collected during the current session (sessions result from postponing and resuming a task).
- Current Task ( $cTa$ ): The artifact interaction data during the whole task (i.e. several sessions belonging to the same task).
- Similar Tasks ( $sTa$ ): Artifact interaction data during tasks of the same type.

- External Tasks (*eTa*): Summarized artifact interaction data from co-workers during tasks of the same type than the current task.

The *sTa* adds the task semantics to the ranking of the artifacts while *eTa* represents the social aspect. The ranking overall is the combination of the universal task semantics, the personal interaction history and the social community profiles. We assume that recent interaction data is more important than old interaction data and personal interaction data is more important than universal or shared interaction data. We therefore assign the importance factors 2 for *cSe*, 1 for *cTa*, 0.5 for *sTa* and 0.25 for *eTa*. The artifact ranking can be calculated as follows:

$$du(a) = 0.25 * du(eTa, a) + 0.5 * du(sTa, a) + du(cTa, a) + 2 * du(cSe, a)$$

$$fr(a) = 0.25 * fr(eTa, a) + 0.5 * fr(sTa, a) + fr(cTa, a) + 2 * fr(cSe, a)$$

$$st(a) = 0.25 * st(eTa, a) + 0.5 * st(sTa, a) + st(cTa, a) + 2 * st(cSe, a)$$

$$Rank(a) = du(a) + fr(a) + st(a) * 2$$

## 4 SWITCH!

We introduce *Switch!*, a tool that uses the described context model in order to recommend artifacts needed next. We present a short scenario on how *Switch!* is used. Then we describe its user interface and architecture.

### 4.1 Usage Scenario

Alice is implementing a new feature to a existing subsystem. She uses *Switch!* which monitors her interactions and the used artifacts and tools in the background. While trying to understand how the subsystem she is extending is designed, she presses the *Switch!* keyboard shortcut. *Switch!* evaluates her interactions and creates a recommendation of artifacts that Alice want to use next. The recommendation contains PDF files of UML diagrams, the email communication about this subsystem with a co-worker, as well as some web pages Alice looked at providing additional framework information. Alice can navigate through the offered artifacts by mouse or keyboard. Alice selects the artifact she wants to switch to. If the respective document is already open it will be brought to front, if it was not, it will be opened.

### 4.2 User Interface Concept

The user interface of *Switch!* is used by the engineers very often but only for short periods of time. Thus the user interface needs to be designed to foster fast recognition of the offered artifacts. We present the recommendations using a visual graphical interface instead of textual ones. This gives the engineer the ability to select the needed artifact quickly on a visual basis. Figure 2 shows the concept of *Switch!* user interface. We de-

signed and implemented the user interface for the *Mac OS X* operating system. Though, its concepts can be implemented on different platforms as well.

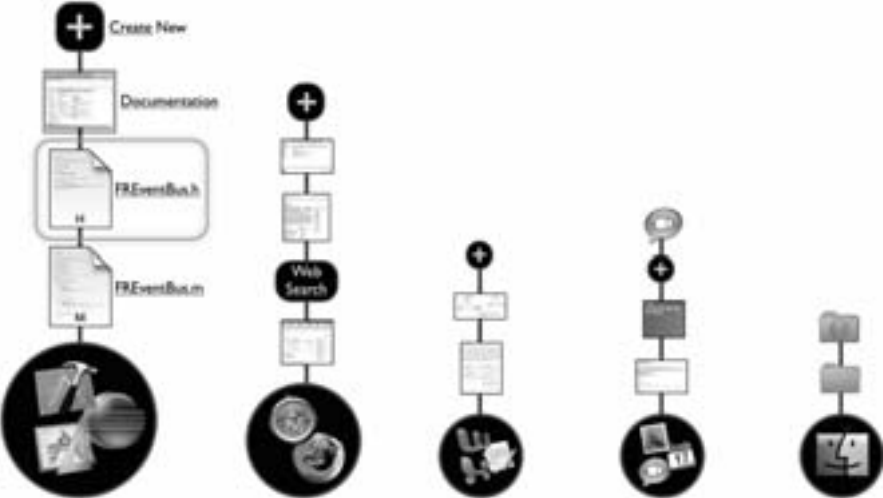


Figure 2: Switch! user interface showing a recommendation

Artifacts with similar types and usually manipulated with similar applications, are grouped together. A group is visually identified by composing icons from its most important applications. Every group contains a maximum of seven artifacts ordered by importance from bottom to top. In order to take advantage of the user’s spatial memory, these groups are positioned in a same order according to their importance for the SE work from left to right.

For a quick overview of recommended artifacts each artifact is displayed using a preview image of its content. Every artifact of currently selected group has a bigger size and includes a title. Besides special functions enable the engineer to fulfill common tasks directly from Switch!, like creating a new class or searching in the address book.

The user can move a selection along the offered elements either by hovering over the elements with the mouse or by using the keyboard. The selection can thereby be moved between the groups or between the recommended group artifacts. The artifact being active before starting *Switch!* is not displayed as switching the same artifact is senseless. The engineer then either clicks the needed artifact or presses the enter key while it is selected. The *Switch!* window is hidden and the artifact is shown in front of other windows.

**4.3 Framework and Architecture**

*Switch!* is build on top of the knowledge sharing framework *TeamWeaver* [MH08]. The two main layers of *TeamWeaver* we are using are the *Context System* and the *Distributed Knowledge Model* layer. The *Context System* layer monitors and interprets the engineers’



behavior while the *Distributed Knowledge Model* layer stores and exchanges the behavioral model. An architecture overview is shown in Figure 3. We shortly describe the main components and how they are used inside *Switch!*. The *TeamWeaver Ontologies* describe the semantic model of tasks and artifacts and the relationships to each other. The ontologies incorporate the task semantics. The *Context Monitor* contains various sensors to existing tools and information sources - such as email clients, web browsers and development environments. These sensors monitor the raw interaction events triggered by the engineer and their semantic representations. This component builds the personal interaction history. The *Context Interpreter* then classifies the current task based on the interaction history data monitored by the *Context Monitor*. The *Local Metadata Store* manages metadata about the artifacts and the tasks of the engineer. In order to use the social aspects, in particular the sharing of community profiles the *P2P Infrastructure* component is used. It enables the sharing of own interaction history data, as well as the receiving of community profiles shared by co-workers. The *Profiler* component creates the own profile to share and aggregates received ones. *Switch!* uses the collected and analyzed interaction data provided by the *Context System* and integrates it with the universal assumptions and the metadata provided by the *Distributed Knowledge Model*. The result is used to create the recommendation by selecting and ranking suitable artifacts.

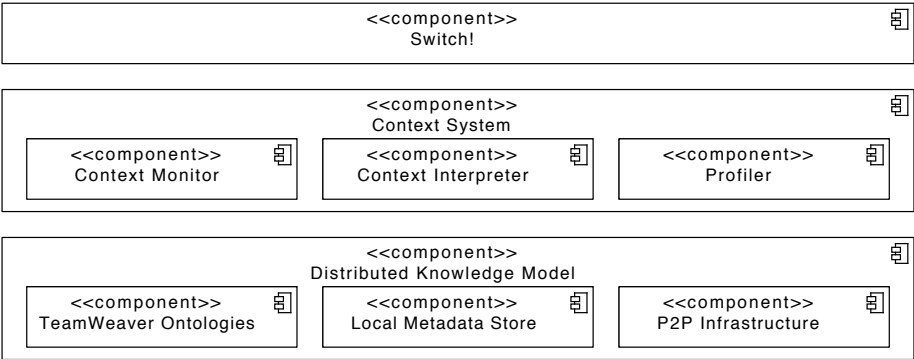


Figure 3: Switch! architecture overview

## 5 Evaluation

We present the results of a paper prototype study conducted to evaluate the *Switch!* concepts and user interface. Then we discuss the prototypical implementation.

### 5.1 Paper Prototype Study

To evaluate our recommendation approach, we conducted an experiment with experienced software engineers by using a paper-based prototype. This prototype was created based on the user interface concepts shown in Figure 2.

The elements of the user interface are separately printed on paper to simulate needed combinations. If a participant selects an artifact the appropriate screenshot is presented to simulate the desktop environment. The prototype is presented by the experiment supervisor on a table, with the participant sitting in front. The supervisor simulates the recommendations depending on the participant input and the algorithm introduced above. Each experiment session is recorded by a video camera to track the physical actions and participants' verbal reactions. Participants should use the paper prototype, as they would work in their everyday environment. They should "think loudly" [Sea99] and express in words what they are doing and why. Each participant has to perform one implementation, one bug fixing and one documentation tasks. The concrete task instructions can be downloaded from the teamweaver site<sup>2</sup>.

Since *Switch!* is developed for *Mac OS X* we focused on engineers primarily working on this platform. We recruited seven engineers from a Munich-based company specialized on *Mac OS X* development and from the TUM Mac development Lab. While *XCode* is used as main IDE by 2 participants, 5 participants mostly use *Eclipse*. In average each simulation session lasts 40-60 minutes including the procedure explanation and discussion.

Six participants claimed that with current environments, is difficult to keep an overview of the needed artifacts and distinguish them from the ones not needed next. Our solution presented to them was rated positively. Five stated they would use a tool with the presented functionality. We observed that all seven participants were not sure what to do and how the artifacts are arranged, when introduced to the *Switch!* user interface. But as they started pointing with the mouse or pressing the cursor keys, the concept of interaction was understood by all of them. It took participants in avg. 4-5 min. to get familiar with the interface. Participants identified the instant magnification and tool reaction to be very useful in such situation. Five engineers recognized the artifacts they should select very quickly by their icon (< 2 sec.). The other two needed to take a closer look at the titles or asked the supervisor for help. Participants did not understand the recommendation criteria quickly. Four participants asked the supervisor several times why the particular artifacts are recommended. Three engineers asked what to do if the needed artifact is not included. Two engineers stated artifact switching using *Switch!* would be too complicated for daily use. These two engineers exclusively use *Eclipse* during the work.

Overall the purpose of *Switch!* and its features were quickly understood. But the reason why a particular artifact is recommended is not sufficiently understood by the engineers. Additional graphical presentation of most relevant relationships and recommendation criteria might considered. Providing a detailed rationale for a recommendation would increase confidence in the recommendation, but increases the information overload as well [RWZ10]. Furthermore engineers wanted to be able to select artifacts not recommended by *Switch!* if necessary. During a discussion at the session end, participant mentioned a) the accuracy of the recommendation, b) performance and c) the possibility to customize artifact groups as additional requirements.

---

<sup>2</sup> <http://www.teamweaver.org/wiki/index.php/switchExperiment>

## 5.2 Prototypical Implementation

The current version of *Switch!* uses the described algorithm to create, visualize and recommend the following artifact types: source code, PDF files, *XCode* documentation, emails, web pages and plain text files. When hitting a global keyboard shortcut the main switching window is shown. It includes (both open and closed) artifacts identified by *Switch!* to be relevant. The artifacts are categorized into the tool groups and sorted inside the groups according to calculated ranks. The preview images are created using the *Quicklook* service of *Mac OS X*. *Quicklook* by default has the ability to create preview images from many different file formats. In this version *Switch!* supports the task types bug fixing, testing and documenting. For well-defined tasks with a limited set of possible artifacts, the current version provides suitable recommendations. We are using *Switch!* when dealing with several bug fixing and testing tasks at once.

## 6 Related Work

Several related systems have been proposed to address engineers information needs as well as the encountered information overload. These can roughly be separated into three categories: Application Launchers, Recommender Systems for software development and Window Switchers.

Application launchers like *Quicksilver* [qui] and *LaunchBar* [lau] for the *Mac OS X* platform or *Launchy* [Kar] for *Windows*, enable the user to quickly start applications, open files or execute predefined commands. These tools usually work by pressing a shortcut and typing short queries containing a few letters. The results of possible documents to open or tools to launch are immediately listed. Frequently used commands are rated higher for the next similar query (equal prefixes). Application launchers allow users to efficiently execute common tasks (e.g. open document, send email). However, these tools do not consider users' context. Users still need to know exactly the artifact they want to switch to, its name and its path.

Recommender Systems for software engineering offer a more advanced functionality. Their goal is to address the current information needs of the engineer by recommending suitable artifacts [RWZ10]. A landscape of recommendation systems used in software development is described in [HM08]. *Dhruv* [ASH06], for example, is a tool that presents information on a bug report taking into account the information given by the bug report itself. *Mylyn* [KM06], hides not needed classes and methods from the Eclipse IDE, by considering previously selected and edited ones. Other recommendation systems such as *Malibu* [Sh08] and *TaskTracer* [DDJ05] are not tightened to software development. They recommend windows depending on the current user activity. Thereby, the activity needs to be recorded manually. *TaskPredictor2* [Sh09] applies machine learning mechanisms to detect unrecorded activities.

Operating systems often use open windows as basis for switching artifacts, assuming that windows represent artifacts. They are designed to offer the open windows in a way the user is able to find the needed ones more easily. The *GroupBar* [Sm03] assigns sev-

eral windows to one task. *SWISH* [OI06] groups similar windows by using the semantics of the titles and interaction history. A similar approach is implemented by *Taskposé* [BSW08] which groups miniature images of the open windows according to their relatedness. Artifacts not represented by a single window - such as a method of a class - cannot be considered by these systems.

## 7 Conclusion

Driven by the problems software engineers face when working simultaneously on multiple tasks and using a large number of artifacts, we designed and implemented a context-aware artifact switching tool, called *Switch!*. First evaluations based on simulations and short usage sessions are promising. Our approach seems to protect engineers from “interruption noise” and reduces the time needed for locating and reproducing needed information. The current version is able to handle source code documents, PDFs, emails and web pages. For the next step we plan to evaluate *Switch!* in a large industrial setting, by measuring saved context switching time. For this *Switch!* requires refinements and extensions to other artifact and task types. At the same time we use this earlier version to identify further heuristics and fine-tune the recommendation approach. Future directions are to investigate graph based artifact relationships modeling for work-flow matching and to integrate community aspects by accessing social media platforms.

## References

- [ASH06] Ankolekar, A.; Sycara, K.; Herbsleb, J. et al.: Supporting online problem-solving communities with the semantic web. In Proceedings of the 15th international conference on World Wide Web, 2006.
- [BSW08] Bernstein, M.; Shrager, J.; Winograd, T.: Taskposé: exploring fluid boundaries in an associative window visualization. In Proceedings of the 21st annual ACM symposium on User interface software and technology, 2008.
- [DDJ05] Dragunov, A.; Dieterich, T.; Johnsrude, K.: TaskTracer: a desktop environment to support multi-tasking knowledge workers. In Proceedings of the 10th international conference on Intelligent User Interfaces, 2005.
- [GM04] González, K.; Mark, G.: "Constant, constant, multi-tasking craziness": managing multiple working spheres. In Proceedings of the SIGCHI conference on Human factors in computing systems, 2004.
- [HM08] Happel, H.J.; Maalej, W.: Potentials and challenges of recommendation systems for software development. In RSSE '08: Proceedings of the 2008 international workshop on Recommendation systems for software engineering, 2008.
- [Kar] Karlin, J.: Launchy. <http://launchy.net>.
- [KDV07] Ko, A.; DeLine, R.; Venolia, G.: Information needs in collocated software development teams. In Proceedings of the 29th international conference on Software Engineering, 2007.
- [KM06] Kersten, M.; Murphy, G.: Using task context to improve programmer productivity. In Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering, 2006.
- [lau] LaunchBar. <http://www.obdev.at/launchbar/>.

- [Ma09] Maalej, W.: Task-First or Context-First? Tool Integration Revisited. In Proceedings of the ACM/IEEE International Conference on Automated Software Engineering. IEEE Computer Society, 2009.
- [MGH05] Mark, G.; Gonzalez, V.; Harris, J.: No task left behind?: examining the nature of fragmented work. In Proceedings of the SIGCHI conference on Human factors in computing systems, 2005.
- [MH08] Maalej, W.; Happel, H.: A lightweight approach for knowledge sharing in distributed software teams. In Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management, Jan 2008.
- [MH09] Maalej, W.; Happel, H.: From work to word: How do software developers describe their work? In Proceedings of MSR '09. 6th IEEE International Working Conference on Mining Software Repositories, 2009.
- [Ol06] Oliver, N.; Smith, G.; Thakkar, C.; Surendran, A: SWISH: semantic analysis of window titles and switching history. In Proceedings of the 11th international conference on Intelligent user interfaces, 2006.
- [qui] Quicksilver. [http://docs.blacktree.com/quicksilver/what\\_is\\_quicksilver](http://docs.blacktree.com/quicksilver/what_is_quicksilver).
- [RWZ10] Robillard, M.; Walker, R.; Zimmermann, T.: Recommendation Systems for Software Engineering. IEEE Software, 2010.
- [Sm03] Smith, G.; Baudisch, P.; Robertson, G.; Czerwinski, M.: GroupBar: The TaskBar Evolved. In Proceedings of OZCHI, 2003.
- [Sea99] Seaman, C.: Qualitative Methods in Empirical Studies of Software Engineering. IEEE Trans. Softw. Eng., 25(4):557–572, 1999.
- [Sh08] Shen, J.; Geyer, W.; Muller, M.; Dugan, C.; Brownholtz, B.; Millen, D.: Automatically finding and recommending resources to support knowledge workers' activities. In Proceedings of the 13th international conference on Intelligent user interfaces, 2008.
- [Sh09] Shen, J.; Irvine, J.; Bao, X.; Goodman, M.; Kolibaba, S.; Tran, A.; Carl, F.; Kirschner, B.; Stumpf, S.; Dieterich, T.: Detecting and correcting user activity switches: algorithms and interfaces. In Proceedings of the 13th international conference on Intelligent user interfaces, 2009.
- [ZG06] Zou, L.; Godfrey, M.: An Industrial Case Study of Program Artifacts Viewed During Maintenance Tasks. In Proceedings of the 13th Working Conference on Reverse Engineering, 2006.

# Adding Semantics to Social Software Engineering: (Re-)Using Ontologies in a Community-oriented Requirements Engineering Environment

Steffen Lohmann<sup>1</sup> and Thomas Riechert<sup>2</sup>

<sup>1</sup>DEI Laboratory, Computer Science Department  
Carlos III University of Madrid, Spain  
slohmann@inf.uc3m.es

<sup>2</sup>Agile Knowledge Engineering and Semantic Web  
Department of Computer Science, University of Leipzig, Germany  
rieichert@informatik.uni-leipzig.de

**Abstract:** Social Software is typically characterized by low formal semantics and weakly structured contents. Software Engineering, in contrast, requires at least a certain degree of formality and structure. In order to face these opposing demands, we propose to ground Social Software Engineering on concepts and technologies from the Semantic Web. In particular, we present a Semantic Wiki based approach in this paper that uses well-known ontologies to represent activities and artifacts that emerge in an online environment for community-oriented Requirements Engineering. We illustrate the benefits of reusing these domain-independent ontologies and show how they can fruitfully be combined and interlinked in one common upper ontology.

## 1 Introduction

The development of software is usually a highly social activity that involves people with diverse backgrounds and from different domains. The crucial role of the ‘social aspects’ of Software Engineering is well-known for a long time but it receives renewed attention these days. The new interest is mainly driven by recent phenomena in the context of online communities and social interaction on the web that are often subsumed under terms such as “Social Software”, “Web 2.0”, or “Social Web” [Por08]. Understanding these phenomena and their underlying principles and success factors and exploiting them to improve the support for social interaction in Software Engineering is currently a main goal in several research efforts and topic in a number of workshops.<sup>1</sup>

In this context, the term “Social Software Engineering” (SSE) has been coined to emphasize the importance of social aspects in software development and to investigate the

---

<sup>1</sup>To name the acronyms of just a few workshops from the last two years that focused on social interaction in software development: SSE at SE '10, CHASE at ICSE '09 & '08, SER 2.0 at CASCON '09, SENSE at SE '09, SofTEAM at SE '09, SoSEA at ESEC/FSE '09 & ASE '08.

application of concepts, principles, and technologies from the domain of Social Software to Software Engineering.<sup>2</sup> Although it is difficult to give an exact definition for “Social Software Engineering”, two aspects are central to this understanding of software development: It is focused on lightweight, community-centered collaboration and uses online environments to share artifacts and knowledge about a software product. These online environments are either accessible via a web browser (such as a web application or website) or integrated in the participants’ working environments (such as collaboration features in an Integrated Development Environment (IDE)).<sup>3</sup>

However, in contrast to Social Software that is characterized by ad-hoc collaboration, weakly structured contents, and low formal semantics, qualitative Software Engineering usually demands structured processes, well-defined artifacts, and high formal semantics. Even though several agile development methods emerged in recent years that are build around the principles of self-organization, frequent interaction, and simplicity [Coc01], at least a certain degree of formality and structure is unavoidable for successful software development. Balancing these opposing demands is a major challenge in Social Software Engineering, as we already pointed out in [LDHH09].

We see a possible solution approach in the application of structured representation formats from the Semantic Web. As we try to show in this paper, ontologies already available in the Semantic Web can be highly suitable to represent collaboratively created artifacts, involved participants, and (self-)organizing structures emerging in online environments for Social Software Engineering.

After a brief introduction into the general topic of applying ontologies in Software Engineering in Sec. 2, we demonstrate how widely adopted ontologies of the Semantic Web can be adequately (re-)used as conceptual grounding in a web environment for community-oriented Requirements Engineering in Sec. 3. We highlight the benefits and possibilities that result from the integration of these ontologies and describe how we combined them in one common upper ontology. Finally, we summarize and discuss the benefits and limitations and draw some general conclusions in Sec. 4.

## 2 Ontologies in Software Engineering

The idea of applying ontologies in Software Engineering is not a new one. Many attempts have been made in the last few years to integrate ontological and software engineering.<sup>4</sup>

---

<sup>2</sup>The term “Social Software Engineering” is sometimes also used to describe the “engineering of social software”. However, this interpretation of the term is not in the focus of the paper.

<sup>3</sup>Similar to the nuanced differences between the meanings of the terms “cooperation” and “collaboration” [MMCM92] there exist nuanced differences between the meanings of “collaboration” and “social interaction” regarding online communities such that “social interaction” is commonly seen as being more ad-hoc, intrinsic, and hedonic but less pre-determined and goal-oriented than “collaboration”. However, these nuanced differences in term meanings shall not be further discussed or elaborated in this paper.

<sup>4</sup>A very popular and condensed definition of the term “ontology” as used in computer science is given with “a specification of a conceptualization” [Gru93]. The main difference between ontologies and conceptual models (such as entity-relationship or UML models) in Software Engineering is often seen in their different scope. Whereas the life-cycle of a model typically ends with a particular project, ontologies are usually valid for several

A popular example is OMG's "Ontology Definition Metamodel (ODM)" [Obj09] specification which defines a family of metamodels for mappings between language constructs of the Semantic Web (such as RDF and OWL) and Software Engineering (such as UML and MOF). Further ideas are expressed in the "Ontology Driven Architecture (ODA)" [TPO<sup>+</sup>06] note and the "Semantic Web Primer for Object-Oriented Software Developers" [KOTW06] of the W3C. But these are only a small portion of the many approaches that have been proposed in this field of research.

An overview on the different application areas of ontologies in Software Engineering is given by Happel and Seedorf [HS06]. They distinguish between the usage of ontologies at development and run-time. Examples for the first category are conceptual models of the problem domain, formal descriptions of components and other software artifacts, or definitions of mappings between different modeling languages (such as proposed by the ODM). At run-time, ontologies are used as part of the application logic or as a conceptual layer that supports dynamic system behavior such as adaptations or automatic discovery and composition of Web Services. Furthermore, they differ between approaches that use ontologies to model a system's context or domain and approaches where ontologies are part of the system or development infrastructure itself.

According to this categorization, our approach is best described as what Happel and Seedorf call "Ontology-enabled development (OED)" [HS06], since ontologies are used at development time to support community-oriented Requirements Engineering.

## 2.1 Related Work

Closely related to our work is research on the application of Semantic Wikis in Software Engineering. Since Semantic Wikis tackle the aforementioned problem of offering an "architecture for participation" by simultaneously maintaining structure and formality<sup>5</sup>, they can be considered the ideal tool for Social Software Engineering. However, this research area is rather young and there exist only few approaches that try to apply Semantic Wikis in Software Engineering so far.<sup>6</sup>

Early work has been evolved in the context of the RISE project: Decker et al. proposed the idea of "self-organized reuse" in which software artifacts are collaboratively created and structured in a Semantic Wiki [DRR<sup>+</sup>05]. An ontology is semi-automatically derived from these structured Wiki contents and can be reused in related projects. It can also be used for reasoning to enable recommendations and consistency checks. Decker et al. illustrate the general applicability of their approach with an example from Requirements Engineering

---

projects, as they describe a specific domain, broader application area, or some more general knowledge facts (cp. [MPH08]).

<sup>5</sup>"Semantic Wikis try to combine the strengths of Semantic Web (machine processable, data integration, complex queries) and Wiki (easy to use and contribute, strongly interconnected, collaborativeness) technologies", according to an understanding of "The Semantic Wiki Community" as expressed at <http://www.semwiki.org> at time of writing.

<sup>6</sup>An introduction into Semantic Wikis and their potentials to support knowledge management in Software Engineering is given in [MPH08].



and propose a basic ontology that defines initial templates for five document types. However, their descriptions remain on a conceptual level; a concrete implementation within a Semantic Wiki or a running prototype is not presented.

A prototypical implementation of a Semantic Wiki for Software Engineering is presented by Happel and Seedorf with their Ontobrowse system [HS07]. Ontobrowse adopts the Semantic Wiki paradigm to share knowledge about software architectures, in particular Service-oriented Architectures (SOAs). The system uses the Web Ontology Language (OWL) to represent architectural descriptions and defines an initial ontology for documenting SOAs that can easily be adapted and extended by a responsible admin.

## 2.2 Discussion of Related Work

Although several approaches have been proposed that apply ontologies in software development, we are not aware of any attempt that systematically reuses and combines more general ontologies from the Semantic Web. Existing approaches either define their own ontologies, reuse software-related ontologies, or provide a framework or infrastructure that assists in the development of ontologies as software artifacts. However, many activities and artifacts are not specific to Software Engineering but can also be found in the same or a similar form in other contexts. This is also and especially the case for community-oriented communication and collaboration that is similarly implemented in various online environments following reoccurring patterns [SL07].

Many widely adopted ontologies exist already in the Semantic Web that are suitable to represent these reoccurring patterns and can therefore be fruitfully reused in Software Engineering. Such a cross-domain reuse has several benefits as we aim to illustrate in the following by an example of Social Software Engineering.

## 3 Ontologies for Community-oriented Requirements Engineering

We developed a web environment for community-oriented Requirements Engineering within the SoftWiki project [Sof09] that borrows several Social Software concepts to support lightweight collaboration among geographically distributed stakeholders [LDHH09]. It aims to foster a more direct engagement of larger groups of stakeholders in the collection, discussion, development, and structuring of software requirements and focuses on early phases of Requirements Engineering. Especially stakeholders that are not familiar with formal Requirements Engineering and corresponding tools are enabled to express their ideas, wishes, and requirements regarding a planned software product.<sup>7</sup>

The web environment is based on OntoWiki, a comparatively mature and feature-rich Semantic Wiki that has been considered particularly suitable for application in Software En-

---

<sup>7</sup>Please refer to [LDHH09] for a detailed description of the collaborative features that are provided by the web environment.

gineering [MPH08, LRA08]. Figure 1 shows a screenshot of the web environment that is conceptually divided into its different user interface parts and annotated with some of the ontology concepts used to represent the corresponding artifacts and information pieces. The web environment has been realized as an OntoWiki plugin and the applied ontologies are all imported and integrated in one common upper ontology that is based on the Semantic Web description languages RDF(S) and OWL [AH08].<sup>8</sup>

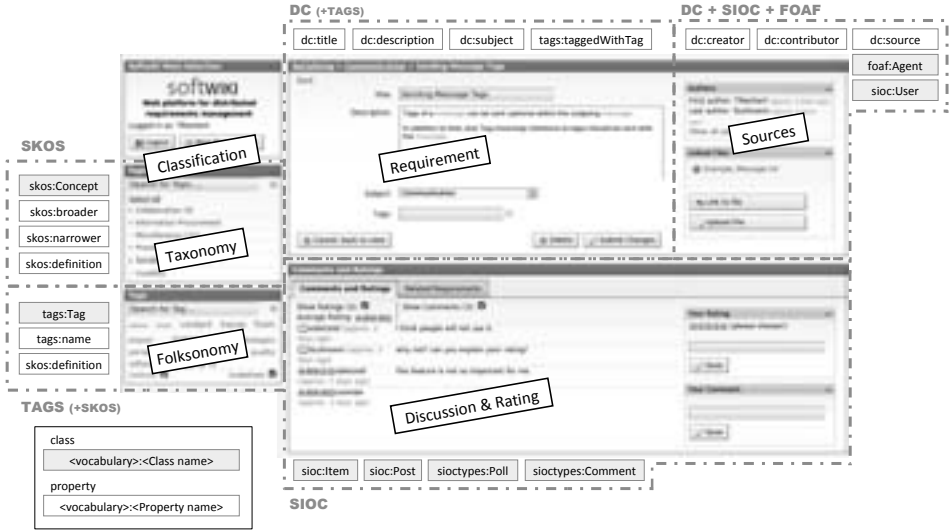


Figure 1: Screenshot of the web environment for community-oriented Requirements Engineering; divided into its conceptual structure and annotated with some of the used ontology concepts.

### 3.1 Representation of Requirements Metadata via Dublin Core

Following the idea of community-oriented Requirements Engineering, requirements are usually collaboratively edited by all registered users of the web environment [LDHH09]. Basically, a single requirement can be seen as a typical information resource with well-known properties, such as a title, a description, and one or more authors that created the requirement. A set of fifteen often used properties for the description of information resources is defined by the “Dublin Core Metadata Element Set” [Dub08]. We used several of these properties to semantically annotate requirements in our web environment, namely title, description, creator, contributor, subject, and source.

Because such metadata is often valuable in the analysis, refinement, and prioritization of requirements (e.g., to get back to the author(s) or document source of a requirement), maintaining it across different tools is usually of high interest. For that reason, common requirements exchange formats include some metadata elements (e.g., RIF [WH05]).

<sup>8</sup>A demo installation of the web environment can be tested online at: <http://softwiki.de/demo/english>

However, using Dublin Core instead to describe basic metadata extends the range of tools that can be used for accessing and refining the requirements to all tools that are capable to read and interpret this standard, including non-CASE tools. For instance, sophisticated word processing software might be used to revise and spell-check requirements without a loss of crucial metadata.

With Dublin Core, it is also possible to represent basic information for revision control in order to maintain traceability in certain cases (e.g., in collaborative editing). Of course, the metadata properties defined by Dublin Core are not sufficient to represent all information that might be captured about a requirement in all possible cases. But they provide at least a major metadata subset that can be valuably reused in Software Engineering.

### 3.2 Classification of Requirements via SKOS and TAGS

The web environment offers two ways of classifying requirements: They can either be assigned to a concept of a pre-defined taxonomy or be enriched with an arbitrary number of freely chosen keywords (so-called “tags”). These tags are aggregated to a “folksonomy” [SCH08] that is visualized as “tag cloud” [SCH08] in the user interface beneath the taxonomy tree (see Fig. 1) and can be used for filtering and navigation [LDHH09].

The taxonomy is represented via the “Simple Knowledge Organization System (SKOS)” [MB09]. SKOS has been developed specifically for the definition of controlled vocabularies and is – such as Dublin Core – formally described in an RDF schema that we import. In concrete, we use the `broader` and `narrower` properties to represent the hierarchical taxonomy structure and the class `Definition` to add definitions to concepts. SKOS allows to represent the taxonomy separately from its requirements what eases its export and import. This idea is similar to that of related work [DRR<sup>+</sup>05, HS07, LRA08], except that a standardized vocabulary like SKOS extends the range of applications and projects that can easily work with the taxonomies.

The tags of the folksonomy are represented by the “TAGS ontology” [New05]. Since we defined `Tag` to be a subclass of `SKOS Concept`, a transformation between both is easily possible. For instance, selected tags might be enriched with a definition and/or added to the taxonomy if desired [LDHH09].

### 3.3 Representation of Stakeholders and Discussions via FOAF and SIOC

Ontologies are also applied to represent the stakeholders that use the web environment. Here, we imported the FOAF (“Friend of a Friend”) vocabulary that allows the representation of persons and their interrelations [BM10]. In particular, we use the FOAF class `Agent` and its subclasses `Person`, `Group`, and `Organization`. However, further FOAF concepts might be linked if desired in order to express additional information about stakeholders, such as contact details or projects the stakeholders are assigned to. Furthermore, the FOAF structure can easily be exported and visualized as social graph with



### 3.4 Integration in the Upper Ontology SWORE

All ontologies were integrated in one common upper ontology that describes their interrelations. The basic building blocks of Requirements Engineering are represented by the three top-level classes *Requirement*, *Source*, and *Reference Point*: Requirements are derived from sources (e.g., stakeholders or documents) and are classified in a certain way (e.g., by adding a concept or tag). The basic structure of this upper ontology (that we call SWORE) is visualized in Fig. 2.<sup>9</sup>

In addition, the SWORE ontology defines a number of domain-specific concepts that are not covered by the imported ontologies but required to represent the implemented features of community-oriented Requirements Engineering (these elements are marked with the namespace `req` in Fig. 2). For instance, a set of pre-defined relation types is provided that describes dependencies between requirements, such as *details*, *entails*, or *invalidates*. It also supports the definition of pointers to parts of a system or prototype if requirements should be linked with these in certain cases [LR08]. Furthermore, it defines sub-types for ratings that allow an advanced reviewing and prioritization of requirements.

The web environment provides a REST interface that enables access to the structured requirements data in the RDF querying language SPARQL. Instance data of SWORE and all integrated ontologies can also be exported and imported in RDF format. This flexible access opens up a large range of application scenarios. As discussed above, it especially supports the utilization of the requirements data in other tools by maintaining semantic interoperability.

## 4 Discussion

Since Software Engineering is highly interdisciplinary, many of its aspects – especially domain-independent ones – are often already formally described by well-designed ontologies of the Semantic Web. Reusing these ontologies can be valuable, as we tried to point out in this paper. We presented an online environment for community-oriented Requirements Engineering as illustration and proof-of-concept, but the general idea is also applicable to other parts of Software Engineering.

A key benefit of this cross-domain reuse of ontologies is the resulting interoperability with further tools, including tools that have not been developed specifically for Software Engineering. This opens up new opportunities to utilize, enhance, and analyze artifacts and metadata. We addressed some examples in this paper, such as an external refinement of the classification structure or an advanced analysis of the social network. The modular structure of the SWORE ontology with its clear conceptual separation additionally facilitates access to single, integrated ontologies. However, due to the ontologies' high degree of self-description (e.g., typed links, class and property descriptions, etc.), all instance data

---

<sup>9</sup>An earlier version of SWORE (without integrated ontologies) is introduced in [RLL07]. The current version is available at: <http://softwiki.de/swore>

can usually also be accessed by more generic Semantic Web tools (e.g., ontology editors, inference engines, etc.) and further processed in various ways, including reasoning and consistency checking.

However, also from a non-technical and non-tool-oriented perspective can the presented approach of ontology reuse be valuable, since it fosters a shared understanding and common practices that are based on elaborated knowledge. Ontologies are normally developed by experts in a field; reusing these ontologies can therefore avoid redundant modeling effort and lowers the risk for wrong interpretations and misconceptions.

Of course, not all aspects of software development can be captured by available ontologies. Additional modeling is usually required if a large part of the collected artifacts and metadata should be formally represented, as in case of our community-oriented Requirements Engineering approach. Furthermore, grounding Software Engineering completely on ontologies raises issues of performance and scalability. Although large improvements have been made in this area recently, short response times and fluent interaction on large-scale datasets are often still a problem with the description languages of the Semantic Web [BS09].

The SWORE ontology describes only a small subset of the many aspects that are part of Requirements Engineering. However, our goal was not a comprehensive domain description but rather the development of an ontology that formally represents artifacts emerging in an online environment for community-oriented Requirements Engineering. SWORE might nevertheless be a good starting point for the development of a more comprehensive ontology for Requirements Engineering. Thus, future work includes an extension of SWORE by integrating further Semantic Web ontologies where applicable and defining domain-specific concepts where not.

## References

- [AH08] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann, 2008.
- [BM10] D. Brickley and L. Miller. FOAF Vocabulary Specification, Version 0.97. <http://xmlns.com/foaf/spec/>, 2010.
- [BS09] C. Bizer and A. Schultz. The Berlin SPARQL Benchmark. *International Journal of Semantic Web and Information Systems*, 5(2):1–24, 2009.
- [Coc01] A. Cockburn. *Agile Software Development: Software Through People*. Addison-Wesley Longman, 2001.
- [DRR<sup>+</sup>05] B. Decker, E. Ras, J. Rech, B. Klein, and C. Hoecht. Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. In *Proc. of SWESE'05*, 2005.
- [Dub08] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1. <http://www.omg.org/spec/ODM/1.0>, 2008.
- [Gru93] T.R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

- [HS06] H.-J. Happel and S. Seedorf. Applications of Ontologies in Software Engineering. In *Proc. of SWESE'06*, 2006.
- [HS07] H.-J. Happel and S. Seedorf. Ontobrowse: A Semantic Wiki for Sharing Knowledge about Software Architectures. In *Proc. of SEKE'07*, pages 506–512. W3C, 2007.
- [KHA09] T. Kramer, T. Hildenbrand, and T. Acker. Enabling Social Network Analysis in Distributed Collaborative Software Development. In *Software Engineering 2009 – Workshopband*, pages 255–266. GI, 2009.
- [KOTW06] H. Knublauch, D. Oberle, P. Tetlow, and E. Wallace. A Semantic Web Primer for Object-Oriented Software Developers. <http://www.w3.org/TR/sw-oosd-primer/>, 2006.
- [LDHH09] S. Lohmann, S. Dietzold, P. Heim, and N. Heino. A Web Platform for Social Requirements Engineering. In *Software Engineering 2009 – Workshopband*, pages 309–315. GI, 2009.
- [LR08] S. Lohmann and A. Rashid. Fostering Remote User Participation and Integration of User Feedback into Software Development. In *Proc. of I-USED'08*, 2008.
- [LRA08] S. Lohmann, T. Riechert, and S. Auer. Collaborative Development of Knowledge Bases in Distributed Requirements Elicitation. In *Software Engineering 2008 – Workshopband*, pages 22–28. GI, 2008.
- [MB09] A. Miles and S. Bechhofer. SKOS Simple Knowledge Organization System Reference. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>, 2009.
- [MMCM92] P.W. Mattessich, M. Murray-Close, and B.R. Monsey. *Collaboration: What Makes It Work*. Amherst H. Wilder Foundation, 1992.
- [MPH08] W. Maalej, D. Panagiotou, and H.-J. Happel. Towards Effective Management of Software Knowledge Exploiting the Semantic Wiki Paradigm. In *Software Engineering 2008*, pages 183–197. GI, 2008.
- [New05] R. Newman. Tag Ontology Design. <http://www.holygoat.co.uk/projects/tags/>, 2005.
- [Obj09] Object Management Group. Ontology Definition Metamodel, Version 1.0. <http://www.omg.org/spec/ODM/1.0>, 2009.
- [Por08] J. Porter. *Designing Social Web Applications*. New Riders, 2008.
- [RLL07] T. Riechert, K. Lauenroth, and J. Lehmann. SWORE - SoftWiki Ontology for Requirements Engineering. In *Proc. of CSSW'07*, pages 111–118. GI, 2007.
- [SCH08] J. Sinclair and M. Cardew-Hall. The Folksonomy Tag Cloud: When is it Useful? *Journal of Information Science*, 34(1):15–29, 2008.
- [SL07] T. Schümmer and S. Lukosch. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons, 2007.
- [Sof09] SoftWiki: Distributed, End-user Centered Requirements Engineering for Evolutionary Software Development. <http://softwiki.de/netzwerk/en/>, 2009.
- [TPO<sup>+</sup>06] P. Tetlow, J. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. <http://www.w3.org/2001/sw/BestPractices/SE/ODA/060103/>, 2006.
- [WH05] R. Wiebel and S. Höh. Requirements Interchange Format (RIF), Version 1.0. <http://www.automotive-his.de/rif/>, 2005.

# Diagnosing Inconsistent Requirements Preferences in Distributed Software Projects

A. Felfernig, M. Schubert, M. Mandl, and P. Ghirardini

Applied Software Engineering  
Graz University of Technology  
Inffeldgasse 16b  
A-8010 Graz, Austria

{alexander.felfernig, monika.schubert, monika.mandl, philipp.ghirardini}@ist.tugraz.at

**Abstract:** Requirements specification is often performed by stakeholders who are organizationally and geographically distributed. In these scenarios preferences regarding requirements can become inconsistent and stakeholders need assistance in identifying reasonable tradeoffs. In this paper we introduce a basic approach that actively supports distributed groups of stakeholders in the specification of consistent requirement preferences. The approach automatically detects *minimal* sets of preferences that have to be adapted in order to restore global consistency in the given collection of preferences. This approach can be seen as a core technology of requirements management processes in (distributed) software projects.

## 1 Introduction

Effective requirements specification is extremely important for successful software development [YWK08]. Especially in organizationally and geographically distributed projects the identification and resolution of inconsistent stakeholder preferences is crucial, for example, *one stakeholder could prefer the inclusion of database system X whereas another stakeholder would like to have the database system Y*. The distributed characteristic of software development makes consistency maintenance of requirements specifications a challenging task. Requirement preferences of stakeholders at different geographical locations have to be integrated. This can result in inconsistent preferences that have to be resolved. The most valuable result for all stakeholders is the one that takes the existing preferences into account as much as possible.

In this paper we show how the concepts of model-based diagnosis [Rei87] can be exploited for determining minimal sets of stakeholder preferences that have to be changed in order to make the global set of preferences consistent. We are interested in minimal adaptations of existing preferences that can be offered to stakeholders engaged in a requirements *negotiation process*, for example, *which requirements should be integrated in the next software release?*

Note that our work relies on the existence of a formal representation of interdependencies in the given set of requirements (see, e.g., [LRA08]) and the existence of a constraint solver [Tsa93] that allows to reason about the consistency of requirements.



The remainder of this paper is organized as follows. In Section 2 we introduce a simple set of requirements with a corresponding model about the dependencies between those requirements. Furthermore, we provide a set of requirements preferences that have been articulated by stakeholders. In Section 3 we introduce the basic concepts for identifying minimal sets of requirements that have to be adapted such that global consistency can be achieved. In Section 4 we provide a discussion of related work. With Section 5 we conclude the paper.

## 2 Working Example

We now introduce a basic set of abstract requirements (including their dependencies) that will serve as a simple working example throughout this paper. We model requirements in the form of constraint variables  $r_i$  with  $\text{domain}(r_i)=\{0,1\}$  where  $r_i=0$  denotes the fact that requirement  $r_i$  should not be integrated in the next software release and  $r_i=1$  indicates the fact that requirement  $r_i$  must be integrated in the next release. In our working example we have to deal with the following set of requirements:

$$R = \{r_1, r_2, r_3, r_4, r_5\}$$

As already mentioned, the domain of each  $r_i$  is specified with  $\text{domain}(r_i)=\{0,1\}$ .<sup>1</sup> On the basis of this given set of finite domain variables representing requirements, we are able to add additional constraints that represent formal dependencies between requirements.

*Requires constraints* indicate the fact that a certain requirement  $r_a$  is allowed to be integrated in a release only if a certain requirement  $r_b$  is also part of the same release, for example, *if the average query response time should be below 100ms, the database system of company A has to be purchased.*

*Incompatibility constraints* denote the fact that a certain requirement  $r_a$  must not be integrated in the same release with requirement  $r_b$ , for example, *the calculation of optimal solutions is incompatible with systems response time below 2 seconds.*

*Resource constraints* define basic properties that must be satisfied by the requirements integrated in a release, for example, *the overall efforts for the next release must not exceed a certain upper bound.*

These are typical constraints included in most of the existing requirements engineering ontologies such as SWORE [LRA08]. Note that the above mentioned constraints act as a kind of background knowledge, i.e., knowledge that is assumed to be correct and that has to be taken into account for planning the new release. In our working example, we will take into account the following set of abstract constraints (background knowledge):  $C = \{c_1, c_2, c_3, c_4, c_5\}$ .

$$c_1: r_1 = 1 \Rightarrow r_3 = 1; \quad /* \text{the inclusion of } r_1 \text{ requires the inclusion of } r_3 */$$

---

<sup>1</sup> Note that requirements can as well be related to variables with domain size  $> 2$ , however, binary variables have been chosen for this working example.

- $c_2: \text{not}(r_1 = 1 \text{ and } r_2 = 1)$       /\*  $r_1$  and  $r_2$  are incompatible \*/  
 $c_3: r_2 = 1 \Rightarrow (r_3 = 1 \text{ and } r_4 = 1)$       /\* the inclusion of  $r_2$  requires  $r_3$  and  $r_4$  \*/  
 $c_4: r_5 = 1 \Rightarrow (r_4 = 0 \text{ and } r_2 = 0)$       /\* including  $r_5$  requires the exclusion of  $r_4$  and  $r_2$  \*/  
 $c_5: r_1 + r_2 + r_3 + r_4 + r_5 < 4$       /\* at most 3 requirements can be included \*/

In order to complete our release planning task we have to collect the requirements preferences of the different engaged stakeholders. For modelling the preferences in our working example, we allow three different types of constraints:

First,  $include(r_i)$  denotes that fact that a certain stakeholder wants to have included requirement  $r_i$  in the next software release.

Second,  $exclude(r_i)$  denotes that fact that a certain stakeholder does not want to include requirement  $r_i$  in the next release.

Finally  $prefer(r_i, r_j)$  denotes the fact the a certain stakeholder prefers the inclusion of requirement  $r_i$  over the inclusion of requirement  $r_j$ .

These three types of requirements preferences have been introduced for the purposes of our working example, however, can be extended by different other constraint types seen as useful in a certain application context. For our simple working example we assume that *three stakeholders* are specifying their preferences regarding the given set of requirements  $R = \{r_1, r_2, r_3, r_4, r_5\}$ . In this context,  $PREF = \cup \text{pref}_{ij}$  denotes the set of stakeholder preferences where  $\text{pref}_{ij}$  denotes the  $j^{\text{th}}$  preference of stakeholder  $i$ .

- $\text{pref}_{11}: r_1 = 1;$       /\* include  $r_1$  \*/  
 $\text{pref}_{12}: r_2 = 1;$       /\* include  $r_2$  \*/  
 $\text{pref}_{21}: r_2 = 0;$       /\* exclude  $r_2$  \*/  
 $\text{pref}_{22}: r_5 = 0;$       /\* exclude  $r_5$  \*/  
 $\text{pref}_{23}: r_3 = 1 \text{ and } r_4 = 0$       /\* prefer  $r_3$  over  $r_4$  \*/  
 $\text{pref}_{31}: r_2 = 1 \text{ and } r_1 = 0$       /\* prefer  $r_2$  over  $r_1$  \*/  
 $\text{pref}_{32}: r_4 = 1 \text{ and } r_3 = 0$       /\* prefer  $r_4$  over  $r_3$  \*/

A corresponding release planning task can be defined as follows.

**Definition 1 (Release Planning Task).** A release planning task can be defined as a constraint satisfaction problem [Tsa93]  $(R, C, PREF)$ .  $R = \cup r_i$  represents a set of finite domain variables where  $\text{dom}(r_i) = \{0, 1\}$ .  $C = \cup c_j$  represents a set of constraints on the variables in  $R$ . Finally,  $PREF = \cup \text{pref}_{ij}$  is a set of stakeholder preferences.

On the basis of this definition of a release planning task we can now introduce the definition of a solution for a release planning task (also denoted as *release plan*).

**Definition 2 (release plan):** a solution (release plan) for a given release planning task  $(R, C, PREF)$  is represented by an instantiation  $I = \{r_1=v_1, r_2=v_2, \dots, r_s=v_s\}$  of the variables in  $R$ , where  $v_o \in \text{dom}_o$  and  $\text{dom}_o$  denotes the domain of variable  $r_o$ .

A release plan is *consistent* if the assignments in  $I$  are consistent with the constraints in  $C$  and the stakeholder preferences in  $PREF$ . Furthermore, a release plan is *complete* if it is consistent and all the variables in  $R$  have an assigned value. Finally, a release plan is *valid*, if it is both consistent and complete.

The release plans for the release planning task  $(R, C, \{\})$  are depicted in Table 1. If we want to include the additionally defined set of stakeholder preferences  $PREF = \{\text{pref}_{11}, \text{pref}_{12}, \text{pref}_{21}, \text{pref}_{22}, \text{pref}_{23}, \text{pref}_{31}, \text{pref}_{32}\}$ , the constraint solver is not able to calculate a corresponding release plan for the release planning task  $(R, C, PREF)$ . The question we have to answer now is: *which is the minimal set of stakeholder preferences that have to be changed in order to be able to calculate a release plan for the defined release planning task?* An algorithm for calculating such minimal sets of stakeholder preferences will be sketched in the following section.

$r_1$	$r_2$	$r_3$	$r_4$	$r_5$
0	0	0	0	0
0	0	1	0	0
1	0	1	0	0
0	0	0	1	0
0	0	1	1	0
1	0	1	1	0
0	1	1	1	0
0	0	0	0	1
0	0	1	0	1
1	0	1	0	1

Table 1: Example release plans for the release planning task  $(R, C, \{\})$ .

Note that we do not expect a complete specification of requirements interdependencies and stakeholder preferences. Stakeholders do not necessarily know all the existing interdependencies between the requirements and also are often not able to completely enumerate their preferences regarding the given set of requirements. However, even for incomplete preferences and dependencies sets our approach is able to effectively support the effective identification and repair of inconsistencies.

### 3 Diagnosing Inconsistent Preferences

In situations where no release plan can be identified for the release planning task  $(R, C, PREF)$  we have to activate a diagnosis functionality [Rei87] that helps us to identify those stakeholder preferences that are responsible for the given inconsistency, i.e., preferences that have to be changed in order to be able to find at least one consistent release plan. We are interested in *minimal changes* since we want to keep the original set of stakeholder preferences the same as much as possible. The calculation of minimal changes to stakeholder preferences is based on the identification of *conflict sets* [Jun04].

**Definition 3 (Conflict Set).** A conflict set is a set of constraints  $CS \subseteq \text{PREF}$  s.t.  $CS \cup C \cup R$  does not allow the calculation of a release plan.  $CS$  is minimal if there does not exist a conflict set  $CS'$  with  $CS' \subset CS$ .

In the working example we are able to identify the three conflict sets  $CS_1: \{\text{pref}_{11}, \text{pref}_{12}\}$ ,  $CS_2: \{\text{pref}_{11}, \text{pref}_{31}\}$ , and  $CS_3: \{\text{pref}_{12}, \text{pref}_{21}\}$ . All three are conflict sets since  $\{\text{pref}_{11}, \text{pref}_{12}\} \cup \text{PREF}$ ,  $\{\text{pref}_{11}, \text{pref}_{31}\} \cup \text{PREF}$ , and  $\{\text{pref}_{12}, \text{pref}_{21}\} \cup \text{PREF}$  are inconsistent, i.e., no release plan exists. Furthermore, all three conflict sets are minimal: there does not exist a subset  $CS'$  such that  $CS' \subset CS$  and  $CS' \cup \text{PREF}$  is inconsistent.

In order to restore consistency in the given set of preferences  $\text{PREF}$  we have to resolve each of the identified conflict sets by deleting at least one of the elements in the conflict set. A systematic way to calculate minimal diagnosis is to apply the concepts of *model-based diagnosis* (for details, see [Rei87]).

**Definition 4 (Stakeholder Requirements Diagnosis).** A stakeholder requirements diagnosis is defined as a set  $\Delta \subseteq \text{PREF}$  s.t. the release planning task  $(R, C, \text{PREF})$  has a corresponding solution. A diagnosis  $\Delta \subseteq \text{PREF}$  is minimal if there does not exist any stakeholder requirements diagnosis  $\Delta'$  with  $\Delta' \subset \Delta$ .

The major concept discussed in [Rei87] is the Hitting Set Directed Acyclic Graph (HSDAG) algorithm that allows us to calculate all minimal sets of stakeholder preferences that have to be adapted in order to be able to identify at least one release plan. The construction of such a HSDAG is exemplified in Figure 1. The approach assumes the existence of a corresponding algorithm that supports the identification of minimal conflict sets. In our prototype implementation we apply the QuickXplain algorithm proposed in [Jun04].

Let us now assume that our conflict detection algorithm returns  $CS_1: \{\text{pref}_{11}, \text{pref}_{12}\}$  as the first conflict set. Since QuickXplain guarantees the minimality of calculated conflict sets, we only have to delete at least one element from each identified conflict set in order to be able to calculate a corresponding minimal diagnosis, i.e., a set  $\Delta$  of stakeholder preferences that have to be deleted from  $\text{PREF}$  in order to make the release planning task  $(R, C, \text{PREF}-\Delta)$  consistent (at least one release plan can be found).

To resolve the conflict set  $CS_1$ , we can either delete the stakeholder preference  $\text{pref}_{11}$  or the preference  $\text{pref}_{12}$ . We now decide to delete the preference  $\text{pref}_{11}$  from the given set of preferences  $\text{PREF}$  and again call the conflict detection algorithm with the reduced set of preferences  $\text{PREF} = \{\text{pref}_{12}, \text{pref}_{21}, \text{pref}_{22}, \text{pref}_{23}, \text{pref}_{31}, \text{pref}_{32}\}$ . For the given release planning task  $(R, C, \text{PREF}-\{\text{pref}_{11}\})$  we are still not able to identify a solution and the conflict detection algorithm returns the next conflict set which is  $CS_3: \{\text{pref}_{12}, \text{pref}_{21}\}$ .

Again, we have two alternatives to resolve the conflict. If we delete the element  $\text{pref}_{12}$ , we are able to derive the first diagnosis  $\Delta_1 = \{\text{pref}_{11}, \text{pref}_{12}\}$  since there exists at least one solution for the release planning task  $(R, C, \text{PREF} - \{\text{pref}_{11}, \text{pref}_{12}\})$ . If we delete the element  $\text{pref}_{21}$ , we are able to derive the second diagnosis  $\Delta_2 = \{\text{pref}_{11}, \text{pref}_{21}\}$  since at least one solution exists for the release planning task  $(R, C, \text{PREF} - \{\text{pref}_{11}, \text{pref}_{21}\})$ . Note that if a diagnosis has been identified, the conflict detection algorithm will not be able to calculate further conflicts [Rei87, Jun04].

If we decide to delete the second element of  $\text{CS}_1$  (which is  $\text{pref}_{12}$ ) from the given set of stakeholder preferences in  $\text{PREF}$ , the conflict detection algorithm returns another minimal conflict set for the remaining preferences  $\text{PREF} - \{\text{pref}_{12}\}$  which is  $\text{CS}_2: \{\text{pref}_{11}, \text{pref}_{31}\}$ . By resolving  $\text{CS}_2$ , we are able to derive one further diagnosis which is  $\Delta_3 = \{\text{pref}_{12}, \text{pref}_{31}\}$ .  $\Delta_3$  is a diagnosis since the release planning task  $(R, C, \text{PREF} - \Delta_3)$  allows the calculation of at least one solution. Note that all three identified diagnoses  $\Delta_i$  are minimal since there does not exist a  $\Delta' \subset \Delta$  that fulfills the property of a diagnosis.

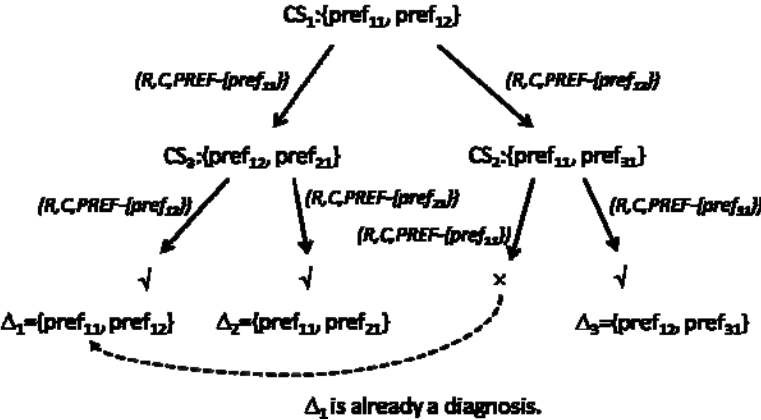


Figure 1. Calculating minimal diagnoses with the HSDAG algorithm [Rei87].

This way we are able to determine all minimal sets of stakeholder preferences that have to be changed (or deleted) in order to be able to find a release plan – such alternative diagnoses are then displayed to the stakeholder who is in charge of deciding on which minimal set of preferences should be changed.

The presented approach is a way to effectively support distributed decision making processes in situations where stakeholders articulate contradicting preferences regarding the set of requirements to be integrated into the next software release. Currently we have implemented the sketched conflict detection and diagnosis algorithms – an evaluation of those algorithms clearly shows their applicability in interactive settings [FFS09].

We are currently working on the design of a graphical user interface that supports requirements specification processes in distributed scenarios. With this implementation we want to evaluate the applicability of diagnosis concepts.

## 4 Related Work

The authors of [FFJ04] show how to apply the concepts of model-based diagnosis for the identification of minimal sets of faulty constraints in configuration knowledge bases – in this context, conflicts are induced by positive and negative examples for results expected to be calculated by a configuration knowledge base. In [SPF07] such minimal sets are called minimal exclusion sets, in [God97, McS04] the complements of such sets are denoted as maximally successful sub-queries. The calculation of minimal conflict sets in our context relies on the QuickXplain algorithm which has been introduced in [Jun04]. This algorithm is based on the principle of *divide and conquer* and thus guarantees logarithmic runtime performance in terms of the needed number of consistency checks.

The major contribution of our paper is to show how to apply the concepts of model-based diagnosis and conflict detection in the context of distributed requirements engineering scenarios – this approach shows potentials to significantly improve the overall quality of requirements engineering in terms of improved effectiveness in the identification and resolution of inconsistent sets of stakeholder preferences. We are currently working on the development of algorithms that further improve the performance of the diagnosis and the underlying conflict detection. First related results have been published in [SFM09].

Knowledge-based technologies enable a formal reasoning about the properties of a given set of stakeholder preferences. There already exist applications of knowledge-based technologies in the context of requirements engineering and release planning scenarios (see, for example [REP03, Goe08] ). All those approaches rely on the existence of a corresponding requirements engineering ontology (see, for example, [LRA08]) and a formal model of the dependencies between a given set of requirements. Compared to the existing work in the field of knowledge based systems & requirements engineering our work extends existing concepts with automated diagnosis techniques that help to reason about inconsistent requirements sets and to actively support stakeholders in release planning and decision making scenarios.

## 5 Conclusions

The effective identification of inconsistent preferences is crucial for a successful stakeholder decision support in (distributed) requirements engineering scenarios. In this paper we showed how to apply the concepts of model-based diagnosis to achieve this goal. The presented diagnosis approach is a key enabling technology that allows effective consistency management in different requirements engineering scenarios, such as open source projects or large and geographically/organizationally distributed software projects. Future work will include the recommendation of diagnoses, i.e., the identification those diagnoses with the highest utility for the stakeholders engaged in the decision making process. Finally, we plan detailed empirical studies in distributed projects.

## References

- [FFJ04] Felfernig, A.; Friedrich, G.; Jannach, D.; Stumptner, M.: Consistency-based Diagnosis of configuration knowledge bases, *AI Journal*, 152(2):213-234, 2004.
- [FFS09] Felfernig, A.; Friedrich, G.; Schubert, M.; Mandl, M.; Mairitsch, M.; Teppan, E.: Plausible Repairs for Inconsistent Requirements, 21<sup>st</sup> International Joint Conference on Artificial Intelligence (IJCAI'09), Pasadena, California, pp. 791-796, 2009.
- [God97] Godfrey, P.: Minimization in cooperative response to failing database queries. *International Journal of Cooperative Information Systems*, 6(2):95-149, 1997.
- [Goe08] Göknil, A.; Kurtev, I.; van den Berg, K.: A metamodeling approach for reasoning about requirements. 4<sup>th</sup> European Conference on Model Driven Architecture – Foundations and Applications, LNCS, 5095, pp.310-325, Berlin, Germany, 2008.
- [Jun04] Junker, U.: QuickXplain: Preferred Explanations and Relaxations for Over-Constrained Problems. *AAAI'04*, San Jose, AAAI Press, pp. 167-172, 2004.
- [LRA08] Lohmann, S.; Riechert, T.; Auer, S.: Collaborative Development of Knowledge Bases in Distributed Requirements Elicitation. *Software Engineering (Workshops): Agile Knowledge Sharing for Distributed Software Teams*, pp.22-28, 2008.
- [McS04] McSherry, D.: Maximally Successful Relaxations of Unsuccessful Queries, 15<sup>th</sup> Conference on Artificial Intelligence and Cognitive Science, Galway, Ireland, pp. 127-136, 2004.
- [Rei87] Reiter, R.: A theory of diagnosis from first principles, *AI Journal*, 23(1):57-95, 1987.
- [REP03] Ruhe, G.; Eberlein, A.; Pfahl, D.: Trade-off analysis for requirements selection, *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 13(4):354-366, 2003.
- [SFM09] Solving Over-Constrained Problems through Network Analysis, *IEEE Conference on Adaptive and Intelligent Systems*, Klagenfurt, Austria, pp. 9-14, 2009.
- [SPF07] O'Sullivan, B.; Papadopoulos, A.; Faltings, B.; Pu, P.: Representative Explanations for Over-Constrained Problems, *AAAI'07*, pp. 323-328, 2007.
- [Tsa93] Tsang, E.: *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.
- [YWK08] Yang, D.; Wu, D.; Koolmanojwong, S.; Brown, A.; Boehm, B: WikiWinWin: A Wiki Based System for Collaborative Requirements Negotiation, *HICCS*, p. 24, Waikoloa, Big Island, Hawaii, 2008.

# User-Driven Requirements Engineering for Mobile Social Software

Norbert Seyff, Florian Graf

Centre for HCI Design  
City University London  
London, EC1V 0HB  
n.seyff@soi.city.ac.uk  
florian.graf.1@city.ac.uk

**Abstract:** Social software is an important phenomenon which allows end-users to be socially connected whenever and wherever they want. The broad distribution of sophisticated mobile devices strengthens the importance of Mobile Social Software while the number of mobile social applications is increasing. Emerging paradigms such as mobile and service-oriented computing raise challenges for the design and development of social software. This paper discusses a novel approach for requirements engineering which is tailored to the needs of developing Mobile Social Software. We present a mobile requirements elicitation tool allowing social software users to document needs in situ. Furthermore, the tool automatically captures contextual information. We also discuss how the gathered requirements and contextual information can inform future social software development.

## 1 Introduction

Social software is an exciting and important phenomenon in today's software and business world [Tm03] and its weaving into the fabric of daily life is faster than expected. Social software enables its users to communicate, interact and share data with other individuals and gives them the possibility to increase their social interaction. The ongoing widespread use of social software triggers new research and application challenges and opportunities [HB08], [NS08], [Or06]. Based on a literature review, we identified following emerging trends within social software: (i) provision of social software for mobile devices, (ii) integration of location-based services and social software and, (iii) integration of social software and service-centric systems.

Mobile devices such as smartphones have advanced significantly over the last years. Today's end-users are familiar with these sophisticated mobile devices and an increasing number of Mobile Social Software (MoSoSo) is available [At06]. Mobile Social Software allows end-users to be socially connected anytime, anywhere and often integrates information on location and time.



Traditional social software systems often provide a fixed, predefined functionality which is used by special-interest communities. Novel software engineering paradigms, such as service-oriented computing, drive the personalization of software systems [GTR06] and strengthen immediate system development and deployment. We foresee that service-centric Mobile Social Software services will provide just the right functionality at the right time to social software users.

Developing Mobile Social Software is challenging and little is known on software engineering methods and tools in this new field. Thus, research on social software engineering is needed to support the realization of customized and personalized Mobile Social Software for end-users and social communities. Disciplines such as requirements engineering (RE) [KS98] are important to gather an individual end-user's needs and to address the design factors critical to the success of mobile social applications [Sm05]. Research in the field of RE is considered to be particularly crucial for social software engineering [CHS06] because state-of-the-art RE approaches provide neither specific method nor tool support.

The research presented focuses on user-led RE for Mobile Social Software. More specifically, our aim is to improve support for requirements discovery for such systems by identifying individual end-user's needs and contextual information relevant to the early design process. Our goal is to address the needs of tomorrow's Mobile Social Software engineering based on research strategies that address both social and technical issues. In Section 2 we highlight the research goal in more detail and describe research questions and objectives. Section 3 presents a novel mobile requirements blogging tool for end-users, which allows them to document needs themselves in situ. Furthermore, we describe how the tool uses context sensing technologies to automatically gather information about the end-users' context and environment. Section 4 discusses how gathered individual end-user's needs and contextual information can be communicated and visualized in order to inform social software design. Section 5 reviews the research questions, gives a conclusion and highlights future work.

## **2 Research Goal and Questions**

The goal of our research is to develop and evaluate a tool-supported requirements discovery method for Mobile Social Software. We are focusing on user-led requirements documentation and in situ capturing of individual stakeholders' needs with the help of everyday mobile devices such as mobile phones. In addition to requirements discovery, the method is intended to support the on-site identification of contextual information relevant to inform system design. This tool-supported method is supposed to consider recent trends in social software including mobility, location-awareness, and service-centric system design.

Based on these goals, more specific research questions (RQ) can be identified; they define the focus of our research:

- RQ 1: How to discover the individual needs of Mobile Social Software end-users?
- RQ 2: How to identify contextual information relevant to Mobile Social Software engineering?
- RQ 3: How to distribute and communicate gathered end-users' needs and contextual information to inform future system design activities?

By their nature, most requirements elicitation approaches focus on gathering requirements fulfilling the needs of the majority of stakeholders [SFS05]. We foresee, that requirements elicitation tools installed on mobile devices such as smartphones suggest the potential to support end-users in capturing individual needs in situ. Therefore, our first research objective is to develop a mobile tool for end-users allowing them to capture their needs wherever and whenever they want.

Mobile devices provide sophisticated context sensing capabilities. For example, this includes detecting the user's position with the help of built-in GPS functionality. The second research objective is set to explore how this functionality can be used in order to automatically gather relevant contextual information for Mobile Social Software design.

Gathering individual end-user's needs is a prerequisite to building customized software systems for end-users and social communities. However, these requirements need to be distributed in order to start system development and provide a solution. Therefore, the third research objective focuses on the exploration of suitable ways to distribute and communicate information gathered with a mobile requirements elicitation tool for end-users.

### **3 Gathering Individual End-User Needs and Contextual Information**

Cultural, emotional, social and material aspects of life often trigger end-users' requirements [SFS05]. Understanding these contextual triggers and upcoming requirements is particularly important for Mobile Social Software whose scope is the support of social interaction in a volatile mobile context [CHS06]. Kolko et al. [KJR07] highlight that understanding particular characteristics and patterns of life, and investigating new locations and situations, supports stakeholders in envisioning how mobile technology can facilitate social interactions.

Our research is more particularly motivated by the work of Blomberg et al. [BBG03] who suggest that people are better able to describe what they do when they actually perform these activities. Mobile requirements elicitation tools enable end-users of future social software to blog their needs in situ. For example, while enjoying a cup of coffee with a friend, an end-user could come up with the idea that it would be nice to invite all their other friends to the café by just pressing a button of an envisioned mobile social application. Our mobile requirements elicitation tool would enable the future system end-user to immediately document this need on-site.

We envision further benefits of mobile RE tools for end-users besides enabling anytime anywhere requirements documentation. State-of-the-art requirements elicitation approaches often rely on requirements analysts to document end-users' needs [Ma04]. This is generally costly, time-consuming and it limits the number of end-users that can be involved [Se09]. Easy-to-use mobile RE tools could significantly strengthen end-user involvement in RE by providing adequate guidance and support capturing needs in a structured manner without the help of an analyst.

Following the first and second research objectives, we have developed the iRequire prototype which allows end-users' requirements, blogging and capturing contextual information about the end-user's environment.

### 3.1 iRequire: Mobile Social Software End-Users Document Needs

The development of iRequire was informed by earlier research on how mobile RE tools can help analysts document end-users' requirements in the field [Se09], [Se09]. Furthermore, we explored on-site end-users' requirements capturing in a study where end-users documented requirements with the help of standard mobile phone features (e.g. audio recording). In this study, we used Microsoft SenseCam [Ho06] to capture information on the end-users' environment.

Our vision was to provide an easy-to-use mobile tool which can be installed on widespread mobile platforms and is available to a large number of end-users. By documenting individual needs end-users provide the basis for customized and personalized Mobile Social Software. The developed iRequire prototype is available for Windows Mobile smartphones and enables Mobile Social Software end-users to blog their upcoming needs while interacting with their social community. To simplify tool usage, iRequire provides a touch screen user interface. It was designed to be a self-explanatory tool which guides Mobile Social Software end-users via a wizard-like user interface. Following four steps, end-users can document needs in a structured way.

***Taking a picture of the environment.*** After displaying the start screen, iRequire invites the future Mobile Social Software end-user to capture pictures of her surroundings or objects that are related to her need (see Figure 1). Our initial studies on using mobile phones and SenseCam equipment for end-user requirements elicitation revealed that such pictures can be used to refer to the (social) environment and provide a starting point for expressing a need. An example could be a user who takes a picture of friends sitting in a café and documents a need referring to this social information (see Figure 1).

***Documenting a need.*** In the next step, iRequire invites the end-user to blog her needs and ideas regarding an envisioned Mobile Social Software system (see Figure 1). iRequire offers the functionality to record short text-based requirements descriptions (e.g. *I would like to invite friends for a coffee at the push of a button*). However, studies with mobile RE tools for analysts [NF09] have shown that using a mobile tool's (virtual) keyboards for entering text is not always comfortable. Audio recording was often the preferred choice to capture a need. Therefore, iRequire provides a dictaphone-like feature to allow the quick documentation of upcoming ideas and requirements.



Figure 1: Taking a picture of the environment (left) and documenting a need (right) using iRequire

**Describing the relevant task and providing a rationale.** In the last elicitation step, iRequire asks the end-user to describe the task which she intends to support with her requirement and why the reported need is important to her. As for the description of her needs, the end-user can capture this information via audio recording or a short textual description. An example for a task supported by the requirement could be: *“Inform my friends that I am here at the café having a cappuccino”*. A rationale for this need could be: *“I do not want to give all my friends a call because this takes too much time”*. Gathering information on the task and the rationale clarifies the vision of the new system and helps social community members and requirements analysts understand the end-user’s individual needs.

**Reviewing a summary.** In a last step, iRequire displays a summary of the information captured. After a final confirmation, the system stores the captured information and, if requested, redirects the end-user to the start screen to capture another need.

We tried to minimize the time needed for capturing needs with iRequire. The current three elicitation steps are intended to balance out the end-user’s effort in documenting the ideas and information needed for future analysis. Time for documenting needs and ideas might be highly limited, especially during social interactions. Therefore, iRequire enables Mobile Social Software end-users to skip any of the above steps if, for example, they prefer to just provide a requirement description.

The iRequire tool enables end-users to document needs in everyday language since social software end-users are familiar with documenting information in this form (e.g. in a blog or a Wiki). However, iRequire is following the VOLERE Template [RR06] in documenting requirements identity, description, rationale and originator to support structured requirements documentation.

### **3.2 Capturing Contextual Information with iRequire**

iRequire does not only enable future Mobile Social Software end-users to blog their needs, it also captures contextual information. For example, this includes taking a picture of the environment. However, there are situations where stakeholders might not be able to take a picture of the environment or where a captured picture is not self-explanatory. Therefore, iRequire encourages end-users to provide additional contextual information by describing the task, which will be supported by the captured need, as well as documenting a need's rationale.

However, while designing iRequire, our goal was to go beyond end-user driven documentation of contextual information. Nowadays, context sensing technologies enable mobile devices to gather contextual information without end-user interaction. The current iRequire prototype provides initial capabilities for automated context sensing. Currently, iRequire can detect the end-user's position (longitude, latitude) with the help of GPS. When blogging a need, this feature automatically detects the end-user's position and stores this information in the local database. Furthermore, iRequire automatically captures the date and time when a need is documented by storing a timestamp together with the need. These features allow iRequire to gather contextual information on the end-user's environment even if the end-user does not have time to take a picture.

The automatically gathered contextual information is only relevant for RE activities and system design if end-users document requirements in situ. However, we foresee that end-users will also document requirements using iRequire in situations which are not related to needs. For example an end-user might document an idea while she is having a break in a café although the need itself is related to commuting activities, which she performed before. This means that the automatically captured contextual information might not be useful. Therefore, iRequire encourages end-users to provide additional contextual information by describing the task which will be supported by the captured need, as well as documenting a need's rationale.

## **4 Distributing End-User Needs and Contextual Information to Inform Mobile Social Software Design**

Gathering and documenting individual end-users' requirements and contextual information is seen as a first step towards the design and development of Mobile Social Software that is customized to the needs of a particular end-user or that of a social community. Following the third research objective, we have started to explore strategies on how to distribute and visualize the gathered information and how to enable members of a social community to discuss discovered needs in order to inform system design.

## 4.1 Distribution and Visualization of End-User Needs

The information captured with iRequire is stored in a local database on the mobile device. Desktop PC equipment can be used to access this information. End-users themselves can then actively distribute this information. This could include emailing relevant needs to peers within a social community. However, we envision to provide more systematic approaches on how gathered end-users' needs and contextual information can be distributed.

This includes introducing mechanisms which automatically distribute information, e.g., in order to present the end-user's requirements to social community members. Traditionally, requirements are communicated and presented using text-based specifications [KS98]. However, web-based technologies provide new ways to communicate and visualize gathered information. We have prototyped a visualization solution which uses Google Maps [Go10] to present end-users' needs. This solution specifically helps highlighting contextual information gathered with the help of the iRequire tool.

Figure 2 shows how the current prototype visualizes an end-user's requirement and related contextual information. In particular, the balloon highlights the originator of the need and presents a description (e.g. *I would like to invite friends for a coffee at the push of a button*). Furthermore, social community members can access more detailed information on the need, which includes the rationale. Using Google Maps also allows the presentation of contextual information in an intuitive and well-known way. This includes displaying the gathered picture of the environment next to the needs description. Furthermore, using Google Map's key feature allows the presentation of the captured location (e.g. the location of Lisa's favourite café). The example also highlights how such an approach could be used to start requirements negotiation. As shown in Figure 2, social network users can rate the visualized need.



Figure 2: Requirements Visualization with Google Maps

## 4.2 How End-User Needs Can Inform RE Activities

Requirements elicitation typically begins with an informal and incomplete mission statement which can be represented by using high-level goals, functions and constraints for the planned software-intensive system [KS98]. These initial requirements are the basis for further refinements elaborated in an iterative and incremental manner. Individual end-users' needs can be seen as a starting point for the development of Mobile Social Software. However, consecutive discussions and negotiations are needed to agree on fully specified requirements.

End-users' needs gathered with iRequire could, for example, be used to inform state-of-the-art requirements elicitation approaches such as EasyWinWin [GS05]. Traditionally, EasyWinWin uses brainstorming to gather initial users' needs which are prioritized and negotiated in consecutive steps. A modified version of EasyWinWin could consider end-users' needs gathered with iRequire in addition to or replacing brainstormed needs. Contextual information gathered with iRequire could inform requirements negotiation by supporting stakeholders in better understanding initial needs. For example, pictures taken with iRequire enable stakeholders to see the environment in which a need emerged. Furthermore, automatically captured contextual information such as location information could inform system design. For example, a system implementing Lisa's requirement *to invite her friends at the push of a button*, needs to automatically detect Lisa's current position. Assuming that Lisa documented her need in situ, the automatically gathered contextual information could inform developers about the availability of GPS positioning data. If iRequire was not able to document Lisa's position while she was documenting her requirement, developers might need to consider other technologies than GPS to provide a system which is able to detect Lisa's position.

We foresee that the information gathered with iRequire could be used in web-platforms for social software requirements engineering such as the system presented by Lohmann et al. [Lo09]. Furthermore, an integration of negotiation functionality in social networks (e.g. Facebook) and the use of requirements visualisation features (e.g. based on Google Maps) would enable existing social communities to discuss the data gathered with iRequire to design Mobile Social Software tailored to their particular needs.

## 5 Conclusion and Future Work

Paradigms such as mobile and service-oriented computing provide new opportunities for social software, but also require tailored software engineering methods and tools. In this paper we presented preliminary answers to research questions focusing on end-user driven requirements elicitation for Mobile Social Software. The main contribution of our work is the iRequire tool, which enables future end-users to blog their individual needs and expectations regarding a future mobile social application in situ. Presenting iRequire also gives preliminary answers to research question one and research question two. The tool allows end-users to document individual needs on-site and provides automatic context sensing capabilities (e.g. capturing of GPS location information).

Research question three focuses on the distribution of gathered end-user needs and contextual information in order to inform system design activities. We presented a novel way how to visualize documented information using Google Maps and discussed strategies how RE activities could benefit from the gathered requirements and contextual information. To the best of our knowledge iRequire is the first mobile RE tool enabling end-users to document their needs in situ. Initial studies suggest that end-users of Mobile Social Software are able to document their needs with iRequire. However, future research is needed to investigate the benefits and limitations of end-user driven requirements elicitation.

***Improving iRequire's usability and utility.*** We plan to conduct advanced usability and utility studies to explore end-user driven requirements elicitation in more detail. Based on the results of these studies we will enhance the iRequire prototype and provide guidance and support on how to use iRequire in mobile software engineering projects.

***Immediate distribution of needs.*** The current iRequire prototype stores captured information locally on the mobile device. We plan to provide an advanced version of iRequire, enabling the distribution of upcoming needs via mobile networks. This feature will enable end-users to immediately inform interested parties (e.g. social community members and requirements engineers) about upcoming needs. However, this research also needs to address privacy issues that are raised by the distribution of needs.

***Improved capturing of contextual information.*** iRequire's automatic context capturing capabilities are currently limited to location and time. We will provide extended automated context sensing capabilities in the near future, which will allow iRequire to more precisely describe the environment in which a need was captured. For example, we intend to make the tool capable of sensing Bluetooth signals of other mobile devices in order to detect the proximity of nearby mobile social software users.

***Informing system design with contextual information.*** In our current research we have shown that contextual information enables us to visualize requirements in a novel way. Furthermore, we argue that contextual information supports stakeholders in understanding end-user needs in which situation a need emerged. However, more sophisticated studies are needed to investigate the relation between end-user needs and context and the correlation between contextual information and future system design activities. We envision that contextual information could be in particular relevant for the design and development of context-aware and context-adaptive mobile social software.

***Investigating RE in social networks.*** In this paper we have started to discuss how requirements and contextual information gathered with iRequire could inform RE activities. However, we envision that individual end-user needs could be reviewed and discussed in existing social networks (e.g. Facebook). The integration of RE features in social networks would enable social communities to discuss needs and expectation in a familiar environment. Our vision includes integrating requirements prioritization and requirements visualization (e.g. using Google Maps) in social networks. Using such features, social community members would be able to more easily contribute to the design of social software.



## References

- [At06] Arnall T.: Mobile social software applications - A growing list of social applications that work in a mobile context. [Online] Available: <http://www.elasticspace.com/2004/06/mobile-social-software> [Accessed] Jan 10, 2010.
- [BBG03] Blomberg, J., Burrell, M. and Guest, G.: An Ethnographic Approach to Design. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, Jacko, J.A., Sears, A., Ed., Lawrence Erlbaum Associates, Mahwah, 2003, pp. 964-986.
- [CHS06] Counts S., Hofte H. ter, Smith I.: Mobile social software: realizing potential, managing risks. In *Proc. Conference on Human Factors in Computing Systems (CHI)*, Montreal 2006, pp. 1703-1706.
- [Go10] Google: Google Maps API [Online] Available: <http://code.google.com/apis/maps/> [Accessed] Jan 10, 2010.
- [GTR06] Granebring A., Thilénius P., Révay P.: SOA is a driver for Personalization. In *Proc. 13th European Conference on Information Technology Evaluation*, Genua 2006, pp. 250-251.
- [GS05] Grünbacher, P., Seyff, N.: Requirements Negotiation. In *Engineering and Managing Software Requirements*, Aurum, A., Wohlin, C., Ed., Springer, 2005, pp. 143-162.
- [HB08] Hammouda I., Bosch J., Jazayeri M., Mikkonen T.: Home page – The 1st International Workshop on Social Software Engineering and Applications. [Online] Available: <http://www.cs.tut.fi/sosea08> [Accessed] Jan 10, 2010.
- [Ho06] Hodges S. et. Al.: SenseCam: A Retrospective Memory Aid. In *Proc. 8<sup>th</sup> International Conference on Ubiquitous Computing*, California 2006, pp. 177-193.
- [KJR07] Kolko, B. E., Johnson, E., Rose, E.: *Mobile Social Software for the Developing World. In Communities and Social Computing*, Springer Berlin / Heidelberg, 2007, pp. 385-394.
- [KS98] Kotonya G., Sommerville, I.: *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, Chichester, 1998.
- [Lo09] Lohmann, S et. al.: A Web Platform for Social Requirements Engineering. In *Proc. International Workshop on Software Engineering within Social Software Environments*, Kaiserslautern, 2009.
- [Ma04] Maiden, N.: Systematic Scenario Walkthroughs with ART-SCENE. In *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*, Alexander, I., Maiden, N., Ed., John Wiley & Sons, 2004, pp. 161-178.
- [NS08] Nurcan S., Schmidt R.: Home-page – The First Workshop on Business Process Management and Social Software. [Online] Available: <http://crinfo.univparis1.fr/users/nurcan/BPMS2/> [Accessed] Jan 10, 2010.
- [Or06] O'Reilly T.: Web 2.0 Compact Definition: Trying Again. O'Reilly radar, Dec. 10, 2006. [Online] Available: <http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html> [Accessed] Jan 10, 2010.
- [RR06] Robertson S. Robertson J.: *Mastering the Requirements Process*, Second Edition. Addison-Wesley Professional, 2006.
- [SFS05] Sutcliffe, A., Fickas, S., Sohlberg, M.M.: Personal and Contextual Requirements Engineering. In *Proc. 13th IEEE International Conference on Requirements Engineering*, Paris, 2005, pp 19-28.
- [Se09] Seyff N. et. al.: Scenarios in the Wild: Experiences with a Contextual Requirements Discovery Method. In *Proc. Requirements Engineering: Foundation for Software Quality*, Amsterdam, 2009 pp.147-161.
- [Sm05] Smith I.: Social-Mobile Applications. *IEEE Computer* 38(4), 2005, pp. 84-85.
- [Se09] Seyff N. et al.: Exploring how to use scenarios to discover requirements. In *Requirements Engineering* 14 (2), 2009, pp. 91-111.
- [Tm03] Tepper, M.: The rise of social software. *netWorker magazine* 7(3), 2003, pp. 18-23.

SE | 10  
SOFTWARE ENGINEERING

**Doktorandensymposium**



# Doktorandensymposium der SE 2010

Alexander Pretschner

Fraunhofer IESE und TU Kaiserslautern  
67663 Kaiserslautern  
pretschner@cs.uni-kl.de

Ziel des Doktorandensymposiums war es, jungen Wissenschaftlerinnen und Wissenschaftlern konstruktive Rückkopplung zu ihren Dissertationsvorhaben durch erfahrene Forscher und anderen Doktoranden außerhalb ihrer Forschungsgruppe zu geben. Durch sorgfältig angefertigte Gutachten wurde sichergestellt, dass die Doktoranden unabhängig davon, ob ihre Einreichungen akzeptiert wurden, von ihrer Einreichung zum Symposium profitieren konnten.

Im Gegensatz zu Beiträgen für Workshops oder für die Konferenz sollten ganz bewusst nicht einzelne technische Resultate präsentiert und diskutiert werden, sondern vielmehr das – auch unfertige – Gesamtvorhaben mit Problemstellung, (erwarteter) Lösung und (erwartetem) Beitrag.

Von dreizehn eingereichten Beiträgen wurden sechs angenommen und in jeweils halbstündigen Vorträgen vorgestellt. Die Einreichungen stammten aus allen Gebieten des Software Engineering mit einer leichten thematischen Konzentration auf modellbasierte Ansätze. Akzeptiert wurden die Beschreibungen folgender Vorhaben:

Ekaterina Boutkova: Variantenmanagement in Anforderungsdokumenten  
Henning Groenda: Product Certification of Component Performance Specifications  
Patrick Könemann: Design Decisions in Model-driven Software Development  
Holger Röder: Ein Ansatz zur Entwicklung operationalisierter Usability-Anforderungen  
Konrad Voigt: Semi-automatic Matching of Heterogeneous Model-based Specifications  
Claus Wonnemann: Towards Information Flow Auditing in Workflows

Im Anschluss an jeden Vortrag fand eine etwa viertelstündige ausnahmslos sehr konstruktiv und engagiert geführte Diskussion statt, in der neben technischen Aspekten insbesondere das Gesamtvorhaben der Dissertation bzgl. wissenschaftlicher Methodik, Abgrenzung zu vergleichbaren Ansätzen und Umfang durchleuchtet wurde. Wertvolle Hinweise wurden dabei nicht nur durch die anwesenden Hochschullehrer, sondern auch durch andere Doktoranden gegeben. Als Abschluss der Veranstaltung präsentierten die anwesenden Professoren kurz ihre – durchaus heterogenen – Vorstellungen dessen, was für sie eine gute Dissertation auszeichnet.

Die ausnahmslos sehr positive Rückkopplung durch die Doktoranden nach der Veranstaltung legt nahe, dass die genannten Ziele des Symposiums vollumfänglich erreicht worden sind.



# Variantenmanagement in Anforderungsdokumenten

Ekaterina Boutkova

Group Research and Advanced Engineering  
Daimler AG  
Postfach 2360  
89013 Ulm

[ekaterina.boutkova@daimler.com](mailto:ekaterina.boutkova@daimler.com)

**Abstract:** Effektive Methoden für Variantenmanagement in Anforderungsdokumenten sind die essentielle Voraussetzung für eine effektive und effiziente Wiederverwendung von Anforderungsdokumenten. Dies führt zu einer Entwicklungszeitreduktion und somit zu einem Wettbewerbsvorteil insbesondere in der Automobilindustrie. Als Ausgangspunkt der Dissertation wurde die State of the Practice im Bereich der existierenden Ansätze zur Variantendokumentation in Lastenheften bei der Daimler AG erhoben und analysiert. Die Analyse hat gezeigt, dass die existierenden Methoden nicht alle Anforderungen der Entwickler der Automotive-Domäne erfüllen und ein großes Verbesserungspotenzial zur Qualitätssteigerung der Lastenhefte im Bezug auf Variantenmanagement aufweisen. Die Literaturrecherche hat ergeben, dass eine Menge unterschiedlicher Ansätze für Variantenmanagement existiert, allerdings wurde kein Ansatz gefunden, der sich explizit mit dem Thema der Zuordnung von natürlichsprachlich-definierten Anforderungen zu Varianten beschäftigt. Das Ziel meiner Dissertation ist, einen Ansatz für effektive und effiziente Zuordnung von einzelnen Anforderungen zu Varianten in Anforderungsdokumenten zu entwickeln, der die Anforderungen der Automobilindustrie erfüllt. Dieser Beitrag stellt die aktuelle Situation in der PKW-Entwicklung der Daimler AG, sowie die Leitgedanken meiner Dissertation vor.

## 1 Motivation

Die Automobilindustrie ist durch eine große Anzahl an Produktvarianten geprägt. Die Gründe für die Produktvielfalt sind vielseitig. Als prominentes Beispiel können im Falle der Daimler AG die Modellreihen (z.B. A-Klasse, B-Klasse oder S-Klasse) betrachtet

werden. Dabei werden für jedes Modell mehrere Ausführungsarten und Ausstattungsvarianten angeboten.

Die Reduktion der Entwicklungszeit für neue Fahrzeuge ist der Leitgedanke der letzten Jahrzehnte in der Automobilindustrie. Um die Gesamtentwicklungszeit reduzieren zu können, müssen alle Entwicklungsschritte so kurz wie möglich gehalten werden ohne die Qualität zu vernachlässigen. Eine Möglichkeit der Reduzierung der Entwicklungszeit ist die Wiederverwendung. Die Anforderungsdokumente in der Automobilindustrie bieten großes Wiederverwendungspotential.

Das Ziel meiner Dissertation ist die Vereinfachung und Verbesserung des Variantenmanagements in Anforderungsdokumenten. In diesem Beitrag wird zuerst die Begrifflichkeit geklärt (Kapitel 2). Danach wird ein Überblick über die aktuelle Landschaft von Lastenheften bei der Daimler AG (Kapitel 3.1) sowie die existierenden Variantendokumentationsansätze (Kapitel 3.2) gegeben. Im Kapitel 4 werden die identifizierten Defizite der existierenden Ansätze vorgestellt. Kapitel 5 bietet einen tieferen Einblick in das Thema der Dissertation sowie in den aktuellen Stand der Arbeit

## 2 Definitionen

Die erste Aktivität bei der Analyse der existierenden Ansätze zur Dokumentation von Varianten in Lastenheften war die Festlegung der Begriffe im Bereich der Variabilität. Die in der Literatur gefundenen Begriffe [CE00], [CN01], [PB05], [Re08] wurden in Zusammenarbeit mit Entwicklern aus unterschiedlichen Bereichen diskutiert und an die Spezifika und das Entwicklungsumfeld der Daimler AG angepasst. Basierend auf den Ergebnissen dieser Zusammenarbeit wurden folgende Begriffe definiert:

- *Variabilitätskriterium (VK)* ist eine Eigenschaft eines Bezugsgegenstandes die variieren kann.
- *Ausprägung eines Variabilitätskriteriums (AVK)* ist ein zulässiger Wert des Variabilitätskriteriums.
- Die *Variante* ist ein Objekt, dessen Eigenschaften eine Kombination der Ausprägungen der Variabilitätskriterien eines Bezugsgegenstandes darstellen. Dabei muss sich dieses Objekt von allen anderen Objekten in mindestens einem Merkmal unterscheiden.
- Eine *zulässige Variante* besteht aus einer zulässigen Kombination der Ausprägungen der Variabilitätskriterien für einen Bezugsgegenstand.
- Die *Variabilität* eines Bezugsgegenstandes beschreibt die Menge alle zulässigen Varianten dieses Bezugsgegenstandes.

Die umfassende Vorstellung der Begriffe inklusive Beispiele können in [Bo09] nachgeschlagen werden.

### 3 State of the Practice

Dieses Kapitel stellt die aktuelle Welt der Anforderungsdokumente bei der Daimler AG sowie einen Überblick der existierenden Ansätze für Variantenmanagement in Lastenheften dar.

#### 3.1 Anforderungsdokumente bei der Daimler AG

Für die Dokumentation der Anforderungen bei der Daimler AG existieren vor allen drei Typen von Lastenheften: das Fahrzeuglastenheft (FLH), das Systemlastenheft (SLH) und das Komponentenlastenheft (KLH). Das FLH enthält umfangreiche Beschreibung des neuen Fahrzeugs. Das SLH stellt die interne technische Spezifikation für komponentenübergreifende Systeme dar. Das KLH stellt die zentrale technische Spezifikation für Serienvergaben dar. Es bildet die Basis für die Anfrage an Lieferanten und für die Angebotserstellung durch die Lieferanten. Die Beziehungen zwischen einzelnen Lastenhefttypen sind in Abbildung 1 präsentiert. Alle Lastenhefte bei der Daimler AG werden in natürlicher Sprache spezifiziert.

Für die Verwaltung von SLH und KLH wird das Tool DOORS von IBM/Telelogic verwendet. Ein SLH bzw. KLH besteht aus einem bzw. mehreren DOORS-Modulen, die jeweils mehrere tausend DOORS-Objekte beinhalten können.

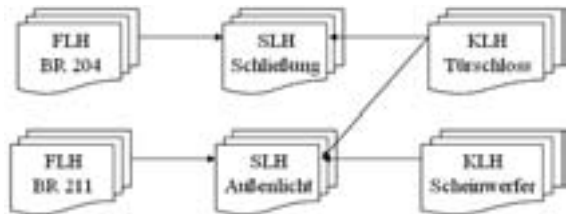


Abbildung 1: Beziehungen zwischen FLH, SLH und KLH

Da die Systeme und Komponenten für die verschiedenen Baureihen (BR) oft nur geringe Unterschiede aufweisen existiert ein großes Wiederverwendungspotenzial für Anforderungen. Diese Erkenntnis erlaubt die Varianten eines Systems bzw. einer Komponente im gleichen Lastenheft zu spezifizieren. Wichtig ist, dass alle Anforderungen eines Lastenheftes zu den jeweiligen Baureihen und Varianten zugeordnet sein müssen, um die Ausleitung von baureihenspezifischen Lastenheften für eine System- bzw. Komponentenvariante zu ermöglichen. Diese Lastenhefte werden für die Ausschreibungen benötigt, da die Lieferanten eine Spezifikation für eine konkrete Komponentenvariante, die für eine bestimmte Baureihe festgelegt wurde, benötigen. Wegen des Umfangs der Lastenhefte ist die Zuordnung von Anforderungen zu Varianten ein sehr aufwendiger Prozess.

#### 3.2 Existierende Ansätze

Im Rahmen der Analyse der existierenden Ansätze wurde eine Befragung der Entwickler aus den Bereichen „Mechanische Systeme“ und „Elektrisch/Elektronische-Systeme“ durchgeführt und eine Auswahl existierender KLH aus diesen Bereichen analysiert. Zuerst wurde die Begrifflichkeit geklärt (siehe Kapitel 2), danach wurden die einzelnen



Ansätze analysiert und bewertet. Anschließend wurden die Anforderungen an einen Variantendokumentationsansatz erhoben (siehe Kapitel 4).

In diesem Beitrag werden die Ansätze aufgelistet und deren Vor- und Nachteile kurz vorgestellt. Die vollständige Studie zu dem State of the Practice kann aus der Publikation [Bo09] entnommen werden. In Rahmen der Studie wurden folgende Ansätze identifiziert, analysiert und bewertet:

1. Variantendokumentation im Anforderungstext
2. Variantendokumentation in Attributsspalten (mit/ohne Common)
3. Variantendokumentation mit Multi-Enumeration-Attribut (mit/ohne Common)
4. Variantendokumentation durch Kapitelstruktur
5. Variantendokumentation in unterschiedlichen Doors-Modulen
6. Variantendokumentation im Anforderungspool
7. Code-Regel-Ansatz

Zu den Vorteilen dieser Ansätze gehören die einfache Handhabung (1-4), die transparente Zuordnung von Anforderungen zu den Varianten und die Verbindung mit Stücklisten (7). Die Nachteile sind hoher Aufwand und hohe Fehlerwahrscheinlichkeit durch manuelle Zuordnung sowie die fehlende Möglichkeit diese Zuordnung zu überprüfen. Außerdem skalieren diese Ansätze schlecht. Bei einer großen Anzahl von Varianten geht der Überblick schnell verloren. Zusätzlicher Nachteil des Code-Regel-Ansatzes ist ein nicht intuitives Verständnis der verwendeten Codes.

## 4 Problembeschreibung

Bei der Analyse der existierenden Ansätze wurden mehrere Defizite entdeckt. Ein Problem ist der große Aufwand für die Zuordnung von Anforderungen zu Varianten. Ein Lastenheft bei der Daimler AG kann aus mehreren Tausend Anforderungen bestehen, die bis zu 100 Varianten zugeordnet werden müssen. Beim Einfügen einer neuen Variante müssen alle Anforderungen auf die Gültigkeit für diese Variante überprüft werden. Bei der Verwendung des Common-Ansatzes müssen zusätzlich alle schon zugeordneten Anforderungen mit dem Common-Wert überprüft werden. Falls neue Anforderungen im Zusammenhang mit der neuen Variante hinzugefügt werden, muss überprüft werden, ob diese Anforderungen auch für die bereits existierenden Varianten gültig sind. Zur Überprüfung der Zuordnung von Anforderungen zu Varianten können zurzeit nur Reviews verwendet werden.

In Kooperation mit den Entwicklern der Daimler AG wurden Anforderungen an einen Variantenmanagementansatz in Lastenheften identifiziert:

- Skalierbarkeit des Ansatzes: Es ist wichtig, dass ein Ansatz sowohl für die Dokumentation einer kleinen Anzahl an Varianten als auch einer größeren Anzahl an Varianten geeignet ist.

- Gleiche Anforderungen dürfen nur einmal dokumentiert werden um die Größe von jetzt schon umfangreichen Lastenheften nicht zu steigern.
- Die Zuordnung von Anforderungen zu Varianten bzw. BR muss erkennbar und vergleichbar sein. Es muss möglich sein, Lastenhefte mehrerer Varianten zu vergleichen.
- Die Anforderungen, die für alle Varianten gültig sind müssen erkennbar sein.
- Die Beziehungen zwischen einzelnen VK bzw. AVK müssen prüfbar sein, um die Konsistenz der Varianten überprüfen zu können.
- Die Zuordnung von Anforderungen zu Varianten muss automatisch überprüfbar sein, um die Qualität der Lastenhefte zu garantieren.

Im nächsten Kapitel das Thema meiner Dissertation und die ersten Ergebnisse meiner Arbeit vorgestellt werden.

## 5 Dissertationsthema

Parallel zur der Analyse der existierenden Ansätze wurde eine umfangreiche Literaturrecherche durchgeführt, die in diesem Beitrag nicht vorgestellt wird. Basierend auf der Ergebnisse der Literaturrecherche und State of the Practice wurden zwei Arbeitsthemen identifiziert. Das erste Thema beschäftigt sich mit der Identifikation der Variabilität, Variabilitätsmodellierung und Auswahl der zulässigen Varianten für die Systeme und Komponenten. Das zweite Thema beschäftigt sich mit der Zuordnung von Anforderungen zu Varianten in Lastenheften. Die beiden Themen haben gegenseitige Wechselwirkungen.

Für den Bereich des Variabilitätsmanagements von Systemen und Komponenten können die Ansätze aus der Literaturrecherche (z.B. [BF01], [KC90], [LM03]) übernommen werden. Ein essentieller Teil des Variantenmanagements ist die Variabilitätsmodellierung. Aus der großen Menge der in Literatur beschriebenen Variabilitätsmodellierungsansätze können zwei Hauptgruppen identifiziert werden: Merkmalsmodellierung (Feature Modeling) und Erweiterung von Modellierungssprachen (z.B. UML).

Die Merkmalsmodellierung hat große Akzeptanz in der Praxis bekommen [LM03]). Auch bei der Daimler AG wird dieser Ansatz in der Design- und Implementierungsphase der Softwareentwicklung verwendet. Die Merkmale stellen eine Abstraktion von Anforderungen dar. Deshalb könnte die Merkmalsmodellierung helfen die beiden Themen der Variabilität zu verbinden.

Bei der Literaturrecherche wurde kein Ansatz gefunden, der sich mit der Zuordnung von Anforderungen zu Varianten beschäftigt. Basierend auf identifizierten Problemen und Anforderungen der Entwickler sowie den Ergebnissen der Literaturrecherche wurde das Thema des *Variabilitätsmanagements in Anforderungsdokumenten* für meine Dissertation ausgewählt. Das Ziel meiner Arbeit ist einen Ansatz auszuarbeiten, der basierend auf den Merkmalsmodellierungsansatz für die Produkte, die Zuordnung von Anforderungen

zu einzelnen Varianten unterstützt und eine halb-automatische Unterstützung für die Überprüfung der Zuordnung bietet.

Eine der ersten Erkenntnisse auf diesem Gebiet war, dass im Falle der Daimler AG ein Paradigmenwechsel stattfinden muss: Von der baureihenorientierten Entwicklung hin zu einer produktlinienorientierten Entwicklung. Eine BR ist in den meisten Fällen kein VK, weil BR keine Eigenschaft einer Komponente ist, die variiert. Die BR kann die Begründung sein, warum eine Eigenschaft variiert. Eine mögliche Vorgehensweise ist die Produktvariabilität einer Komponente bzw. eines Systems und BR-Variabilität zu trennen. Dieser Vorgehensweise (Product-Set-Ansatz) wurde von M.Weber und M.O. Reiser [WR08] vorgeschlagen. Durch Verfolgung des Ansatzes zur Trennung der Variabilität wurde eine weitere Definition notwendig: Eine *Produktvariante* ist eine zulässige Variante, die für die Produktion ausgewählt wurde. Die Produktvariante kann dann eine oder mehreren BR zugeordnet werden.

Die Basis meines Ansatzes ist die Strukturierung eines Lastenhefts anhand eines Variabilitätsmodells. Der ausgedachte Prozess des Variantenmanagements in KLH besteht aus folgenden Schritten:

- Im ersten Schritt werden die VK und AVK eines Bezugsgegenstandes identifiziert. Danach wird das Variabilitätsmodell erstellt.
- Im zweiten Schritt werden die zulässigen Varianten anhand des Variabilitätsmodells bestimmt.
- Danach können die Kriterien der Product Sets bestimmt werden und die ausgewählten Produkte können zu Product Sets zugeordnet werden
- Parallel mit dem Schritt 3 kann die Struktur eines Domänen-Lastenhefts für eine Komponente angelegt werden. Die VK und AVK stellen die Knoten in der Struktur eines Lastenheftes dar. Die variantenspezifischen Anforderungen werden zu den entsprechenden VK bzw. AVK zugeordnet. Für die Unterstützung der Entwickler werden mehrere Prüfverfahren (z.B. die Strukturprüfung, Zuordnungsprüfung) bereitgestellt.
- Nachdem Schritte 3 und 4 erfolgreich abgeschlossen sind, kann die Zuordnung von Anforderungen zu Produkten automatisch erfolgen.
- Nach der Zuordnung von Anforderungen zu Produkten, können die Anforderungen den Baureihen zugeordnet werden. Danach können die BR-Lastenhefte erstellt werden, die mit BR-spezifischen Anforderungen gefüllt werden. Abschließend können automatisch die produktspezifischen Lastenhefte für eine konkrete BR erstellt werden.
- Nach der Ausschreibung der Komponente und dem Abstimmungsprozess mit dem Lieferant wird überprüft, ob die BR-spezifischen Anforderungen in das Domänen-Lastenheft übernommen werden müssen.

Dieser Prozess wurde für das KLH der Sonnenblende angewendet. Dieses KLH wurde ausgewählt, weil es im Vergleich mit anderen KLH nicht so umfangreich ist. Es enthält circa 400 komponentenspezifische Anforderungen. Aktuell sind drei Produktvarianten geplant, die in den nächsten drei Baureihen eingebaut werden. Zurzeit wird das Variabilitätsmanagement mit Spalten-Ansatz realisiert. Beim Spalten-Ansatz muss der

Entwickler alle 400 Anforderungen manuell zu drei Baureihen zuordnen. Also er muss 1200-mal die Entscheidung treffen und richtige Attributwert eintragen. Mit dem neuen Ansatz muss der Entwickler die VK/AVK-Kapiteln zu Produkten zuordnen (circa 50 Entscheidungen).

Der beschriebene Ansatz bietet weitere Vorteile. Der Arbeitsaufwand für die Zuordnung der Anforderungen zu Varianten wird reduziert. Das halb-automatische Verfahren zur Überprüfung der Zuordnung ermöglicht die Steigerung der Qualität der Lastenhefte. Die strukturierte Vorgehensweise unterstützt die Wiederverwendung von Anforderungen. Außerdem ist der vorgeschlagene Ansatz unabhängig von Anforderungsdokumentationstools. Zu den Nachteilen dieses Ansatzes gehören der initiale Schulungsaufwand für Entwickler im Bereich Merkmalsmodellierung und der Aufwand für die Erstellung und Pflege von Variabilitätsmodellen, der aber durch geeignete Werkzeuge unterstützt werden kann.

## 6 Ausblick

In der ersten Anwendung wurde ein einfaches Beispiel (kleines Lastenheft mit weniger VK/AVK) verwendet. Im nächsten Schritt sind Pilot-Projekte mit Lastenheften aus verschiedenen Bereichen in der Größe, die für Lastenhefte der Daimler AG eher typisch sind (10.000 Anforderungen und 50 VK/AVK) geplant. Außerdem ist die Erweiterung des Ansatzes für SLH vorgesehen. Parallel zu diesen Arbeiten läuft die Entwicklung der Algorithmen für die Prüfverfahren.

## Literaturverzeichnis

- [BF01] Bosch, J.; Florijn, G.; Greefhorst, D.; Kuusela, J.; Obbink, H.; Pohl, K.: Variability Issues in Software Product Lines, Proceedings of PFE-4, 2001, pp. 11–19.
- [Bo09] Boutkova E: Variantenmanagement in Anforderungsdokumenten: State of the Practice. SEISCONF2009, München, 2009.
- [CE00] Czarnecki, K.; Eisenecker, U.: Generative Programming. Addison-Wesley, 2000.
- [CN01] Clements, P.; Northrop, L.: Software Product Lines: Practices and Patterns, SEI Series in Software Engineering. Addison-Wesley, 2001.
- [KC90] Kang, K.; Cohen, S.; Hess, J.; Novak, W.; Peterson, S.: Feature Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-021, 1990.
- [LM03] Lichter, H.; von der Maßen, T.; Nyßen, A.; Weiler, T.: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung. 2003.

- [PB05] Pohl, K.; Böckle, G.; van der Linden, F.: *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer Verlag, 2005.
- [Re08] Reiser, M.O.: *Managing Complex Variability in Automotive Software Product Lines with Subscoping and Conguration Links*. Dissertation, TU Berlin, 2008.
- [WR05] Reiser, M.O.; Weber, M.: *Using Product Sets to Define Complex Product Decisions*. In: H. Obbink and K. Pohl (Hrsg.): *SPLC 2005, LNCS 3714*, Springer-Verlag, pp. 21–32, 2005.

# Ein Ansatz zur Entwicklung operationalisierter Usability-Anforderungen

Holger Röder

Institut für Softwaretechnologie, Universität Stuttgart  
roeder@informatik.uni-stuttgart.de

**Abstract:** Die Rolle von Usability in der Praxis heutiger Software-Entwicklung ist nicht klar definiert: obwohl gute Usability als Ziel weithin akzeptiert ist, wird dieses Ziel nur selten systematisch verfolgt. Die Einbindung entsprechender Aktivitäten in existierende Entwicklungsprozesse und insbesondere die Berücksichtigung von Usability-Anforderungen erweisen sich als schwierig. In der Konsequenz hängt die ergonomische Gestaltung interaktiver Softwaresysteme häufig allein vom Willen und vom Geschick einzelner Beteiligter ab.

Der im Rahmen des vorgestellten Dissertationsvorhabens entwickelte Ansatz versucht, diesen Schwierigkeiten zu begegnen und Usability-Aspekte in der Anforderungsanalyse systematisch zu behandeln. Er sieht die Ergänzung einer funktionalen Spezifikation durch Interaktionsanforderungen vor, die durch Auswahl und Operationalisierung geeigneter Prinzipien der Interaktionsgestaltung entwickelt werden. Funktionale Interaktionselemente (z. B. einzelne Schritte in einem Use Case) werden dazu auf ein Modell der Systeminteraktion abgebildet. Aus einem Katalog von Interaktionsmustern werden anhand dieser Abbildung die Gestaltungsprinzipien ausgewählt, die auf die spezifischen Interaktionselemente zutreffen und im gegebenen Nutzungskontext anwendbar sind. Die Anwendung der Interaktionsmuster wird schließlich in Form von Interaktionsanforderungen spezifiziert. Diese Anforderungen sind umsetzbar und prüfbar; sie ermöglichen die Operationalisierung von Usability-Vorgaben und erleichtern die Berücksichtigung in Architekturentwurf, Implementierung und Test.

## 1 Einleitung und Motivation

Gute Usability ist als Ziel bei der Entwicklung interaktiver Softwaresysteme weithin akzeptiert. Dennoch sehen die Prozessmodelle für die Software-Entwicklung kaum Aktivitäten vor, die sich explizit der ergonomischen Gestaltung der Software widmen; existierende Ansätze (z. B. [Met05]) zur Integration von Usability Engineering und Software Engineering sind kaum etabliert. In der täglichen Praxis wird das Ziel guter Usability nur selten systematisch verfolgt [MVSC05]. Stattdessen wird darauf vertraut, dass die Beteiligten intuitiv zu einer ergonomischen Lösung kommen – ein Vorgehen mit ungewissem Ausgang, das der Bedeutung der Usability kaum gerecht wird. Dies gilt insbesondere für den Bereich der Anforderungsanalyse [HKRM09].

Lehrbücher propagieren die Spezifikation quantifizierter Usability-Anforderungen auf Grundlage eines Qualitätsmodells, das messbare Teilqualitäten wie Benutzungseffizienz

oder Benutzerzufriedenheit vereint. Usability-Anforderungen dieser Art sind schwer zu erheben, da angemessene Zielwerte unbekannt sind; sie nicht operationalisierbar, ihre Umsetzung bleibt unklar; und schließlich sind sie nur mit großem Aufwand überprüfbar. In der Praxis sind quantifizierte Usability-Anforderungen nur selten anzutreffen [Bre09].

Rückt man von der Maximalforderung quantifizierter Usability-Anforderungen ab, können als Usability-Anforderungen auch qualitative Anforderungen betrachtet werden, denen der Versuch einer Operationalisierung von Usability zugrunde liegt. Hierzu zählen etwa UI-Styleguides, die sich auf die *physische* Gestaltung der Benutzungsschnittstelle beziehen. Prototypen der Benutzungsschnittstelle dienen ebenfalls als Vorgabe für die weitere Entwicklung. Sie zeigen allerdings in der Regel nur einen kleinen Ausschnitt der Benutzungsschnittstelle eines Systems. Zudem können Aspekte der *logischen* Interaktionsgestaltung nicht immer explizit dargestellt werden, etwa die Fehlertoleranz bei Benutzereingaben. Auch für diese Aspekte lassen sich jedoch operationalisierte Usability-Anforderungen formulieren: *Interaktionsanforderungen* spezifizieren Vorgaben, die ausdrücklich die ergonomische Gestaltung des Systems zum Ziel haben. Sie sind häufig funktional formuliert, etwa wenn sie die Plausibilitätsprüfung von Eingaben spezifizieren; sie beziehen sich dabei jedoch nicht auf die fachliche Funktionalität des Systems an sich, sondern darauf, wie diese Fachlichkeit möglichst benutzerfreundlich verfügbar gemacht werden kann.

Existierende Methoden sehen eine Entwicklung von Interaktionsanforderungen, z. B. als Ergänzung zu UI-Prototypen, nicht vor. Im Rahmen des Dissertationsvorhabens soll untersucht werden, wie Interaktionsanforderungen mit Hilfe des INSPIRE<sup>1</sup>-Ansatzes systematisch entwickelt werden können.

## 2 Problemstellung und Lösungskonzept

Zunächst soll der verfolgte Ansatz anhand eines Beispiels eingeführt und veranschaulicht werden. Betrachtet wird die Entwicklung eines Web-basierten Konferenzmanagementsystems, das u. a. die Einreichung und Begutachtung von Beiträgen für das Tagungsprogramm unterstützen soll. Die Benutzer (z. B. Mitglieder des Programmkomitees) arbeiten nur sporadisch mit dem System. Gute Usability ist ein wichtiges Ziel.

Die geforderte Funktionalität des Systems ist in Form von Use Cases spezifiziert. Der linke Teil von Abbildung 1 zeigt den Use Case GUTACHTEN ABGEBEN, der beschreibt, wie Gutachter ihre Gutachten für eingereichte Tagungsbeiträge in das System eintragen können. Aus funktionaler Sicht ist der Use Case vollständig: er beschreibt alle Schritte der Interaktion zwischen Benutzer und System, die für die erfolgreiche Abgabe eines Gutachtens nötig sind. Usability-Vorgaben werden an separater Stelle in der Anforderungsspezifikation genannt; sie sind im rechten Teil von Abbildung 1 auszugsweise dargestellt. Eine solche Separierung von Funktionalität und Usability ist jedoch unzureichend: es ist nicht klar, wie die Anforderungen im konkreten Fall, also z. B. für einen einzelnen Use Case, auszulegen sind. Die Umsetzung der Use Cases bleibt somit ohne konkrete Vorgabe, eine Prüfung ist nicht möglich. In der Konsequenz ist die Berücksichtigung der Usability-Vorgaben in der Anforderungsspezifikation nicht sichergestellt.

---

<sup>1</sup>Das Akronym INSPIRE steht für *Integration and Specification of Interaction Requirements*.

## Funktionale Spezifikation mit Use Cases

Use Case	UC-7: Gutachten abgeben
Beschreibung	Der Gutachter gibt sein Gutachten (Kritik, Änderungsvorschläge, Empfehlung über Annahme oder Ablehnung) für einen eingereichten Beitrag ab.
Akteur	Gutachter
Normaler Ablauf	1. Der Gutachter wählt den Beitrag, für den er das Gutachten abgeben will, aus der Liste der eingereichten Beiträge aus.
	2. Das System zeigt die Details des eingereichten Beitrags an.
	3. Der Gutachter trägt sein Gutachten ein.
	4. Der Gutachter schickt sein Gutachten ab.
	5. Das System speichert das Gutachten und benachrichtigt das Programmkomitee per E-Mail über das neu eingegangene Gutachten.
	6. Das System zeigt die Liste der eingereichten Beiträge an.

## Separate Usability-Anforderungen

...

Das System soll auch für gelegentliche Benutzer ohne Schulung gut bedienbar sein.

Das System soll fehlerhafte Eingaben des Benutzers erkennen und korrigieren.

Der Benutzer muss den Zustand des Systems und die Resultate der eigenen Aktionen jederzeit erkennen können.

...

Abbildung 1: Use Case und separate Usability-Anforderungen

Um diesem Problem zu begegnen, sieht der INSPIRE-Ansatz vor, die funktionale Spezifikation um operationalisierte, auf einzelne Interaktionsbestandteile zugeschnittene Usability-Anforderungen zu ergänzen. Als Wissensbasis für die Entwicklung derartiger *Interaktionsanforderungen* wird ein Katalog ergonomischer Prinzipien und Interaktionsmuster verwendet. Diese *Interaktionsmerkmale* beschreiben in strukturierter Form bewährte Gestaltungsprinzipien. Das so verfügbar gemachte Wissen ermöglicht auch Entwicklern ohne Usability-Expertise die Entwicklung von Interaktionsanforderungen.

Abbildung 2 zeigt das prinzipielle Vorgehen bei der Entwicklung einer Interaktionsanforderung. Zunächst klassifiziert und attribuiert der Entwickler die vorhandenen *Interaktionselemente*, also z. B. einzelne Schritte oder Abläufe im Use Case. Im Beispiel ① klassifiziert er Schritt 4 des NORMALEN ABLAUFs als Interaktionsschritt. Unter Berücksichtigung von Semantik und Kontext des Interaktionsschritts weist er die Attribute EINGABE und NICHT REVERSIBEL zu, da ein Gutachter das Abschicken des Gutachtens nicht rückgängig machen kann. (Das System verschickt im nächsten Schritt E-Mails an das Programmkomitee, die nicht zurückgerufen werden können.)

Anschließend kann der Entwickler nun aus dem im rechten Teil von Abbildung 2 skizzierten Merkmalkatalog diejenigen Interaktionsmerkmale auswählen, die auf die einzelnen Interaktionselemente zutreffen, also anwendbar und aus ergonomischer Sicht empfehlenswert sind. Diese Anwendbarkeit ist für alle Interaktionsmerkmale definiert; sie stellt eine formale Beschreibung des Anwendungskontexts – in der Terminologie von Tidwell [Tid05] also des *use-when* – des jeweiligen Interaktionsmerkmals dar. Im Beispiel ② existiert im Merkmalkatalog das Interaktionsmerkmal RÜCKFRAGE, das sich auf NICHT REVERSIBLE EINGABEN bezieht und Benutzer daran hindert, versehentlich ungewollte Aktionen auszuführen, die sie nicht rückgängig machen können.

Die Anwendung eines Interaktionsmerkmals wird für jeden konkreten Fall in Form einer *Interaktionsanforderung* spezifiziert. Im Beispiel ③ soll das Interaktionsmerkmal RÜCKFRAGE auf Schritt 4 des Use Case angewendet werden; dies spezifiziert der Entwickler



## Funktionale Spezifikation mit Use Cases

Use Case	UC-7: Gutachten abgeben
Beschreibung	...
Akteur	Gutachter
Normaler Ablauf	...
	4. Der Gutachter schickt sein Gutachten ab.
	5. Das System speichert das Gutachten und benachrichtigt den Gutachter über die Mail.
	...

① Klassifizierung der Interaktionselemente

## Katalog ergonomischer Interaktionsmerkmale

Interaktionsmerkmal	Rückfrage
Beschreibung	Die Ausführung einer Aktion, die nicht rückgängig gemacht werden kann, muss vom Benutzer explizit bestätigt werden. ...
Nutzungskontext	[Laienbenutzer], [Gelegentliche Nutzung]
Relevante Interaktionselemente	Interaktionssequenz [nicht reversibel] Interaktionsschritt [Eingabe] [nicht reversibel]

② Identifizierung anwendbarer Interaktionsmerkmale, Auswahl aus dem Merkmalkatalog

③ Operationalisierung als Interaktionsanforderung, Verknüpfung mit Use-Case-Spezifikation

**Interaktionsanforderung IA-12:**  
4a. Das System weist den Gutachter darauf hin, dass die Bewertung nicht rückgängig gemacht werden kann, und fordert eine Bestätigung.  
4b. Der Gutachter bestätigt die Bewertung.

Abbildung 2: Entwicklung einer Interaktionsanforderung

durch die Interaktionsanforderung IA-12, die mit dem Use-Case-Schritt verknüpft wird. Die Interaktionsanforderung wird im Beispiel in Form von zwei zusätzlichen Interaktionsschritten formuliert, die in den Ablauf des Use Case eingefügt werden. (Alternativ können Interaktionsanforderungen z. B. auch als Präzisierung vorhandener Schritte, als zusätzliche Abläufe oder eigenständige Use Cases formuliert werden.) Die Interaktionsanforderung operationalisiert somit einen bestimmten Usability-Aspekt, der bei der Umsetzung der Funktionalität berücksichtigt werden soll; sie ist dabei selbst umsetzbar und überprüfbar.

### 3 Forschungsfragen und weitere Arbeiten

Bei der Weiterentwicklung des skizzierten Lösungsansatzes steht zunächst die Frage im Vordergrund, wie die Entwicklung von Interaktionsanforderungen methodisch und technologisch unterstützt werden kann. Hierzu sind verschiedene weitere Arbeiten notwendig.

Der Merkmalkatalog umfasst derzeit 34 Interaktionsmerkmale, die aus existierenden Arbeiten zu Interaktionsmustern (vgl. den Überblick in [DF06]) übernommen und angepasst oder durch Auswertung der Gestaltungsgrundsätze in der Norm ISO 9241-110 [ISO06] neu entwickelt wurden. Der Katalog soll im Rahmen der Arbeit um weitere Interaktionsmerkmale ergänzt werden; diese werden insbesondere bewährte Usability-Empfehlungen aufgreifen, die bisher noch nicht in strukturierter Form vorliegen.

Die Verknüpfung der einzelnen Interaktionselemente in der funktionalen Spezifikation mit jeweils anwendbaren Interaktionsmerkmalen aus dem Merkmalkatalog setzt ein einheit-

liches Modell voraus. Für INSPIRE wurde ein Modell der Systeminteraktion konzipiert, das vier Arten von Interaktionselementen unterscheidet, die Gegenstand der Interaktionsgestaltung sein können: das System (als Ganzes), Funktionen, Interaktionssequenzen und einzelne Interaktionsschritte. Die Interaktionselemente werden durch Angabe zusätzlicher semantischer Eigenschaften präzisiert. Das Modell ist prinzipiell nicht auf eine bestimmtes Format der funktionalen Anforderungsspezifikation beschränkt, solange diese die Interaktion zwischen Benutzern und dem System beschreibt. Wie im Beispiel gezeigt, bieten sich jedoch Use-Case-Spezifikationen für die Anwendung des Ansatzes an; bei der weiteren Arbeit werden sie deshalb im Fokus stehen. Durch Auswertung vorhandener Spezifikationen soll die Eignung des Modells für die Klassifizierung und Attributierung der einzelnen Interaktionselemente weiter evaluiert werden.

Weiterhin wird in der Arbeit ein Werkzeug entwickelt, das die Klassifizierung und Attributierung von Use Cases, die Auswahl anwendbarer Interaktionsmerkmale und die anschließenden Spezifikation von Interaktionsanforderungen ermöglicht. Besonderes Augenmerk wird dabei auf die Frage gelegt, wie Entwickler bei der semantischen Attributierung unterstützt werden können. Hier ist ein Expertensystem denkbar, das durch gezielte Fragen eine entsprechende Attributierung erlaubt („Kann der Benutzer den Interaktionsschritt rückgängig machen?“ – falls nein, wird die Eigenschaft NICHT REVERSIBEL zugewiesen).

Schließlich soll in der Arbeit untersucht werden, ob und in welcher Form Interaktionsanforderungen geeignet sind, die Entwicklung von Software mit guter Usability zu erleichtern. Hierbei ist besonders die Frage von Interesse, ob Entwickler mit INSPIRE bessere Resultate erzielen als mit vergleichbaren erfahrungsgestützten Usability-Ansätzen, etwa der Verwendung von Usability-Guidelines. Diese Frage soll empirisch beantwortet werden.

## 4 Verwandte Arbeiten

Das eingangs dargelegte erweiterte Verständnis von Usability-Anforderungen wird von anderen Autoren geteilt. So sind etwa Lauesen und Younessi [LY98] der Ansicht, dass bestimmte funktionale Anforderungen als Usability-Anforderungen zu betrachten sind, wenn sie sich vorrangig auf die ergonomische Gestaltung beziehen.

Zur Entwicklung von Usability-Anforderungen und zur Verwendung von Interaktionsmustern existiert eine Reihe verwandter Arbeiten. Die IESE NFR-Methode [DKK<sup>+</sup>05] etwa sieht die systematische Entwicklung nichtfunktionaler Anforderungen ausgehend von Use Cases vor. Bei diesem Ansatz bildet jedoch ein Qualitätsmodell die Grundlage für die Entwicklung; zudem steht die Formulierung quantitativer Anforderungen im Vordergrund. Schmettow [Sch05] beschreibt die gezielte Verwendung von Interaktionsmustern im Rahmen von Usability-Inspektionen und thematisiert u. a. die Frage, wie zutreffende Muster identifiziert werden können. Das Konzept eines wiederverwendbaren Katalogs von Usability-Merkmalen findet sich auch bei Cysneiros u. a. [CWK05] und ist dort Grundlage der Zielmodellierung. Dabei bleibt jedoch unklar, wie die Entscheidung für die Berücksichtigung bestimmter Usability-Produktmerkmale dokumentiert wird. Ein dem INSPIRE-Ansatz direkt vergleichbares Vorgehen zur Entwicklung spezifischer Usability-Anforderungen unter Verwendung von Interaktionsmustern ist nicht bekannt.

## 5 Zusammenfassung und Ausblick

In dieser Arbeit habe ich einen Ansatz zur Entwicklung von Interaktionsanforderungen vorgestellt, der sich auf bewährte Prinzipien ergonomischer Interaktionsgestaltung stützt und diese systematisch mit einer funktionalen Spezifikation verknüpft. Auf diese Weise soll dem Problem vager, pauschaler oder nicht vorhandener Usability-Anforderungen begegnet werden, das in vielen Projekten in der Praxis anzutreffen ist.

Die notwendigen weiteren Arbeiten am INSPIRE-Ansatz habe ich dargelegt; hier sind zunächst die Erweiterung des Merkmalkatalogs, die Evaluierung des Modells und die Entwicklung eines geeigneten Werkzeugs zu nennen. Diese Arbeiten schaffen die Voraussetzung für die anschließende praktische Evaluierung des Ansatzes.

Bei den Gutachtern bedanke ich mich herzlich für die konstruktiven und hilfreichen Anmerkungen zu meinem Dissertationsvorhaben.

## Literatur

- [Bre09] Manuel Breitfeld. Usability-Anforderungen in der Praxis. Diplomarbeit, Institut für Softwaretechnologie, Universität Stuttgart, 2009.
- [CWK05] Luiz Marcio Cysneiros, Vera Werneck und Andre Kushniruk. Reusable Knowledge for Satisficing Usability Requirements. In *Proc. 13th IEEE Int'l RE Conference*, 2005.
- [DF06] Andy Dearden und Janet Finlay. Pattern Languages in HCI: A Critical Review. *Human-Computer Interaction*, 21(1), 2006.
- [DKK<sup>+</sup>05] Jörg Dörr, Daniel Kerkow, Tom König, Thomas Olsson und Takeshi Suzuki. Non-Functional Requirements in Industry – Three Case Studies Adopting an Experience-based NFR Method. In *Proc. 13th IEEE Int'l RE Conference*, 2005.
- [HKRM09] Juho Heiskari, Marjo Kauppinen, Mikael Runonen und Tomi Männistö. Bridging the Gap Between Usability and Requirements Engineering. In *Proc. 17th IEEE Int'l RE Conference*, 2009.
- [ISO06] DIN EN ISO 9241 Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung, 2006.
- [LY98] Soren Lauesen und Houman Younessi. Six Styles for Usability Requirements. In *Proc. 4th Int'l Workshop on RE: Foundation for Software Quality*, 1998.
- [Met05] Eduard Metzker. *Adoption-Centric Usability Engineering*. Dissertation, Fakultät für Informatik, Universität Ulm, 2005.
- [MVSC05] Ji-Ye Mao, Karel Vredenburg, Paul W. Smith und Tom Carey. The State of User-Centered Design Practice. *Communications of the ACM*, 48(3), 2005.
- [Sch05] Martin Schmettow. Towards a Pattern Based Usability Inspection Method for Industrial Practitioners. In *Proc. INTERACT 2005*, 2005.
- [Tid05] Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly, 2005.

# Design Decisions in Model-driven Software Development

Patrick Könemann  
Informatics and Mathematical Modelling  
Technical University of Denmark  
2800 Kgs. Lyngby, Denmark  
pk@imm.dtu.dk

## Abstract:

Numerous design decisions are made in model-driven software development which are mostly implicit and not documented properly. Hence, the design knowledge is usually 'in the designers mind' and maybe propagated orally, if at all. There exist tools which tackle that problem for architectural decisions which refer to the higher level architecture of the system to develop, but these decisions are not linked to the design models and are only for documentation.

The goal of this project is to contribute concepts and a tool for explicit design decision support in model-driven software development. The contribution is twofold: a technology for creating and storing model differences is developed which will be used for storing model changes for reusable decisions; furthermore, an integration of decision management tools with modeling tools is presented. This allows explicit decision making with tool support and decision reuse which improves the design documentation process and allows sharing and reusing design knowledge.

## 1 Introduction

Software systems are developed on different levels of abstraction, as the Model-driven Architecture [OMG03] proposes, e.g. starting with a higher level architectural design, continuing with more detailed design models up to code which might be partially generated. Architectural knowledge can be stored and documented in terms of architectural decisions [CND07, TJH07, ZGK<sup>+</sup>07] which has two major advantages: the design is documented explicitly, and alternative solutions are considered and decisions are made consciously.

Developers also make decisions concerning the design of the system; examples for such decisions are the use of design patterns to solve particular design issues (e.g. from the Gang of Four [GHJV95]) or individual changes with consequences which should be made explicit (e.g. use of polymorphism or the movement of design elements). These decisions usually stay 'in the designers mind'. Such undocumented knowledge is also called tacit knowledge [HAZ07] and might be propagated orally or eventually gets lost.

We believe that it is also worth capturing and storing design decisions on the design-level and not only on the higher architectural level. That design knowledge can then be propagated along and linked to the models. Furthermore, developers benefit from recurring decisions as a form of best practices and patterns.

The complete idea of this project is described in Sect. 2 and consists of two major parts: Sect. 3 discusses how existing decision concepts from the architectural level can be used to support decision making in model-driven development (MDD); Sect. 4 introduces a technology for storing and reusing model differences (similar to a patch for code) for storing model changes resulted from design decisions. Sect. 5 concludes the paper.

## 2 The Idea

After defining the term *design decision*, this section discusses the requirements for introducing design decision support in MDD processes.

A design decision in terms of the system's architecture consists of a *design issue* or *problem*, several *constraints* and *assumptions*, one or many *solutions*, and a *rationale*, amongst others [SLK09]. Harrison et al. distinguish in [HAZ07] between *application-generic* and *application-specific* decisions, at which the former are reusable decisions and the latter are only documented for one particular project.

Our focus for design decisions is the design model: a design decision addresses a particular design problem (*issue*), considering one or many solutions (*alternatives*), and describes changes in the model according to the selected solution (*outcome*). This definition is based on the one in [Zim09], in particular concerning the distinction between application-generic parts (issues and alternatives) and application specific parts (outcomes).

Besides the problem that tacit knowledge vanishes eventually, developers mostly use their own well-known solutions instead of considering other, maybe better suited alternatives. An additional problem comes up when the same decision recurs (the same or very similar changes are made to the model), e.g. in similar projects: the same solution is probably realized over and over again which leads to redundant and error-prone work. Further problems are presented in more detail in [Kön09b] which all in all lead to the following goals for this project.

- Document the rationale and consequences for decisions explicitly.
- Link changes in the design model to the outcomes of decisions to make the design knowledge easily accessible at the design model and vice versa.
- Store design decisions in terms of best practices and patterns for reuse for other developers and similar projects.
- Support the developer in making design decisions and considering alternatives; focus on reusable and well-known decisions, e.g. from previous projects.

The idea for realizing these goals is the following. Tool support and several meta models for architectural design decisions already exist, however, they focus on the documentation of a project and rarely on reuse. Thus this project aims to reuse existing decision management tools and a technology is developed for storing design model changes and linking them to the decisions.

### 3 Design Decision Support in MDD

Some tools already exist for documenting and maintaining decisions, e.g. AREL [TJH07], ADDSS [CND07], and AdkWik [ZGK<sup>+</sup>07]. The latter two allow knowledge reuse in later projects but none of them links the decisions to the design models in the process. This project proposes an integration of such decision management tools with a modeling tool.

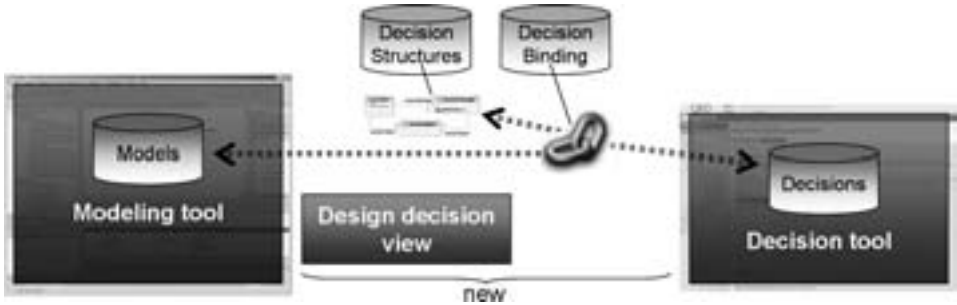


Figure 1: Integration of decision management with modeling tools

Figure 1 shows a modeling tool and a decision management tool at the left-hand and right-hand side, respectively. The center of the figure sketches our extensions for integrating both tools: a decision structure repository stores decision related application-generic design model changes; a decision binding repository contains application-specific bindings which link particular decisions to the respective design model elements; furthermore, an extension to the graphical user interface (design decision view) is added to the modeling tool which addresses the use cases listed in [Kön09b], for instance browsing through made design decisions. Formal meta models for the decision structure and binding as well as the tool interfaces are already defined.

In principle we support any modeling compliant to the Essential Meta Object Facility [OMG06] and any decision management tool which implements the respective interfaces. Our prototype, however, only supports the decision management tool AdkWik and any EMF-based<sup>1</sup> modeling tool so far.

To conclude, we introduce interfaces which the tools must implement – this makes the integration independent of the actual tools. All interfaces are already described in [Kön09a] and a prototype is under development in collaboration with the IBM Research Lab, Zurich.

#### Related Work

There are several decision management systems available but none of them connects the decisions to the actual models. ADDSS [CND07] and AREL [TJH07], for instance, are

<sup>1</sup>Eclipse Modeling Framework: <http://www.eclipse.org/modeling/emf/>

tools for the documentation of architectural decisions. Moreover, they allow sharing and analyses of the decisions, respectively. AdkWik [ZGK<sup>+</sup>07], which we use in our prototype, allows to capture, store, and share design knowledge as well; in addition, it supports the user in making and reusing decisions.

Some modeling tools like the IBM Rational Software Modeler<sup>2</sup> and Borland Together<sup>3</sup> provide pattern authoring capabilities similar to the intention of decision structures. However, a meta model for expressing relations between them as well as tool supported guidance is missing.

### 4 Model Differences

For this project, we require to calculate and store arbitrary model changes and apply them to another model as part of a reusable design decision (similar to how a patch is used for code). There are neither technologies nor concepts yet for storing self-contained model differences such that they are reusable, i.e. applicable to other models. This project builds on top of existing technologies for model differencing and provides additional concepts for self-contained model differences and strategies for their creation and application.

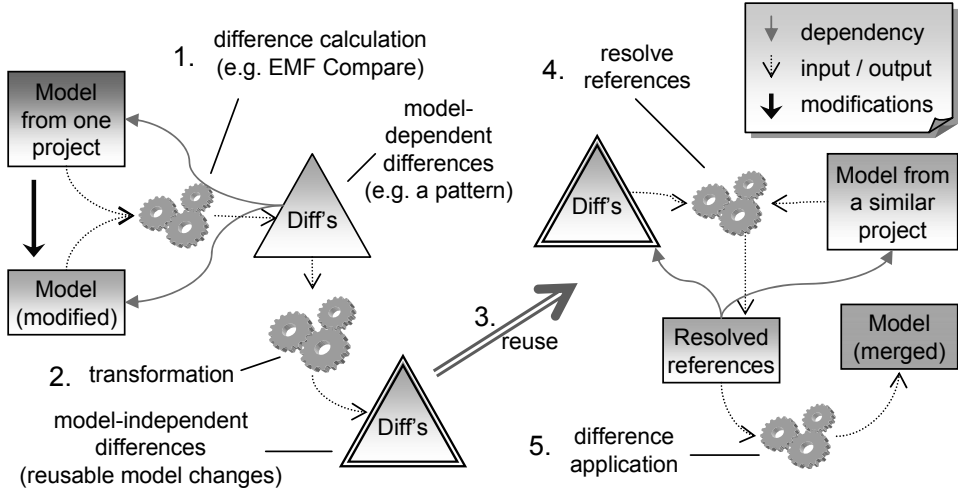


Figure 2: Creation and application process of model-independent differences

Figure 2 illustrates the use of so-called model-independent differences: changes made in a model from one project are extracted and applied to a model from a similar project (which must be capable of applying them). Examples for such changes are design patterns or other recurring changes like refactorings. The individual steps in Fig. 2 are as follows.

<sup>2</sup>Available at: <http://www.ibm.com/software/awdtools/modeler/swmodeler>  
<sup>3</sup>Available at: <http://www.borland.com/us/products/together/>

1. The *difference calculation* is performed with existing technologies, e.g. with EMF Compare<sup>4</sup>, and produces differences which are not applicable to other models.
2. A *transformation* creates self-contained model-independent differences; they are in particular valid without the original models (like a patch).
3. Now the differences can be stored for later reuse, e.g. in a database for best practices.
4. Before the differences can be applied to another model, a so-called *reference resolution* determines *whether* and *where* the changes could be applied.
5. If the previous step was successful, the *changes are applied* to the other model.

The resulting differences from step 2 are *model-independent* in terms of two meanings: first, the differences are self-contained, i.e. independent from the original model they were calculated from (unlike most other differencing technologies); second, the differences are also applicable to other models, e.g. from another project – thus, they are independent from the concrete contents of the original models (which is not supported by any other approach so far). The details are described in [Kön09c].

This technology will be used to store and reuse design model changes for reusable design decisions. A prototype is freely available at <http://modeldiff.imm.dtu.dk> and its contribution to the Eclipse project EMF Compare is in progress.

## Related Work

Even though none of the existing model differencing concepts have proper support for the creation of self-contained differences, some concepts and goals are very similar to ours. EMF Compare is a generic and extendable comparison framework for all EMF-based models including UML and many more domain specific languages. Unfortunately, the differences are not self-contained and in particular strictly bound to the original model. EPatch<sup>5</sup>, an extension for EMF Compare, is a language for describing self-contained patches; but unlike model-independent differences, epatches are not applicable anymore if the models have changed. AMOR [BLSW09] is an adaptable model versioning technique for concurrent versioning systems and defines refactorings as a set of model differences. However, they focus on self-adaptive difference detection and transfer within versioning systems and do not support generic difference creation and transfer.

## 5 Conclusion

This project's main contribution is to introduce explicit design decision support for MDD processes. To do so, we 'imported' already existing ideas and concepts from related work of architectural decision management and integrate that with modeling concepts and tools.

---

<sup>4</sup>Available at <http://www.eclipse.org/modeling/emf/?project=compare>

<sup>5</sup>[http://wiki.eclipse.org/EMF\\_Compare/EPatch](http://wiki.eclipse.org/EMF_Compare/EPatch)



To operate on models, the second major contribution is explicit and generic patching support for models. It includes a meta model for the definition and the process for the creation and application of self-contained model differences. That part is already finished and performs well for all tested modeling languages, UML amongst others.

A prototype for design decision support is in development, building on top of the framework for model differencing. The evaluation of the concepts and its realization is planned in an industrial project.

## References

- [BLSW09] Petra Brosch, Philip Langer, Martina Seidl, and Manuel Wimmer. *Towards end-user adaptable model versioning: The By-Example Operation Recorder*. ICSE Workshop on Comparison and Versioning of Software Models, pp. 55–60, 2009.
- [CND07] Rafael Capilla, Francisco Nava, and Juan C. Duenas. *Modeling and Documenting the Evolution of Architectural Design Decisions*. In proceedings of SHARK-ADI '07, p. 9, IEEE Computer Society, 2007.
- [GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Massachusetts, 1st edition edition, January 1995.
- [HAZ07] Neil B. Harrison, Paris Avgeriou, and Uwe Zdun. *Using Patterns to Capture Architectural Decisions*. IEEE Software, 24(4): pp. 38–45, 2007.
- [Kön09a] Patrick Könemann. *Integrating a design decision management system with a UML modeling tool*. IMM-Technical Report-2009-07, Technical University of Denmark, 2009.
- [Kön09b] Patrick Könemann. *Integrating Decision Management with UML Modeling Concepts and Tools*. In WICSA/ECSA, pp. 297–300. IEEE Computer Society, 2009.
- [Kön09c] Patrick Könemann. *Model-Independent Differences*. ICSE Workshop on Comparison and Versioning of Software Models, pp. 37–42, 2009.
- [OMG03] Object Management Group. *MDA Guide V1.0.1*. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>, June 2003.
- [OMG06] Object Management Group. *Meta Object Facility (MOF) Core Specification*. <http://www.omg.org/cgi-bin/doc?formal/2006-01-01>, January 2006.
- [SLK09] Mojtaba Shahin, Peng Liang, and Mohammad Reza Khayyambashi. *Architectural Design Decision: Existing Models and Tools*. In proceedings of WICSA, IEEE Computer Society, 2009.
- [TJH07] Antony Tang, Yan Jin, and Jun Han. *A Rationale-based Architecture Model for Design Traceability and Reasoning*. Journal of Systems and Software, 80(6): pp. 918–934, 2007.
- [ZGK<sup>+</sup>07] Olaf Zimmermann, Thomas Gschwind, Jochen M. Küster, Frank Leymann, and Nelly Schuster. *Reusable Architectural Decision Models for Enterprise Application Development*. In proceedings of QoSA, volume 4880 LNCS, pp. 15–32, Springer, 2007.
- [Zim09] Olaf Zimmermann. *An Architectural Decision Modeling Framework for Service-Oriented Architecture Design*. Dissertation, University of Stuttgart, 2009.

# Semi-automatic Matching of Heterogeneous Model-based Specifications

Konrad Voigt  
SAP Research CEC Dresden  
Chemnitzer Str. 48  
01187 Dresden, Germany  
konrad.voigt@sap.com \*

## Abstract:

IT-systems are often described by a variety of specifications such as UML, Java, BPMN, WSDL, etc. These heterogeneous specifications constitute different views on the same system, resulting in the challenge of matching. That is in context of Model Driven Engineering the discovery of semantic correspondences between model elements, which can be used for tasks such as transformation or trace link generation. Although support by semi-automatic matching has been proposed, current approaches show deficits. They leave room for improvement in matching quality, do not appropriately address scalability, and miss a thorough evaluation. They were proposed to specifically target differencing and versioning in contrast to matching.

We tackle these issues by proposing a configurable combination of matchers that considers both: meta-models and models. We propose to adopt established schema matching techniques and to utilize information gained from meta-model matching for the task of model matching. Additionally, the graph qualities planarity and reducibility are used to apply graph isomorphism and clustering algorithms for enhanced structural matching.

Finally, we build upon a generic model-based infrastructure (EMF) allowing for an easy integration of heterogeneous specifications and the realization of scenarios from the area of service engineering for our proposed evaluation.

## 1 Introduction

IT-systems have to deal with a diversity of specifications, which are often heterogeneous, thus naturally leading to redundancy and interdependencies. Maintaining and defining relations and dependencies between specifications is a nontrivial task and has been a known challenge since the early 1970s [ARNRSG06]. The inception of model-driven engineering (MDE) entails the use of models as specifications for different domains, such as the modelling of business processes, system requirements, architecture, and tests. Although

---

\*This work is done in context of a doctoral work at the chair of Software Technology of Prof. Uwe Aßmann at the Technical University of Dresden. The work was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The author takes responsibility for the contents.

MDE advocates a support of linking and round trip engineering by model transformations, reality still fails to deliver on this promises [Sel03].

This is tackled by semi-automatic calculation of links based on similarity as in model differencing and model matching. Semi-automatic model differencing aims on calculating the difference between models, e. g. which element has been removed. In contrast, meta-model matching calculates similarities between elements, i. e. it targets link creation. Matching and differencing can be applied on meta-models as well as models. Recently, several approaches arose to support model differencing with focus on model versioning, e. g. targeting the meta-model [FHLN08] level. Thereby, EMF Compare [BP08], SiDiff [SG08] and DSM-Diff [LGJ07] are the ones supporting differencing at meta-model and model level. However, due to their aim of differencing rather than matching, none of the approaches tackles the *similarity* calculation for trace link generation, which has been identified as a major challenge in model traceability [ARNRSG06]. Automatic trace link generation is supported by most model transformation engines, however nowadays many transformations are implemented manually (e. g. in Java), thus lacking an automatic trace link generation.

We propose to tackle these problems of trace link generation by presenting MatchBox; a configurable matching system capable of meta-model matching along with an infrastructure to manage and create mappings. Thereby, we adapt and extend schema matching techniques [VIR10] for model matching. To present these contributions, we first introduce the challenges of model matching by means of trace link creation in Section 2. Afterwards, we describe our contributions to address these challenges in detail by explaining MatchBox, its components, process, and current status. Subsequently, we present our evaluation and give an overview of related work, to finally conclude in Section 5.

## 2 Challenges of Matching for Trace Link Creation

The illustrative scenario for trace link generation in a MDE process is depicted in Figure 1 on the left. Thereby, a model is transformed into another, preferably this is done using a transformation engine. If so, an approach as in [GV09] will create trace links connecting the original and resulting model. Then, the resulting model is adjusted manually, e. g. to serve specific user needs, which potentially leads to unlinked elements. In the worst-case scenario a transformation engine creating the links is missing. The arising challenge is to create the missing trace links. The right side depicts our approach on applying our matching system which creates trace links based on the calculated similarities of elements on meta-model and model level. The projection of the meta-model elements on model level is illustrated, denoting the reuse of meta-model mappings for search space reduction on model level.

In the context of trace link creation, the participating models (esp. on instance level) easily constitute 5000 or more elements<sup>1</sup>, which need to be evaluated for trace links. Considering a matching approach, 5000 x 5000 elements have to be compared resulting in 25 million

---

<sup>1</sup>For instance 400 Line Java Class represented in JaMopp (<http://jamopp.inf.tu-dresden.de/>)

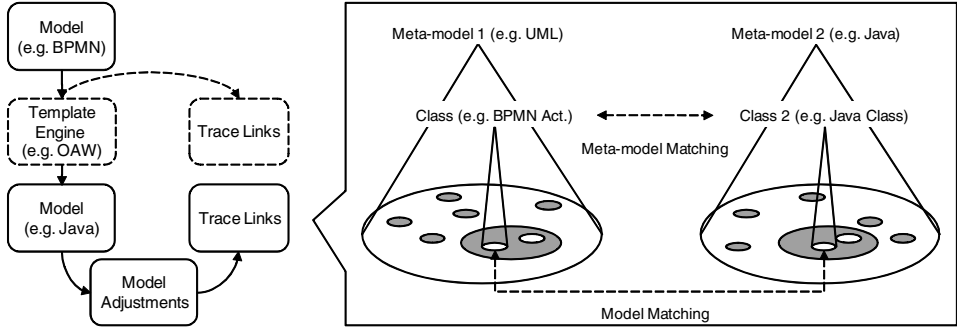


Figure 1: Example trace link scenario and relation to model and meta-model matching

calculations. Therefore, applying matching techniques for trace link creation requires a high *scalability*. In this context scalability means that the system is capable of handling instances of growing dimensions and that they are processed in a reasonable time.

Another challenge covers the *quality* of trace links obtained, i. e. the amount of correctly identified trace links and the ratio of all identified links to the correct set of links. To support common traceability scenarios as impact, orphan or system analysis [GV09], a certain quality has to be ensured in order to provide reliable results.

Finally, a significant *evaluation* of the concepts is a major challenge. In order to demonstrate the capabilities and limitations and evaluate the framework an evaluation based on a broad range of examples is needed providing a variety in structure and complexity as well as heterogeneity in specifications (models).

### 3 The MatchBox Approach

We propose MatchBox a system that provides an infrastructure for meta model and model matching which addresses the challenges of trace link generation. Thereby, we make the following contributions (1) A generic architecture for a model matching infrastructure for trace link generation, (2) a graph-based clustering approach to cope with the dimensions of model matching, (3) planar graph-based matching techniques for improved matching results, (4) a utilization of meta-model matching results for model matching, and (5) a large-scale evaluation concept for model matching approaches.

Figure 2 depicts an overview of MatchBox along with its four processing steps, each supported by a component. The process starts with the adaptation component (1) which is responsible for an import of meta-models and models into our internal graph-based representation. The clustering component (2) separates the input graph into different parts, thus reducing the dimension of the matching problem. The matching core (3) is an exchangeable component. It follows an interface that requests a similarity value between two model elements which is the task of matching. The matching techniques applied range from name-bases similarity, to data types, parent-child relations, and graph-based

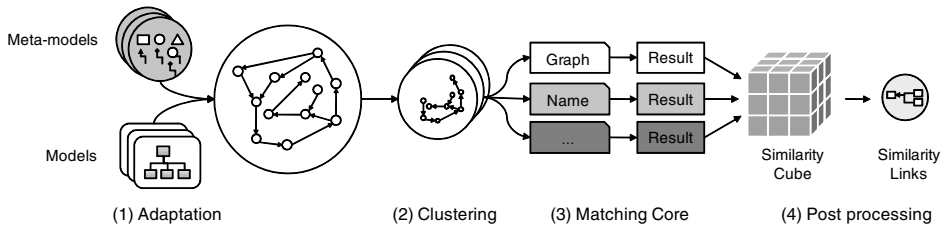


Figure 2: Conceptual overview on the process, internal model and matching in MatchBox

ones. For further details please refer to [VIR10]. Finally the post-processing component (4) uses created mappings (links) to remove contradicting or duplicate mappings. Put simply, MatchBox takes two models as input and produces mappings as output. In the following, we will argue how we tackle the challenges given in Section 2.

**Scalability** The challenge of scalability is tackled by following a divide-and-conquer approach, thus reducing the actual problem dimension. This is achieved by clustering the input graph into smaller sub graphs (clusters) which are matched with each other. These clusters will be determined by results given from meta-model matching, thus arranging elements by their corresponding mapped type and strongly connected components. Applying isomorphism algorithms on graphs in general results in NP-complete problems. However, algorithms based on certain graph characteristics as planarity or reducibility are promising to determine matchings in reasonable time, thus increasing scalability.

**Matching Quality** The matching on both layers makes use of common schema matching techniques such as name edit distance and parent-child similarity propagation as described in [VIR10] and additional graph matchers using structural information and graph characteristics. Again isomorphism or graph edit distance algorithms for general graphs are NP-complete, but, for instance planarity provides a linear algorithm for the calculation of an error-tolerant approximate graph edit distance on attributed graphs [NB04]. In first experiments on meta-model level, we achieved promising results<sup>2</sup>. Our first prototype of our meta-model matching system MatchBox is based on EMF which provides a common space for integration of models. Therefore, we do not need to struggle with the extraction of information and integration.

**Evaluation of MatchBox** Our evaluation has to consider meta-model matching and model matching. The *matching quality* is commonly determined by using the statistical measures precision, recall and F-Measure from the field of information retrieval [Rij79]. These measures are calculated by comparing the result obtained by a matching system to a so-called gold-standard which is the perfect match. For meta-model matching, we propose to use model transformations as gold-standards which enable an automatic extraction

<sup>2</sup>An improvement up to 25% compared to traditional matching techniques (name, name path and children [VIR10])

of gold-standards and an extensible evaluation base. One example of a source for gold-standards is the ATL-Zoo<sup>3</sup> consisting of more than hundred model transformations. For our evaluation of model matching we will make use of Service Engineering [CVW08] and several service descriptions and models in the context of the TEXO project each being made of 8 models with 10 up to 15,000 elements. We will further investigate the mappings contained in the SAP Enterprise Service Repository (ESR)<sup>4</sup> as a base for our evaluation, which will grant access to at least 30 real world gold standards for services interface mappings. Regarding *scalability* we will apply our approach to a variety of synthetic and real-world models with differing size and complexity as given in the previous paragraph. Thereby, we will take measurements such as runtime and memory consumption. Finally, we propose to compare MatchBox with other matching approaches.

## 4 Related Work

Model transformation engines in general allow for trace link generation by default. However, we especially tackle the problem of having no transformation engine present. The problem of model element similarity calculation is dealt with by semi-automatic matching. However, current systems lack quality in terms of matching results obtained, scalability, and a comprehensive evaluation. There are three systems supporting layer independent matching, i. e. semi-automatic matching of meta-model and models.

EMF Compare [BP08] is the most prominent one possessing a broad user base since it is an EMF-related Eclipse project. We applied experiments with more than 120 meta-models<sup>5</sup> comparing it with MatchBox and noted a quality improvement when using MatchBox in terms of precision and recall regarding the matches by factor 3-4. This is due to the fact that EMF Compare uses a simple algorithm which computes label and string edit distances. Furthermore, EMF Compare present a weak concepts for scalability by only considering tree-based neighbours. DSMdiff [LGJ07] uses type information of meta-models to encode characteristics in strings being compared. The structural information used is as simple as the number of children or references. This leaves room for improvement in matching quality. Again they do not present a concept to cope with the dimensions of model matching and thus scalability. SiDiff [SG08] presents an algorithm traversing a tree bottom up and top down similar to our parent-child similarity propagation. However, in contrast to them we consider the graph structure as a whole. Finally, all approaches consider model differencing, thus adding potentially misleading information by match result interpretation. In contrast, we propose matching only. All of the matching approaches evaluate their systems by a limited number of simple artificial examples e. g. two similar meta-models (e. g. UML and MiniJava [FHLN08]) or two trivial models (UML-Class diagrams [BP08, LGJ07]) which indeed yield good results due to many identical elements. In contrast, we target a comprehensive real-world evaluation.

---

<sup>3</sup><http://www.eclipse.org/atl/zoo>

<sup>4</sup><http://esworkplace.sap.com>

<sup>5</sup>ATL-Zoo, ModelCVS and Ontology Alignment Contest

## 5 Conclusion

We have presented MatchBox, a system for layer independent matching allowing for meta-model and model matching based on a combining approach making use of graph characteristics for matching. We have given an overview of the current state of the art in matching and have identified challenges in link creation that we address with MatchBox. Thereby, we cover the complete matching process, by contributing improved matching techniques, using clustering for increased scalability and a real-world evaluation based on service descriptions and implementations. The work on the graph characteristics and use of meta-model matching information is currently ongoing. In parallel, we implement and extend the architecture and infrastructure step-wise to a complete version of MatchBox.

## References

- [ARNRSG06] N. Aizenbud-Reshef, B.T. Nolan, J. Rubin, and Y. Shaham-Gafni. Model traceability. *IBM Systems J.*, 45(3):515–526, 2006.
- [BP08] C. Brun and A. Pierantonio. Model differences in the eclipse modeling framework. *Upgrade, Special Issue on Model-Driven Software Development IX*, 2008.
- [CVW08] Jorge Cardoso, Konrad Voigt, and Matthias Winkler. Service Engineering for The Internet of Services. In *Enterprise Information Systems X*. Springer, 2008.
- [FHLN08] Jean-Rémy Falleri, Marianne Huchard, Mathieu Lafourcade, and Clémentine Nebut. Metamodel Matching for Automatic Model Transformation Generation. In *Proc. of MoDELS '08*, pages 326–340, 2008.
- [GV09] Birgit Grammel and Konrad Voigt. Foundations for a Generic Traceability Framework in Model-Driven Software Engineering. In *Proc. of the ECMDA Traceability Workshop '09*, 2009.
- [LGJ07] Y. Lin, J. Gray, and F. Jouault. DSMDiff: a differentiation tool for domain-specific models. *European Journal of Information Systems*, 16(4):349–361, 2007.
- [NB04] M. Neuhaus and H. Bunke. An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. *LNCS*, pages 180–189, 2004.
- [Rij79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.
- [Sel03] B. Selic. The pragmatics of model-driven development. *IEEE software*, 20(5):19–25, 2003.
- [SG08] Maik Schmidt and Tilman Gloetzner. Constructing difference tools for models using the SiDiff framework. In *ICSE Companion '08*, pages 947–948. ACM, 2008.
- [VIR10] Konrad Voigt, Petko Ivanov, and Andreas Rummler. MatchBox: Combined Meta-model Matching for Semi-automatic Mapping Generation. In *Proc. of ACM SAC '10*, 2010.

# Product Certification of Component Performance Specifications

Henning Groenda  
FZI Forschungszentrum Informatik  
Haid-und-Neu-Str. 10-14  
76131 Karlsruhe  
groenda@fzi.de

**Abstract:** In software engineering, performance analyses and predictions play an important role in the selection of components and the evolution of complex component-based systems. These analyses and predictions are based on parameterized performance specifications. However, the quality of the specifications and their trustworthiness usually remain unspecified. In existing approaches, it remains unclear if a specification can be reused in another context and which effect its use may have on the quality of the analysis or prediction. In this paper, we propose a test-based approach to validate parameterized performance specifications against deployed component implementations. The validation is used to certify the quality and valid parameter ranges of the specifications.

## 1 Introduction

Performance analyses and predictions play an important role in each life cycle phase of a component-based software system. Their application in the development phase enables software engineers to improve architecture and design by comparing design alternative based on objective measures. In the deployment phase, performance specifications guide the selection and sizing of an appropriate execution environment as well as the deployment of components within this environment. In the maintenance or evolution phase, analyses allow to examine performance bottlenecks or the effect of modifications on the performance and allow guiding maintenance effort.

All of these performance analyses are based on specifications of the component's performance-relevant behavior. The specifications are always abstractions of the performance-relevant behavior to allow faster analyses than by executing the software itself. The degree of parameterization of these specifications depends on the supported scenarios. Example scenarios are changes in the usage profile, assembly or deployment of components. Due to the complexity of these scenarios the specifications are usually only valid for a certain combination of parameter ranges. However, the quality of the specifications for such a combination is usually not stated. Having such statements enables identifying potential risks and reasoning about the overall quality of the analysis or prediction.

Existing performance prediction approaches rely on the capability of software engineers



to select suitable performance specifications of a component. Research focused on validating prediction approaches with the assumption that the specifications were suitable for the situation at hand. Validating performance specifications was a human activity in the selection process. Missing validity statements require substantial effort to reason about the suitability or quality of an performance specification. Especially if specifications should be reused in different context, for example because of late composition, effort has to be put in revalidation or maybe even recreation. This process step requires expert knowledge and is time-consuming which reduces the usefulness of specifications of unknown quality.

In this paper, we propose an approach to certify performance specifications and explicitly state the quality and limitation of these specifications. It is based on a test-based validation of the specifications against deployed component implementations. The validation is trustworthy as it is based on objective and verifiable information. This enables third party validation and eases testing the validity. Additionally, it aids in the protection of interests and trade secrets in marketplaces as it is sufficient to publish the certified specifications.

The paper is structured as follows. Section II shows what needs to be covered to reason about the quality of performance specifications. Section III sketches the certification process. Section IV point out and discusses related work. Section V concludes the paper.

## 2 The Quality of Performance Specifications

Validation of specifications is often made based on test suites. For example, the functional validation if an application server fulfills the Java platform enterprise edition requirements. Each test in such a test suite provides a simple pass or fail outcome by comparing expected and experienced results.

However, if performance is considered the comparison of expected and experienced results is much more difficult. Performance is for example influenced by input parameters, internal state, the performance of other required services, resource contention, and the deployment platform. Depending on the used measurement method and granularity, performance measurements often influence the performance itself. Measuring wall-clock time is error prone due to interruptions in multi-threading systems, whereas profiling has a heavy performance impact itself. See [KKR09] for details on measurement methods.

Performance specifications consists of connected resource demands which may depend on parameters. They can be as easy as a single value (e.g. mean) covering huge code areas but also as complex as an executable specification (e.g. byte code). The presented approach hence focuses on the following three aspects to ensure the overall specification quality:

*Input Range* The input range specifies the parameter ranges of parameterized specifications for which the specification is valid. For example, if a specification has the parameter *file size* and it is validated for file sizes between 3 and 50 MB this is stated as input range.

*Resource Demand* Resource demands are measured in the number of experienced byte code instructions. This metric is not influenced by the measurement itself and can, in combination with Kuperberg's approach [Kup09], be validated for any given hardware and

software environment. For each considered byte code instruction, a *Validation Estimator* expresses the maximal accepted deviance from the specified value.

### 3 Certification of Performance Specifications

Our certification approach implements a test-based validation of performance specifications against deployed component implementation. The certification process is explained in the following paragraphs.

At the beginning, the component creator issues a validation request to the certification authority. He has to provide the component specification, the corresponding implementation, and the quality statements including the parameter range combinations which should be used for validation. Validity statements contain the information discussed in chapter 2.

An evaluator within the certification authority then assesses the validity of the specification. Execution of the following steps is necessary to decide upon issuing a certificate.

*Generate Testcases* In this step, test cases are generated automatically which are used to evaluate the specification. Automatic generation is chosen to ensure reproducibility of validation results and reduce the necessary human effort. Stratified random sampling is used to generate parameters. If the initial seed is stored, the results can be reproduced but unwanted implementation optimization on test values is still hindered. Overall, the samples cover the specified input range for each parameter.

Both directions, specification against executed implementation and vice versa, are checked by the test cases. The forward direction checks if the statements in the specification are correct, for example that a resource demand depends linearly on an input parameter. The backward direction checks that the implementation does not contain more dependencies than the ones specified, for example unspecified calls to external services, another sequence of calls, or unspecified dependencies to return values of calls.

*Instrument Implementation* The implementation is instrumented to log the CPU resource demands in form of issued byte code instructions. These demands are correlated to the specified demands.

*Deploy Implementation* The instrumented implementation and mock-ups for required services are deployed in the target environment.

*Run Testcases* The testcases derived in the first step are run on the deployed implementation and measurements of the issued CPU resource demands are gathered. A test case is run until the requested sample size is reached, which can be either specified directly or being calculated by a confidence level, e.g. if confidence interval of the sample mean is below the deviation threshold. The coverage of the test cases on the implementation allows to reason about the quality of the test.

The presented approach currently has the following limitations and assumptions:

- The approach currently considers only the validation of resource demands for processors. Hard disk or network requests are not validated yet.

- It focuses on the performance needs of business information systems. It does not consider guarantees, for example by a schedulability analysis, which are important in many embedded systems. The methodology is intended to reason about common and not best or worst case execution times.
- In a former prototype, resource demand specifications are only validated for fixed resource demands which are measured execution times and not byte code instructions. It is described with a small case study in [Gro09].
- Checks that the implementation does not contain more dependencies than specified are currently not implemented.
- Data dependencies of components are currently not validated yet. However, the support for parameters of the used performance specifications allows taking these into account later on.
- Plain java objects are considered. Code weaving which is for example used in Java Enterprise Edition application servers is not considered yet.

## 4 Related Work

The research on component certification started in the early 90s and is still ongoing as shown in Alvaro et al.'s survey [AAM05]. The survey shows the history of component certification and that certification approaches developed in the 90s focus on statements about the reliability of software components using test cases or mathematically analyzable models. Starting around 2000, the focus of the approaches shifted towards the certification of extra-functional aspects in general and the prediction of systems built out of certified components, e.g. [SW01, HMSW03].

Wallnau also did some basic work on classifying certification approaches, for example the 10 useful distinctions for certification approaches in [Wal04]. According to this classification the approach presented in this paper aims at reducing the gap between the knowledge about what a component does to what it actually does. It supports a *descriptive* certification of *objective* measures. The software *products* will be examined *empirically* with a given *context* in a *procedural* manner.

Hissam, Moreno, Wallnau et al. introduced the concept of predictable assemblies in [HMSW03] and later extended it to the *Predictable Assembly from Certifiable Components (PACC)* approach, described in [Wal03]. PACC allows the prediction of the runtime behavior of a software system from the properties of its components and their patterns of interactions. PACC's performance reasoning framework currently focuses on fixed-priority preemptive scheduling, making it suitable to analyze hard real-time systems [MM08]. In contrast to the approach proposed in this paper, the authors concentrate on the support for hard real-time systems instead of business information systems. Additionally, they focus on the small areas and conditions in which some correctness properties can be proven.

Alvaro et al. show in [ALC07] the need of component certification within component-based software development for business information systems and discuss similarities and interdependencies between component selection and certification. The authors also provide a framework for component selection [AAM07b] as well as an selection process [AAM07a]. The approach proposed in this paper focuses solely on the extra-functional aspect performance and could later on be integrated this framework.

The effort of testing and reusing components was addressed by Weyuker in [Wey98]. She states that high reliability and availability requirements lead to enormous costs. Additionally, components have to be tested in isolation and after integration so savings of components-of-the-shelf are not sure. Reusing components requires retesting for stability, reliability, stress, and performance testing.

The approach presented in this paper will allow testing components for specified range combinations. This allows reasoning if a specification is suitable for the planned analysis.

## 5 Conclusion

In this paper, we showed how the combination of Input Range, Resource Demand, and Validation Estimator is used to reason about the quality of performance specifications. We presented our approach and explained the process of assessment and certification as well as listed current limitations and assumptions.

The presented approach aids companies in offering components in marketplaces. The publication of certified performance specifications in a marketplace is sufficient for potential customers to evaluate and select components using prediction approaches like the PCM<sup>1</sup>. However, the interests and intellectual properties of the offering companies are still protected as the specifications only contain a highly abstract view on the component's behavior and keep the disclosure of details on the used algorithms and techniques to a minimum. Having certified performance specifications additionally supports software engineers in late composition of components. The software engineers gain the knowledge if the performance specifications fulfill their requirements for the intended composition which can in turn ease the evaluation of components and reduce the necessary effort. The certification of specifications also supports a predictable assembly of components. The information contained in these specifications allows increasing confidence in the results produced by validated performance prediction approaches or identifying potential risks. Last but not least, certification by independent authorities provides a mean for quality assurance. If the development of components is given to a contractor the stipulation of certification enables the contracting body to trust the performance of a developed component beyond a few test cases while keeping its own quality assurance effort low. The contracting body can focus on its own expertise and does not need to employ performance engineers just for quality assurance.

As a next step the measurement of byte code instructions instead of execution times is

---

<sup>1</sup><http://www.palladio-approach.net>

planned. This allows the certification independent of the target environment. The execution times can be calculated based on benchmarking the instructions in the target environment. It is also planned to advance considered parameters to data dependencies like randomness which influence resource demand during compression. It is further planned to examine the necessary effort in terms of test cases and runs for the requested validation.

## References

- [AAM05] Alexandre Alvaro, Eduardo Almeida, and Silvio Meira. Software component certification: a survey. *SEAA*, pages 106–113, September 2005.
- [AAM07a] Alexandre Alvaro, Eduardo Almeida, and Silvio Meira. Component Quality Assurance: Towards a Software Component Certification Process. *IRI 2007*, pages 134–139, August 2007.
- [AAM07b] Alexandre Alvaro, Eduardo Almeida, and Silvio Meira. Towards a Software Component Certification Framework. *QSIC 2007*, pages 298–303, October 2007.
- [ALC07] Alexandre Alvaro, Rikard Land, and Ivica Crnkovic. Software Component Evaluation: A Theoretical Study on Component Selection and Certification. Technical Report MDH-MRTC-217/2007-1-SE, Mälardalen University, November 2007.
- [Gro09] Henning Groenda. Certification of Software Component Performance Specifications. In *WCOP 2009*, pages 13–21, 2009.
- [HMSW03] Scott Hissam, Gabriel Moreno, Judith Stafford, and Kurt Wallnau. Enabling predictable assembly. *J. Syst. Softw.*, 65(3):185–198, 2003.
- [KKR09] Michael Kuperberg, Martin Krogmann, and Ralf Reussner. TimerMeter: Quantifying Properties of Software Timers for System Analysis. In *QEST'09*, pages 85–94, September 2009.
- [Kup09] Michael Kuperberg. FOBIC: A Platform-Independent Performance Metric based on Dynamic Java Bytecode Counts. In Felix C. Freiling, Irene Eusgeld, and Ralf Reussner, editors, *Dependability Metrics Research Workshop*, pages 7–11. Department of Computer Science, University of Mannheim, May 2009.
- [MM08] Gabriel A. Moreno and Paulo Merson. Model-Driven Performance Analysis. In Stefan Becker, Frantisek Plasil, and Ralf Reussner, editors, *QoSA 2008*, volume 5281 of *LNCS*, pages 135–152. Springer, September 2008.
- [SW01] J. Stafford and K. Wallnau. Is Third Party Certification Necessary? In *4th ICSE Workshop on Component-Based Software Engineering*, 2001.
- [Wal03] Kurt C. Wallnau. Volume III: A Technology for Predictable Assembly from Certifiable Components (PACC). Technical Report CMU/SEI-2003-TR-009, SEI, CMU, 2003.
- [Wal04] Kurt C. Wallnau. Software Component Certification: 10 Useful Distinctions. Technical Report CMU/SEI-2004-TN-031, SEI, CMU, September 2004.
- [Wey98] Elaine J. Weyuker. Testing Component-Based Software: A Cautionary Tale. *Software, IEEE*, 15(5):54–59, Sep/Oct 1998.

# Towards Information Flow Auditing in Workflows

Claus Wonnemann  
Department of Telematics  
Albert-Ludwigs-Universität Freiburg, Germany  
wonnemann@iig.uni-freiburg.de

## Abstract:

The paper proposes an approach for compliance audits in workflow environments based on the tracking of information flow. Requirements are formalized as a binary relation on the workflow principals. The workflows' execution logs are transferred into graph-based representations of the explicit information flows (dataflows) and adherence to compliance requirements is checked while traversing these graphs. The scope and limits are discussed and the major milestones for further work are outlined.

## 1 Introduction

Many compliance requirements that are applicable to IT systems somehow regulate the flow of information, e.g. when some sensitive information must not be disclosed to unauthorized parties, or if some datum has to pass a number of checks before being released (e.g. to ensure the four-eyes-principle). Such information flow constraints can be expressed as relations among system principals, independently from the concrete implementation. This formalization is stronger than access control policies because it captures information propagation throughout the system (*end-to-end*) rather than access at certain points. Further, it captures *implicit* information flows, which lie outside the scope of access control mechanisms (see below).

This paper proposes an approach for compliance audits in workflow environments based on the a-posteriori tracking of information flow. We present the IFAudit system, which consists of a policy language for the expression of information flow constraints and an audit algorithm to track explicit information flow (dataflow), based on log data. We discuss the scope and limits of the implementation and outline the most important milestones for further work.

**Types of Information Flow** There are two types of information flow: explicit and implicit. Explicit flow is information transfer through “legitimate” channels that are intended for this purpose, such as messages and shared storage. Implicit flow describes the extraction of information through *inference*. This is the case when multiple data items are combined to yield information, e.g. when anonymized patient records are matched against address directories to reduce the anonymity set and, eventually, reveal a patient's identity. Information can also be inferred from the observable behavior of the system. For instance,

by observing the response times of a crypto system, an attacker might infer knowledge on its private key. Here, the system’s timing behavior forms a *covert channel* that enables an attacker to obtain secret information.

## 2 Workflows and Log Data

Workflows formally describe business processes as structured sequences of activities. The execution of a workflow is coordinated by an execution engine which triggers activities, synchronizes their interaction and communication, and writes log data. These log files are the basis for the audits considered in this paper.

Log files of workflow systems are structured as sequences of entries that each represent a workflow activity. An entry comprises (among other attributes) the originator, on which behalf the activity was performed, and the data items that were read (input) and written (output) by the activity. Further, it is assumed that each entry can be attributed to a workflow instance and that the entries are totally ordered (e.g. by providing a timestamp).

Expressed formally,  $\mathcal{L}_{\mathcal{W}}$  denotes the log file for a workflow  $\mathcal{W}$ , with  $\mathcal{L}_{\mathcal{W}} \in \mathcal{A}^*$ , where  $\mathcal{A}$  is the universe of all log file entries.  $\mathcal{L}_{\mathcal{W}}$  can be partitioned into a set  $\{i_0^{\mathcal{W}}, i_1^{\mathcal{W}}, \dots, i_n^{\mathcal{W}}\}$  where  $i_i^{\mathcal{W}}$  denotes the trace (ordered sequence of log entries) for the  $i^{\text{th}}$  execution of  $\mathcal{W}$  ( $0 \leq i \leq n$ ). An entry  $a$  has the attributes *orig* (the originator), *input* (the set of input data items) and *output* (the set of output data items). The attributes of an entry are referenced with a dot (e.g.  $a.orig$ ). For convenience, *activity* and *log entry* (of the activity) are used interchangeably in the following.

## 3 Policies

An information flow policy  $\mathcal{P} = \{r_1, \dots, r_n\}$  is a set of rules that specify which principals in a system may or must not exchange information. The principals in our setting are the originators (subjects) that appear in a log file  $\mathcal{L}_{\mathcal{W}}$ , denoted as  $\mathcal{S}_{\mathcal{W}} = \bigcup_{a \in \mathcal{L}_{\mathcal{W}}} a.orig$ .

The function  $sc : \mathcal{S} \rightarrow \mathcal{D}$  assigns principals security domains from the set  $\mathcal{D}$  of domains. The information flow policy specifies among which security domains information may or must not be exchanged. An activity has the same domain as its originator.

Fig. 1(a) shows a typical multi-level security policy with three domains (a crossed arrow indicates that there must not be information flow). Compliance requirements often demand that some data has to take a specific path, for instance, to ensure that financial statements are checked by the revision department before being published to investors. Fig. 1(b) depicts a corresponding policy: information coming from the *Accounting* domain must not flow directly to the *IR* department, but has to pass through *Revision*.

A policy rule has the form *Restriction*  $\Rightarrow$  *Exception*. *Restriction* is a flow relation  $source \rightsquigarrow target$ , which specifies that information must not flow from domain *source* to domain *target* ( $source, target \in \mathcal{D}$ ). *Exception* is a logical combination of flow re-

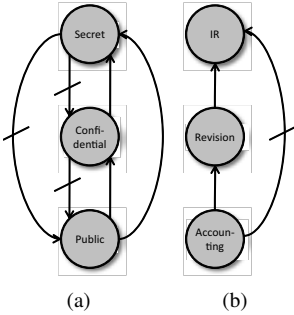


Figure 1: Example Flow Policies.

```

<Policy>      ::= <Rule> | <Rule>, <Policy>
<Rule>       ::= <Restriction> =>
               <Exception>

<Restriction> ::= true | <FlowRel>
<Exception>  ::= false | <FR-DNF>
<FR-DNF>    ::= (<ConClause>) |
               (<ConClause>) ∨ <FR-DNF>

<ConClause>  ::= <FlowRel> |
               <FlowRel> ∧ <ConClause>

<FlowRel>   ::= <Domain>~<Domain>
<Domain>    ::=  $d \in \mathcal{D}$ 

```

Figure 2: Policy grammar in BNF.

lations in disjunctive normal form which defines legitimate flows that might contradict *Restriction* (as in the example in Fig. 1(b)). Fig. 2 gives the grammar for these policies in Backus-Naur-Form.

Using this grammar, the example policies from Fig. 1 have the form

```

Secret~Confidential => false,
Confidential~Public => false,
Secret~Public => false

```

and

```

Accounting~IR => (Accounting~Revision ∧ Revision~IR),

```

respectively.

Besides the (extensional) specification of flow relations, which express whether information may flow between domains, an information flow policy must define which (patterns of) system actions constitute information transmission. In its current state, the IFAudit system only supports the detection of dataflow, i.e. explicit information flow: There is an information flow from domain  $d_1$  to domain  $d_2$ , if (and only if) there is a data item which has been modified by an activity from  $d_1$  and is subsequently read by an activity from  $d_2$ . This approach is similar to the tainting mode in the Perl programming language. It is sufficiently expressive to capture common compliance violations, such as failure to “sanitize” information or illicit propagation of some data item. A more comprehensive and fine-grained analysis is subject of ongoing work (see Section 6).

## 4 Audit Algorithm

The IFAudit system analyzes explicit information flow on the workflow level, i.e. the information transfer that occurs among the activities. Activities are regarded as black boxes of which the internal workings are not necessarily known, but only their I/O. On this abstraction level, IFAudit allows to check whether a workflow instance has exhibited illicit



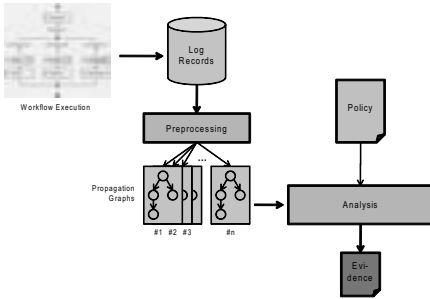


Figure 3: Outline of the IFAudit system.

```

VISIT( $u$ ):
for each (Rule  $r \in OpenRules$  that is closed by  $u$ ) do
  Report violation of  $r$  through flow(s) from  $SRC(r)$  to  $u$ ;
for each (Rule  $r \in \mathcal{P}$  that is opened by  $u$ ) do
   $OpenRules := OpenRules \cup r$ ;
   $SRC(r) = SRC(r) \cup u$ ;
Mark  $u$  as visited;
 $Successors :=$  Direct successors of  $u$ ;
Sort  $Successors$  in ascending order;
for each ( $v$  in  $Successors$ ) do
  if ( $v$  is not marked as visited) then
    VISIT( $v$ );
for each (Rule  $r \in \mathcal{P}$  that is opened by  $u$ ) do
   $SRC(r) = SRC(r) \setminus u$ ;
  if ( $SRC(r) = \{\}$ ) then
     $OpenRules := OpenRules \setminus r$ ;

```

Figure 4: Pseudo-code for VISIT.

information flows with respect to a specified policy.

The structure of IFAudit is outlined in Fig. 3. The log entries of each workflow instance are transferred into a *propagation graph* (PG) that represents the explicit information flows (dataflows) between the involved principals. The PGs are subsequently analyzed for illicit information flows according to the given policy. The PG for a trace  $i_n^W \subseteq \mathcal{L}_W$  is denoted  $PG_n^W$ . It is defined as a graph  $(V, E)$ , where  $V = \{a \in \mathcal{A} \mid a \in i_n^W\}$  is a set of activities (representing the graph’s vertices), and  $E = \{(a, b) \in (\mathcal{A} \times \mathcal{A}) \mid a < b \wedge a.output \cap b.input \neq \{\}\}$  is an asymmetric relation representing the graph’s (directed) edges. An edge  $(a, b)$  in a PG indicates that there was a dataflow from activity  $a$  to activity  $b$ .

**Policy Check** To check whether a workflow instance adheres to a given policy, its PG is traversed in a depth-first fashion. During traversal, it is checked whether an activity marks the starting point of an illicit information flow (i.e. if it is assigned a security domain that is the source of a rule’s restriction). Such rules are kept in a set of “open” rules and checked whether they are “closed” (i.e. violated) by a subsequent activity. An activity closes a rule, if its assigned security class corresponds to the end point of that rule’s restriction, and if the rule’s *exception*-clause evaluates to *false*. Fig. 4 shows the pseudo-code for the procedure VISIT, which checks whether an activity closes or opens a rule before recursively applying itself to each of the activity’s successors. The field *OpenRules* contains the set of rules that have been opened by one or many activities residing above the current activity in the PG. The mapping SRC maps each rule from *OpenRules* to the set of activities that have opened that rule.

## 5 Related Work

The Examina system allows the formalization and a-posteriori checking of privacy policies [Acc08]. However, there is no explicit notion of information flow, i.e. information cannot be tracked over several data items. Research in *process mining* focuses on building models of the process’ control structure (e.g. using Petri nets) [vdAWM04]; concerning

security aspects, there is some work addressing the detection of anomalous process behavior [vdAdM05]. Research on information flow control concentrates almost exclusively on the prevention of illicit flow through static and dynamic methods [SM03]. The a-posteriori detection has previously been addressed by the author [AW09].

## 6 Conclusion and Future Work

The IFAudit system allows to check whether information in workflow applications has flown along specified paths. Therefore it can validate adherence to those compliance rules that constrain information propagation. In its current state, IFAudit is restricted to dataflow, which is arguably the most relevant type of information flow in the compliance context. Since information flow in a computer system is, in general, undecidable [DD77], IFAudit uses a common approximation: it is assumed that an activity reading a datum completely absorbs the contained information and passes it entirely to every output datum. This is a conservative assumption which ensures that every (explicit) information flow is captured. The drawback is that the system is likely to generate many false positives, when an informational dependency is assumed where there is not one. Therefore, precision improvement is a major subject of current and future work (see below).

We currently evaluate the correctness and efficiency of the IFAudit system with a proof-of-concept implementation. For this purpose, a simulation environment for workflows was built, which can execute multiple instances of a workflow model (parametrized with varying input) and write the corresponding log files. Further, it is planned to extend IFAudit in multiple directions:

**Inference** The accumulation of multiple data items causes illicit flow, if, for instance, multiple anonymized patient records can be combined to reveal an identity. In order to detect such flows, we plan to integrate a mechanism that allows the formalization of inference capabilities in terms of inference rules. This way, it can be checked whether there might occur illicit flow whenever multiple data items converge at some principal, possibly in the course of multiple executions. Inference rules can also be used to formalize common compliance policies such as Chinese Wall: here it must be ruled out that information from competing clients ever converge at the same principal.

**Formalization and Detection of Implicit Flow** To capture implicit information flows through covert channels, more sophisticated flow semantics than tainting are needed. The classical notion is *noninterference*, the requirement that the interactions of higher-level users with a system must not influence the interactions of lower-level users [GM82].

We are currently working on an audit algorithm for the detection of implicit flow based on information theory. It measures how much the entropy of hidden (high-level) variables decreases when public (low-level) variables can be observed. If the entropy drops significantly, there is information transmission from the hidden to the public variable. Conversely, noninterference holds when the entropy does not change at all. The advantage of

the audit approach compared to static methods is that the transferred information can be precisely quantified (with respect to the given log data). While static methods can determine whether the possibility of implicit information transfer exists (in principle), audits can determine whether such a transfer has indeed occurred, and to which extent.

**Declassification** Generic flow semantics such as noninterference in its original form are too restrictive for most applications, since they characterize information flows through causal flows and reject secure programs where there is a causal flow without an associated information flow [Rya01]. Further, certain “downward” flows are necessary in virtually every application: a standard password check is considered insecure because even the rejection of a wrong guess reveals some information on the password (namely, what it is *not*). Instead of requiring no inference between high and low *at all*, it is more reasonable to require that the higher-level behavior must not influence lower-level observations *in a critical way*. This issue is known as *declassification* (for an overview see [SS05]). We are currently seeking to integrate a suitable downgrading mechanism into IFAudit that allows to consider encrypted channels and other “sanitizing” effects in a workflow.

## References

- [Acc08] Rafael Accorsi. Automated Privacy Audits to Complement the Notion of Control for Identity Management. In *Policies and Research in Identity Management*. Springer, 2008.
- [AW09] Rafael Accorsi and Claus Wonnemann. Detective Information Flow Analysis for Business Processes. In *BPSC*, pages 223–224, 2009.
- [CS08] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. In *Proceedings 21st IEEE CSF*, pages 51–65, Washington, DC, USA, 2008. IEEE Computer Society.
- [DD77] Dorothy E. Denning and Peter J. Denning. Certification of programs for secure information flow. *Commun. ACM*, 20(7):504–513, 1977.
- [GM82] Joseph A. Goguen and José Meseguer. Security Policies and Security Models. In *Proceedings IEEE Symposium on Security and Privacy*, pages 11–20, 1982.
- [Rya01] Peter Y. A. Ryan. Mathematical Models of Computer Security. In *Foundations of Security Analysis and Design*, pages 1–62, London, UK, 2001. Springer-Verlag.
- [SM03] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.
- [SS05] Andrei Sabelfeld and David Sands. Dimensions and Principles of Declassification. In *Proceedings 18th IEEE CSFW*, pages 255–269. IEEE Computer Society, 2005.
- [vdAdM05] W. v.d.Aalst and A. de Medeiros. Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance. *ENTCS*, 121:3 – 21, 2005.
- [vdAWM04] W. v.d.Aalst, T. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *TKDE*, 16(9):1128–1142, 2004.

## *GI-Edition Lecture Notes in Informatics*

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühlhng, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheimer (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni\_ (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze – Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheimer, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömmme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolfried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web
- P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik
- P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)
- P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)
- P-69 Robert Hirschfeld, Ryszard Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODE 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006
- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-tern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS '06
- P-82 Heinrich C. Mayr, Ruth Brey (Hrsg.): Modellierung 2006
- P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics
- P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications
- P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems
- P-86 Robert Krimmer (Ed.): Electronic Voting 2006
- P-87 Max Mühlhäuser, Guido Röbling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik
- P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODE 2006, GSEM 2006
- P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur
- P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006
- P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006
- P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1
- P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2
- P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen
- P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies
- P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

- P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices
- P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik
- P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik
- P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten
- P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments
- P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)
- P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme
- P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007
- P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops
- P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications
- P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures
- P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1
- P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2
- P-111 Christian Eibl, Johannes Magenheimer, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik
- P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis
- P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1<sup>st</sup> Conference on Social Semantic Web (CSSW)
- P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3<sup>rd</sup> International Conference on IT-Incident Management & IT-Forensics
- P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007
- P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1<sup>st</sup> International Working Conference on Business Process and Services Computing BPSC 2007
- P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and management The 4<sup>th</sup> International Conference on Grid Service Engineering and Management GSEM 2007
- P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007
- P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications
- P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges
- P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik
- P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

- P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Poustchi, Klaus Turowski (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit  
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)
- P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)  
9<sup>th</sup> Workshop on Parallel Systems and Algorithms (PASA)  
Workshop of the GI/ITG Special Interest Groups PARS and PARVA
- P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)  
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde  
Referate der 28. GIL Jahrestagung
- P-126 Rainer Gimmich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)  
10<sup>th</sup> Workshop Software Reengineering (WSR 2008)
- P-127 Thomas Kühne, Wolfgang Reising, Friedrich Steimann (Hrsg.)  
Modellierung 2008
- P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)  
Sicherheit 2008  
Sicherheit, Schutz und Zuverlässigkeit  
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)  
2.-4. April 2008  
Saarbrücken, Germany
- P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)  
Sigsand-Europe 2008  
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems
- P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
1. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-131 Robert Krimmer, Rüdiger Grimm (Eds.)  
3<sup>rd</sup> International Conference on Electronic Voting 2008  
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)  
DeLFI 2008:  
Die 6. e-Learning Fachtagung Informatik
- P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)  
INFORMATIK 2008  
Beherrschbare Systeme – dank Informatik Band 1
- P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)  
INFORMATIK 2008  
Beherrschbare Systeme – dank Informatik Band 2
- P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)  
Didaktik der Informatik – Aktuelle Forschungsergebnisse
- P-136 Andreas Beyer, Michael Schroeder (Eds.)  
German Conference on Bioinformatics GCB 2008
- P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)  
BIOSIG 2008: Biometrics and Electronic Signatures
- P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)  
Synergien durch Integration und Informationslogistik  
Proceedings zur DW2008
- P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)  
Industrialisierung des Software-Managements  
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik
- P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)  
IMF 2008 - IT Incident Management & IT Forensics
- P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)  
Modellierung betrieblicher Informationssysteme (MobIS 2008)  
Modellierung zwischen SOA und Compliance Management
- P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)  
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung
- P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)  
Software Engineering 2009  
Fachtagung des GI-Fachbereichs Softwaretechnik



- P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)  
Datenbanksysteme in Business, Technologie und Web (BTW)
- P-145 Knut Hinkelmann, Holger Wache (Eds.)  
WM2009: 5th Conference on Professional Knowledge Management
- P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung  
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)
- P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)  
Business Process, Services Computing and Intelligent Service Management  
BPSC 2009 · ISM 2009 · YRW-MBP 2009
- P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)  
9<sup>th</sup> International Conference on Innovative Internet Community Systems  
I<sup>2</sup>CS 2009
- P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
2. DFN-Forum  
Kommunikationstechnologien  
Beiträge der Fachtagung
- P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)  
Software Engineering  
2009 - Workshopband
- P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)  
PRIMIUM  
Process Innovation for Enterprise Software
- P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)  
Enterprise Modelling and Information Systems Architectures  
Proceedings of the 3<sup>rd</sup> Int'l Workshop EMISA 2009
- P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)  
Lernen im Digitalen Zeitalter  
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik
- P-154 Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Hrsg.)  
INFORMATIK 2009  
Im Focus das Leben
- P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)  
BIOSIG 2009:  
Biometrics and Electronic Signatures  
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
- P-156 Bernhard Koerber (Hrsg.)  
Zukunft braucht Herkunft  
25 Jahre »INFOS – Informatik und Schule«
- P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)  
German Conference on Bioinformatics 2009
- P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)  
Precision Agriculture  
Reloaded – Informationsgestützte Landwirtschaft
- P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)  
Software Engineering 2010
- P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)  
Software Engineering 2010 –  
Workshopband  
(inkl. Doktorandensymposium)
- P-161 Gregor Engels, Dimitris Karagiannis, Heinrich C. Mayr (Hrsg.)  
Modellierung 2010
- P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)  
Vernetzte IT für einen effektiven Staat  
Gemeinsame Fachtagung  
Verwaltungsinformatik (FTVI) und  
Fachtagung Rechtsinformatik (FTRI) 2010
- P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme  
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

The titles can be purchased at:

**Köllen Druck + Verlag GmbH**

Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn

Fax: +49 (0)228/9898222

E-Mail: druckverlag@koellen.de