

Services Science

LNCS 6275

**Asit Dan**  
**Frédéric Gittler**  
**Farouk Toumani (Eds.)**

# Service-Oriented Computing

**ICSOC/ServiceWave 2009 Workshops**

**International Workshops, ICSOC/ServiceWave 2009**  
**Stockholm, Sweden, November 2009**  
**Revised Selected Papers**

 **Springer**

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison, UK

Josef Kittler, UK

Alfred Kobsa, USA

John C. Mitchell, USA

Oscar Nierstrasz, Switzerland

Bernhard Steffen, Germany

Demetri Terzopoulos, USA

Gerhard Weikum, Germany

Takeo Kanade, USA

Jon M. Kleinberg, USA

Friedemann Mattern, Switzerland

Moni Naor, Israel

C. Pandu Rangan, India

Madhu Sudan, USA

Doug Tygar, USA

## Services Science

Subline of Lectures Notes in Computer Science

### Subline Editors-in-Chief

Robert J.T. Morris, *IBM Research, USA*

Michael P. Papazoglou, *University of Tilburg, The Netherlands*

Darrell Williamson, *CSIRO, Sydney, Australia*

### Subline Editorial Board

Boualem Bentallah, Australia

Athman Bouguettaya, Australia

Murthy Devarakonda, USA

Carlo Ghezzi, Italy

Chi-Hung Chi, China

Hani Jamjoom, USA

Paul Klingt, The Netherlands

Ingolf Krueger, USA

Paul Maglio, USA

Christos Nikolaou, Greece

Klaus Pohl, Germany

Stefan Tai, Germany

Yuzuru Tanaka, Japan

Christopher Ward, USA

Asit Dan Frédéric Gittler  
Farouk Toumani (Eds.)

# Service-Oriented Computing

ICSOC/ServiceWave 2009 Workshops

International Workshops  
ICSOC/ServiceWave 2009  
Stockholm, Sweden, November 23-27, 2009  
Revised Selected Papers



Springer

Volume Editors

Asit Dan  
IBM Software Group  
Somers, NY, USA  
E-mail: asit@us.ibm.com

Frédéric Gittler  
HP Laboratories  
Grenoble, France  
E-mail: frederic.gittler@hp.com

Farouk Toumani  
Blaise Pascal University  
Clermont-Ferrand, France  
E-mail: farouk.toumani@univ-bpclermont.fr

Library of Congress Control Number: 2010934942

CR Subject Classification (1998): D.2, J.1, C.2, H.4, H.3, H.5

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743  
ISBN-10 3-642-16131-6 Springer Berlin Heidelberg New York  
ISBN-13 978-3-642-16131-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2010  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 06/3180 5 4 3 2 1 0

# Preface

This volume contains the proceedings of the scientific workshops that were held in conjunction with the 2009 International Conference on Service Oriented Computing/ServiceWave, held on November 24–25, 2009 in Stockholm, Sweden. Such satellite events traditionally play a key role in stimulating active exchange and interaction related to the conference topics.

This year, the scientific program was particularly rich and addressed various challenging research issues. The selected scientific workshops were organized around the following three main tracks:

**Business Models and Architectures Track.** The Business Models and Architectures Track Focused on the overall modern enterprise. The ability to react quickly to ongoing changes in the marketplace or customer requirements is one of the biggest challenges facing businesses. The three workshops in this track addressed different, yet complementing, facets of the problem. TEAR focused on aligning the enterprise architecture with its business models: adapting the IT infrastructure and changing applications so that they optimally support the new business needs. GLOBALIZATION (SG-PAW) looked at enacting the new business processes by encapsulating organizational work as services that can be combined in new ways, optimizing end-to-end operations across geographical, organizational, and cultural boundaries. Finally, SOC-LOG focused on addressing the challenges of a specific application domain, namely logistics, through developing SOC-based solutions and examining aspects of knowledge management, while bringing together researchers from different, though overlapping, areas (logistics/supply chain management and service-oriented computing/systems).

**Service Quality and Service Level Agreements Track.** Ensuring quality poses new challenges to service engineering, delivery, and evolution. This track included two workshops that approach this challenge from two complementary perspectives. The first workshop, NFPSLAM-SOC, focused on research problems around models, concepts, languages, and methodologies that enable the specifications of non-functional properties and service level agreements in the context of service-oriented computing, with a special focus on transparent, multi-level, and holistic NFP and SLA management of service-oriented systems. The workshop MONA+ concentrated on the problems related to the monitoring and adaptation mechanisms and strategies, with a special focus on the relations and interdependencies between the network layer and the service layer.

**Service Engineering Track.** With the wide adoption of SOA, there is an increasing need for comprehensive engineering principles, methodologies, and tools to support the entire software development lifecycle of service-oriented applications. This track included two workshops addressing this challenge.

WESOA 2010 focused on specific aspects of Software Service Engineering (SSE) and aimed at facilitating the exchange and evolution of ideas on SSE topics across multiple disciplines. The UGS 2009 workshop focused on user-centric approaches. It explored research and development approaches which empower end-users to participate in the generation, combination, and adaption of services to create functionality and solve problems in their work.

June 2010

Asit Dan  
Frédéric Gittler  
Farouk Toumani  
Workshops Chairs  
ICSOC/ServiceWave 2009

# Introduction to the 4<sup>th</sup> Workshop on Trends in Enterprise Architecture Research (TEAR 2009)

Stephan Aier<sup>1</sup>, Joachim Schelp<sup>1</sup>, and Marten Schönherr<sup>2</sup>

<sup>1</sup>Institute of Information Management, University of St. Gallen  
Müller-Friedberg-Strasse 8  
9000 St. Gallen, Switzerland

{stephan.aier, joachim.schelp}@unisg.ch

<sup>2</sup>Deutsche Telekom Laboratories

Ernst-Reuter-Platz 7

10587 Berlin, Germany

marten.schoenherr@telekom.de

**Abstract.** On Nov 23<sup>rd</sup>, 2009 the 4<sup>th</sup> Workshop on Trends in Enterprise Architecture Research (TEAR2009) is held in the “Business Models and Architecture” workshop track of the 7<sup>th</sup> International Conference on Service Oriented Computing (ICSOC 2009), which takes place Nov. 24–27, 2009.

## 1 Introduction

Although the field of enterprise architecture (EA) has gained more and more attention in the previous couple of years, it is still immature. The understanding of the term enterprise architecture is diverse in both practitioner and scientific communities. Regarding the term architecture, most agree on the ANSI/IEEE Standard 1471-2000, where architecture is defined as the “fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.” For enterprise architecture the focus is on the overall enterprise. In contrast to traditional architecture management approaches like IT architecture, software architecture, or IS architecture, EA explicitly incorporates “pure” business-related artifacts in addition to traditional IS/IT artifacts.

EA is important because organizations need to adapt increasingly quickly to changing customer requirements and business goals. This need influences the entire chain of activities of an enterprise, from business processes to IT support. Moreover, a change in a particular architecture may influence other architectures. For example, when a new product is introduced, business processes for production, sales, and after-sales need to be adapted. It might be necessary to change applications, or even adapt the IT infrastructure. Each of these fields will have its own architectures. To keep the enterprise architecture coherent and aligned with the business goals, the relations between these different architec-

tures must be explicit, and a change should be carried through methodically in all architectures.

In previous years the emergence of service-oriented design paradigms (e.g., service-oriented architecture, SOA) contributed to the relevance of EA. The need to design services along business processes forced companies to pay more attention to business architectures. The growing complexity of existing application landscapes led to increased attention to application architectures at the same time. To better align business and IS architectures a number of major companies started to establish EA efforts after introducing SOAs.

Until recently, practitioners, consulting firms and tool vendors have been leading in the development of the EA discipline. Research on EA has been taking place in relatively isolated communities. The main objective of this workshop series is to bring these different communities of EA researchers together and to identify future directions for EA research, with special focus on service-oriented paradigms. An important question in that respect is what EA researchers should do, as opposed to EA practitioners.

## 2 Contributions

Building on its great success in previous years, the 4<sup>th</sup> Workshop on Trends in Enterprise Architecture Research (TEAR 2009) is held in the “Business Models and Architecture” workshop track of the 7<sup>th</sup> International Conference on Service Oriented Computing (ICSOC 2009) in Stockholm on November 23, 2009. The TEAR 2009 call for papers attracted 15 submissions. A total of 7 papers passed the review process successfully, resulting in a 46.6% acceptance rate.

Accepted papers reflect the developments in the field of enterprise architecture as outlined in the introduction.

The papers of the first session highlight current discussions on future directions for EA as a research field. First, Sabine Buckl, Florian Matthes, and Christian Schweda give an overview on recent EA management approaches with their paper “Future Research Topics in Enterprise Architecture Management—A Knowledge Management Perspective.” They take this perspective to identify gaps in current approaches and propose future research topics for the area of EA management. Dirk Stelzer takes a different approach in his review of current approaches and focuses on “Enterprise Architecture Principles: Literature Review and Research Directions.” In his analysis he shows that business principles, IT principles, and EA principles are often mixed up and that research into generic design principles is still in its infancy. He concludes with conceptual foundations and guidance for further research in this field. Finally Sebastian Klöckner and Dominik Birkmeier employ a third perspective in their analysis “Something is Missing: Enterprise Architecture from a Systems Theory Perspective.” In their paper they interpret enterprises as socio-technical systems and therefore choose a systems theory perspective. The authors analyze which features and aspects are necessary for a comprehensive model. They show that especially human actors, as the most flexible and agile elements of enterprises, are not adequately included



in current architectures, presenting the first ideas for integrating this aspect into EA. They also show the corresponding implications of such an inclusion, as well as several areas of further research.

The papers in the second session concentrate on current practices in EA in enterprises, especially in implementing EA. First, Joachim Schelp and Stephan Aier present “A Reassessment of Enterprise Architecture Implementation.” After a summary of the current state of the art in research and practice, they ask why EA seems to be successful in some organizations while it is not in others that also have notations, models, methods, and even dedicated EA tools. In order to understand these differences, they analyze the development of EA in six companies over the last eight years and show that apart from formal structure and processes (1) training and education of architects and non-architects, (2) improving architects’ communication skills, (3) intensifying EA representation in projects, and (4) tool support (not replacements with tools), can significantly contribute to long-term EA success. Then Marlies Van Steenberghe, Jurjen Schipper, Rik Bos, and Sjaak Brinkkemper present “The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement” as an instrument to improve the effectiveness of EA teams. In the past this approach has been applied to many architecture practices to assess their architecture maturity level. They present an analysis of these assessments and give an overview of common strengths and weaknesses in current architecture practices. Finally, Steffen Kruse, Jan Stefan Addicks, Matthias Postina, and Ulrike Steffens focus on EA tool aspects with their paper “Decoupling Models and Visualisations for Practical EA Tooling.” While customized modeling solutions provide scalability, adaptability, and flexibility, they are often in conflict with generic or reusable visualizations. The authors present an approach to augment customized modeling with the techniques of model transformations and higher-order transformations to provide flexible and adaptable visualizations with a minimum of requirements for the underlying enterprise models. They detail their approach with a proof-of-concept implementation, showing how a decoupling can ease EAM approaches and provide appropriate tooling in practice.

The final paper, which is placed in the joint session with the SIG-PAW workshop on service-oriented architectures, is presented by André Miede, Nedislav Nedyalkov, Dieter Schuller, Nicolas Repp, and Ralf Steinmetz. Their paper “Cross-organizational Security—The Service-Oriented Differences” focuses on questions of security architectures in cross-organizational collaboration scenarios. While there is an active research community for SOA security, common literature on the topic has not yet identified the influence of the SOA paradigm on security aspects in a structured manner, especially in an enterprise context. In their paper, they contribute to this goal by identifying the main elements of cross-organizational SOA in the form of a conceptual model and by discussing these elements regarding their impact on security issues. Based on this, they define and structure important research challenges for SOA security. All contributions reflect that EA as a research field is still in its infancy. However, it is evolving and the maturity of the individual research approaches is intriguing.

The contributions show that a single research approach to EA would not be sufficient, but that the diversity of research approaches is a key to investigate in the variety of aspects in this interdisciplinary topic—for the benefit of both research and practice.

### 3 Program Committee

The Program Committee members and reviewers each deserve credit for the excellent final program that resulted from the diligent review of the submissions. The organizing members of the program committee would like to thank all of them and especially all authors submitting contributions to the TEAR workshop series.

#### Members of the Program Committee

Stephan Aier	University of St. Gallen, Switzerland (organizer)
Antonia Albani	Delft University of Technology, The Netherlands
Giuseppe Berio	University of South Brittany, France
Scott Bernard	Carnegie Mellon University, Syracuse University, USA
Udo Bub	Deutsche Telekom Laboratories, Germany
Haluk Demirkan	Arizona State University, USA
Andreas Dietzsch	PostFinance, Bern, Switzerland
Mathias Ekstedt	KTH, Sweden
Ulrich Frank	University of Duisburg-Essen, Germany
Matthias Goeken	Frankfurt School of Finance and Management, Germany
Norbert Gronau	University of Potsdam, Germany
Wilhelm Hasselbring	University of Kiel, Germany
Pontus Johnson	KTH, Sweden (TEAR standing committee)
Dimitris Karagiannis	University of Vienna, Austria
Wolfgang Keller	Objectarchitects, Germany
Marc Lankhorst	Novay, The Netherlands (TEAR standing committee)
Florian Matthes	TU Munich, Germany
Michael zur Mühlen	Stevens Institute of Technology, USA
Tim O'Neill	Sydney University of Technology, Australia
Erik Proper	Radboud University Nijmegen, The Netherlands
Gerold Riempp	European Business School (EBS), Germany
Michael Rosemann	QUT, Australia
Joachim Schelp	University of St. Gallen, Switzerland (organizer)
Marten Schönherr	Deutsche Telekom Laboratories, Germany (organizer)
Gerhard Schwabe	University of Zurich, Switzerland
Elmar J. Sinz	University of Bamberg, Germany
Pedro Sousa	Lisbon Technical University (IST), Link, Portugal
Ulrike Steffens	Offis, Oldenburg, Germany
José Tribolet	University of Lisbon, Portugal
Martin Zelm	CIMOSA, Germany

Stephan Aier, Joachim Schelp, Marten Schönherr

# First International Workshop on SOA, Globalization, People, and Work (SG-PAW): Current State of Affairs and Future Research Directions

Daniel Oppenheim<sup>1</sup>, Francisco Curbera<sup>1</sup>, and Marcelo Cataldo<sup>2</sup>

<sup>1</sup> IBM T.J. Watson Research, Hawthorne, NY 10532, USA  
{music, curbera}@us.ibm.com

<sup>2</sup> Carnegie Mellon University, Pittsburgh, PA 15213, USA  
mcataldo@cs.cmu.edu

## Introduction

On November 23, 2009, the First International Workshop on SOA, Globalization, People and Work (SG-PAW) was held as part of the 7th International Conference on Service Oriented Computing (ICSOC 2009) in Stockholm, Sweden. The workshop focused on the problem of enabling an enterprise to leverage internal and external global services and combine them in new ways that optimize its end-to-end operations. The premise is that the SOA methodology is well suited to address this problem by encapsulating organizational work as services that can cross geographical, organizational, and cultural boundaries. The goal was to combine academics and practitioners to identify together core issues, research challenges, learn from successful attempts or approaches, and propose new formalisms, models, architectures, frameworks, methodologies, or approaches.

This workshop was the second of three half-day workshops in the Business Models and Architecture track. It was preceded by the 4th International Workshop on Trends in Enterprise Architecture Research (TEAR 2009) and followed by the First International Workshop on Service Oriented Computing in Logistics (SOC-LOG 2009). The organizers and many participants of both TEAR and SOC-LOG also attended this workshop, creating a valuable atmosphere for discussion with a broad context. Five papers were presented. This workshop ended in a collaborative discussion that included the organizers and participants from all three workshops. This discussion was facilitated by Richard Hull, and its output is presented in the workshop manifesto below.

## Workshop Manifesto

The following list summarizes the key findings and recommendations made by this group.

## Findings

1. **Process.** There is a fundamental tension between routine and free-form that stems from the desire for standardization, consistency and repeatability vs. the need to continuously evolve and adapt. Standardized processes can be executed routinely, yield predictable results, and lend themselves well to machine-automation. Examples of routine processes include using an ATM to withdraw money from a bank, or requesting a loan from a financial institution. But when unpredictable things happen, there can be a need for almost free-form agility to respond and adapt. This need to quickly modify how things are done occurs frequently in large projects, such as collaborative development of an airplane or a complex software system. Currently, adaptation is handled primarily through human creativity, expertise, and ability to improvise; the failure rate of such projects is very high. There is a need for a framework that would support both routine and free-form, not only during the process design, but especially during execution. The need for process flexibility during runtime also blurs the current separation between the activities of designing a process and performing a task, as both activities become an integral part of doing work.
2. **People.** The need for people as a requirement to ensure effective execution of enterprise processes is not sufficiently understood. Current process definitions address people by specifying roles that are required to execute a task. This approach considers people merely as resources that could, in theory, be replaced by automation. However, some types of processes must rely on people to execute effectively. Humans may be required for a variety of reasons, including to assess the complexity of the domain; to drive recovery when unexpected things happen; to resolve issues at runtime; or to negotiate and coordinate work across enterprise boundaries. Most business process management (BPM) frameworks do not address these different roles of people; nor do they adequately support teamwork around tasks or the creation and execution of dynamic service plans. There is no model of the human system that identifies the different types of actors, teams, or organizations, and that can be used to bring this together with current BPM practices.
3. **Globalization.** Globalization creates an ever-growing abundance of resources, innovation, and specialization. In order for businesses to harness those potential benefits, they require flexible frameworks into which they can plug-and-play relevant entities such as partners, suppliers, service providers, or resources.
4. **Models.** There is a need for capability-oriented models and languages that can address both the routine and the free-form in a uniform way. Beyond providing a well-defined starting point for enterprise-work, they will also provide a formalism that enables ongoing evolution and adaptation in response to new needs or unpredictable events, and do so in a way that can be supported by machines. Such models will have to address many elements of the problems, including business, process, data, IT, people, resources, and organization. Current disciplines tend to focus on a limited subset of these as-

pects; the challenge is to bring them together. BPM, for example, does not adequately support teamwork around tasks or the creation and execution of dynamic service plans. Enterprise architecture (EA) models use a layered approach to bridge the gap between the business and IT that does not adequately consider the role of people, process, or organization. Computer-supported collaborative work (CSCW) focuses on people, awareness, and distributed collaboration to enable cooperative work, but does not adequately connect this with process, data, or organization. Services-oriented computing (SOC) tends to focus on composable bite-size processes that can be executed by machines, but does not provide the flexibility required to scale and support complex cross-organizational work. BPEL4People and similar standards do not address the full scope of cross-enterprise work or the complex needs of humans in their various roles.

### Recommended Research Areas

1. Exploring the new capabilities enabled by human flexibility, creativity, and communication patterns and integrating them systematically into BPM, SOA/SOC, EA, and CSCW.
2. A framework that allows us to understand the trade-offs between automated vs. free-form approaches, what should be done by machine and what by human, where flexibility and creativity is required vs. where not, and how to set up or re-engineer an enterprise with these tradeoffs in mind.
3. A model or theory of non-functional characteristics of people work, such as trust, reputation, or quality. What they are and how to measure them. This will be analogous to non-functional characteristics of SOA services or hardware components.
4. How to ease the understandability, use, and communication of knowledge-rich processes, operations, and services. This relates not only to pre-designed processes but also to dynamically created and/or customized business processes, and should enable non-IT people to take advantage of free-form processes.
5. Models of people, teams, and organizations. This will identify the different types of roles people play in cross-organizational work. It will also focus on issues specific to people, such as incentives, accountability, authority, trust, collaboration, productivity, or quality of output.
6. Relationship between process design and execution, especially when ongoing adaptation and transformation is required.
7. Models, methods, formalisms, and languages that focus on the role people play in the control and coordination of cross-enterprise collaboration in different domains.
8. Dynamic flow engines that can support such models and provide the flexibility required for runtime adaptation and evolution.
9. Adaptation, versioning, and evolution of processes, work, collaborating organizations and collaboration patterns.

10. Extending SOA formalisms and constructs to facilitate the definition, dispatch, and orchestration of work as services that can be carried out by and for organizations.
11. Context, data, and knowledge management as required for managing and coordinating work across organizations and their interrelationship with the domain data, tools, and processes.
12. IT, middleware, systems, tools, and frameworks that support cross-enterprise collaboration, and their relationship with current enterprise or domain-specific tools and IT.
13. Utilization of crowd sourcing and social computing paradigms for the coordination and/or execution of work and business processes that span across organizational boundaries.

# Introduction to the First International Workshop on Service-Oriented Computing in Logistics (SOC-LOG)

Joerg Leukel<sup>1</sup>, André Ludwig<sup>2</sup>, and Alex Norta<sup>3</sup>

<sup>1</sup>University of Hohenheim, Germany

<sup>2</sup>University of Leipzig, Germany

<sup>3</sup>University of Helsinki, Finland

Logistics is of paramount importance for many industries: it plans and realizes the flow of goods from sources to destinations by means of transformations in space, time, and quantity. While existing logistics IT systems provide solid support for static, self-contained logistics systems, research on managing the logistics in supply chains, which are dynamically changing, is less advanced. Service-oriented computing (SOC) is a promising paradigm, which automates inter-organizational processes by loosely coupled software-based services. With the set of design principles, architectural models, concepts, and — last but not least — with its existing and growing set of standards, SOC promotes the adaptiveness of logistics systems and supply chains, a flexible and re-configurable provisioning along multiple supply chains, and their efficiency. The purpose of the First International Workshop on Service-Oriented Computing in Logistics (SOC-LOG 2009) is to present and discuss recent significant developments at the intersection of SOC and logistics systems/supply chain management, and to promote cross-fertilization and an exchange of ideas and techniques between these fields. The relation to ICSOC 2009 is that, on one hand, the conference addresses the core concepts such as interacting business processes, service composition, service operations, and quality of services, and on the other hand, would receive feedback, experiences, and requirements from a highly relevant application domain to validate and advance its current approaches. The focus of this workshop is the study and exploration of SOC's potential to solve coordination problems in logistics systems and supply chains. Specifically, open issues are related to, e.g., service description languages; discovery, composition and coordination of logistics services; negotiation and management of service-level agreements for logistics-service delivery; measuring the efficiency and effectiveness of logistics services. All submissions received were single-blind peer reviewed by at least two members of the international program committee. In total, we received nine submissions from five countries. Based on the review reports, we accepted five papers, an acceptance rate of 55.6%. We would like to thank the program committee members and authors for all of their hard work and participation in the lively workshop. We hope that SOC-LOG will help with the exchanging new ideas and with the networking and sharing of ideas. More information on SOC-LOG 2009 is available at <http://soclog09.wifa.uni-leipzig.de/>.

# Introduction to the Third Workshop on Non-functional Properties and Service Level Agreements Management in Service-Oriented Computing (NFPSLAM-SOC 2009)

Flavio De Paoli<sup>1</sup>, Ioan Toma<sup>2</sup>, Hui Li<sup>3</sup>, Wolfgang Theilmann<sup>4</sup>, Marcel Tilly<sup>5</sup>,  
Andrea Maurino<sup>1</sup>, and Ramin Yahyapour<sup>6</sup>

<sup>1</sup> Università di Milano-Bicocca  
`{depaoli, maurino}@disco.unimib.it`  
<sup>2</sup> STI Innsbruck, University of Innsbruck  
`ioan.toma@sti2.at`  
<sup>3</sup> SAP Research, CEC Karlsruhe,  
`hui01.11@sap.com`  
<sup>4</sup> SAP Research, CEC Karlsruhe,  
`wolfgang.theilmann@sap.com`  
<sup>5</sup> European Microsoft Innovation Center,  
`marcel.tilly@microsoft.com`  
<sup>6</sup> Dortmund University of Technology,  
`ramin.yahyapour@udo.edu`

The Third Workshop on Non-functional Properties (NFPs) and Service Level Agreements Management (SLAM) in Service-Oriented Computing was held on November 23, 2007 in Stockholm, Sweden in conjunction with the International Conference on Service Oriented Computing. The first edition of the workshop was organized at the ICSOC 2007, followed by the second edition at ECOWS 2008. The workshops constituted a series of successful forums, each with more than 30 participants and 12 paper presentations. The workshop aimed to bring together researchers and industry practitioners to foster a greater understanding of how the management of NFP and SLAs can assist businesses utilizing service-oriented architectures (SOA) as well as to investigate the resulting research issues. These issues were felt to be highly relevant due to the increasing popularity of SOA and the fact that while the foundations of SOA functionality are now well understood, non-functional properties are not. The workshop keynote was given by Paolo Traverso, whose talk, “From Software Services to a Future Internet of Services” provided a context for much of the work presented in the workshop papers. The talk focused on the core role of real services,” in the Future Internet and the paradigm shift required to model, monitor, adapt, and compose such services. The talk has generated interesting discussions on what services are and how their properties, especially non-functional ones, are crucial to support future scenarios with a next generation of technological platforms. High-quality papers were submitted to the workshop, allowing nine papers to be accepted. These were arranged into two broad themes: Service Level Agreements (SLAs) and NFPs in service-related tasks. In the first theme, six papers were presented.



The paper “A Framework for Multi-level SLA Management” proposes a technical architecture for a multi-level SLA management framework. The core concepts of the framework include four different roles, three layers of business, software and infrastructure management, a service life-cycle model, and the conceptualization of basic data store and functional flows. The framework and architecture are evaluated on an open reference case supporting a retail chain scenario. The paper “Runtime Prediction of Service Level Agreement Violations for Composite Services” proposes an approach for predicting SLA violations at runtime, which uses measured and estimated facts (instance data of the composition or QoS of used services) as the input for a prediction model. The prediction model is based on machine learning regression techniques, and trained using historical process instances. The third contribution, “Using SLA Mapping to Increase Market Liquidity,” discusses a solution that derives SLA templates from a large number of heterogeneous SLAs in the market, and, by using these templates instead of the original SLAs, facilitates SLA mapping (i.e., mapping of offers to requests). The approach is validated through simulation and comparison with alternative approaches in which SLAs are predefined. The paper “Translation of Service Level Agreements: A Generic Problem Definition” explores the dependencies between different SLAs, and formalizes the problem of converting an agreement for a composed service into individual agreements for the services from which it is composed. In “On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management,” the authors advocate the use of compliance governance dashboards. The paper points out the major issues in this domain, identifies the concepts and models that underlie the problem, and addresses how IT can effectively support compliance analysis in SOAs. Finally, the paper “EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications” presents an interesting study on the performance of small instances in Amazon EC2. The authors show that the performance of virtual instances is relatively stable over time with fluctuations of mean response time. They also show that different, supposedly identical instances often have very different performance.

Issues on management of NFPs in service-related tasks have been discussed in three different papers. The first paper: “Transformation of Intermediate Nonfunctional Properties for Automatic Service Composition,” proposes a transformation technique for automatic composition that identifies binding information in the selection stage from intermediate abstract NFPs. The classification of abstraction level in NFPs, a model to define abstract and concrete NFPs, and an algorithm for transformation from intermediate to concrete level are also presented. The paper “Dealing with Fixable and Non-fixable Properties in Service Matchmaking” presents a matchmaking approach under bounded uncertainty implemented using constraint programming. The matchmaking approach is transformed into a quantified constraint satisfaction problem. Finally, the paper “Ontology-Based Feature Aggregation for Multi-valued Ranking” focuses on the ranking of discovered Web services, proposing a novel approach based on non-functional properties of services: information that is available about services by analyzing their

description that is available on the Web, their hyperlink relations, monitoring information, etc. The approach is making use of semantic technologies, aggregating the various real-world service aspects as described above in a unified model and providing different rank values based on those aspects. The organizers would especially wish to thank the people who made NFPSLAM-SOC 2009 successful. First of all, Paolo Traverso who provided stimulating insights. Then the program committee members and the additional reviewers for their work that ensured the high-quality of accepted contributions. A special thanks to ICSOC Chairs that allowed and supported us for realizing the Third edition of NFPSLAM-SOC, and finally, all the authors and participants for providing the content of the workshop.

# Introduction to the Second International Workshop on Service Monitoring, Adaptation and Beyond (MONA+)

Ranganai Chaparadza<sup>1</sup>, Dimka Karastoyanova<sup>2</sup>, Raman Kazhamiakin<sup>3</sup>,  
Andreas Metzger<sup>4</sup>, and Marco Pistore<sup>5</sup>

<sup>1</sup> Fraunhofer FOKUS, Germany,

<sup>2</sup> IAAS, University of Stuttgart, Germany,

<sup>3</sup> FBK-IRST, Trento, Italy,

<sup>4</sup> Paluno, University of Duisburg-Essen, Germany,

<sup>5</sup> FBK-IRST, Trento, Italy

Advances in modern technology and the constantly evolving requirements implied by dynamic business and operational environments impose new challenges for engineering and provisioning service-based applications (SBAs). SBAs have to become drastically more flexible; they should be able to operate and evolve in highly dynamic environments and to adequately identify and react to various changes in these environments. In such a setting, adaptation becomes a key capability as it enables SBAs to continuously change themselves to satisfy new requirements and demands. The ability of the SBA to adapt relies on the presence of monitoring mechanisms to identify, detect, and even predict critical events and situations. A variety of approaches and techniques addressing different forms of monitoring and adaptation have been proposed to date. Still, for delivering robust, dependable, and highly adaptable SBAs, the definition of holistic approaches is crucial. This requires the integration of the efforts of researchers from various disciplines and research areas. More specifically, this requires the integration across the different layers of an SBA, including the business layer, the service composition and coordination layer, the infrastructure layer, and the network layer. In addition, different competences, such as requirements engineering, design, quality assurance, realization, and management need to be brought together to devise the required holistic approaches. The main objectives of MONA+ 2009 were to bring together researchers from the different communities working on SBA monitoring and adaptation, and to start identifying shared research challenges towards developing comprehensive holistic approaches for multi-layer monitoring and cooperative adaptation techniques across the layers involved while taking into account different types of triggers to adaptation, ranging from faults, changes in goals, policies and context of operation, etc. The special focus for this second edition of the workshop was on the relations and interdependencies between the network and the service layer. Specifically the workshop addressed how the monitoring and adaptation mechanisms provided at those two layers can better interoperate, and how to better support an integrated design and management of monitoring and adaptation across those two layers. The proceedings of the workshop provide a rich

collection of high-quality papers, thanks to the authors and to the over 50 participants of the workshop, who provided valuable feedback. The papers address a number of relevant research challenges, from which the community at large can benefit towards developing holistic approaches for multi-layer monitoring and cooperative adaptation techniques. Frameworks for designing and embedding autonomic principles of operation of services and network functions are also provided.

The MONA+ 2009 Organizers were: Ranganai Chaparadza (Fraunhofer FOKUS, Germany; acting as PC Chair), Dimka Karastoyanova (IAAS, University of Stuttgart, Germany), Raman Kazhamiakin (FBK-IRST, Trento, Italy), Andreas Metzger (Paluno, University of Duisburg-Essen, Germany; acting as PC Chair), Marco Pistore (FBK-IRST, Trento, Italy).

# 5th International Workshop on Engineering Service-Oriented Applications (WESOA 2009)

Christian Zirpins<sup>1</sup>, George Feuerlicht<sup>2,3</sup>, Winfried Lamersdorf<sup>4</sup>, Guadalupe Ortiz<sup>5</sup>, Yen-Jao Chung<sup>6</sup>, Wolfgang Emmerich<sup>7</sup>, and Robert Johnson<sup>8</sup>

<sup>1</sup> Karlsruhe Institute of Technology,  
Christian.Zirpins@kit.edu

<sup>2</sup> Prague University of Economics,  
jirif@vse.cz

<sup>3</sup> University of Technology,  
Sydney, jiri@it.uts.edu.au

<sup>4</sup> University of Hamburg,  
lamersdorf@informatik.uni-hamburg.de

<sup>5</sup> University of Extremadura,  
gobellot@unex.es,

<sup>6</sup> IBM Research,  
JYChung@us.ibm.com

<sup>7</sup> University College London,  
W.Emmerich@cs.ucl.ac.uk

<sup>8</sup> IBM SoftwareGroup,  
robertdj@us.ibm.com

## Workshop Goals and Contents

The availability of comprehensive methodologies and tools based on sound software engineering principles is of critical importance to practitioners involved in developing service-oriented applications. Limitations of traditional software engineering approaches have led to the emergence of software service engineering (SSE) as a new research discipline, but this area is still immature with many remaining open issues. Service-oriented applications tend to be process-driven, loosely coupled, composed from autonomous services and influenced by diverse socio-economic contexts. Such applications need to provide multiple, flexible and sometimes situational interaction channels within and beyond organizational structures and processes. In many cases, service-oriented applications represent transactions of dynamic, process-driven business networks and drive interaction protocols between fluid configurations of autonomous service providers. In other cases, service-oriented applications are used in the context of social communities, where they are created by a large number of participants for very specific or even situational needs. In such domains it is not enough to focus on complex distributed software systems alone, but it is necessary to consider a broader socio-technical perspective. Engineering of such software systems requires continuous, collaborative and cross-disciplinary development processes, methodologies and tools that synchronize multiple software development lifecycles (SDLCs) of

various SOA artifacts. It is the challenge of service systems engineering to not only cope with these circumstances but to capitalize on them with radically new approaches. There is an urgent need for research community and industry practitioners to agree on comprehensive engineering principles, methodologies and tool support for the entire SDLC of service-oriented applications. The WESOA series of workshops provides an annual forum for SSE researchers and practitioners and facilitates exchange and evolution of ideas across multiple disciplines. The 5th WESOA event was held in Stockholm on November 23, 2009. The workshop started with a keynote presentation by Hugo Brand of Oracle on the convergence and unification of SOA, EDA and BPM concepts, giving an industry perspective on SOA standardization. The technical sessions consisted of eight high-quality papers representing a rich variety of topics revolving around principles, methods and application domains of SSE. A number of authors addressed various aspects of service variability including work on design for adaptation of service-based applications by Bucchiarone et al., a conceptual framework for legacy-to-SOA migration by Razavian et al., the MINERVA framework for continuous business processes improvement by Delgado et al., and work on service customization by variability modeling by Stollberg et al. Another focus was on runtime aspects of service-oriented applications. This included work on encapsulating Web forms as Web services by Vogel et al., adapter patterns for resolving mismatches in service discovery by Hyun Jung La et al. and work on runtime migration of WS-BPEL processes by Zaplata et al. The technical sessions concluded with work on quality models for choreography by Mancioffi et al. The workshop concluded with a discussion about the fundamental principles of SSE and the issues raised by individual presentations.

## Workshop Organization

WESOA 2009 was organized by an international group of researchers comprising the authors of this article. The event would not have been possible without the contribution of the program committee. Our thanks go to the following experts: Sudhir Agarwal (KIT Karlsruhe), Marco Aiello (Univ. of Groningen), Sami Bhiri (DERI Galway), Vincenzo D'andrea (Univ. of Trento), Florian Daniel (Univ. of Trento), Valeria de Castro (Univ. Rey Juan Carlos), Gregorio Diaz (Univ. of Castilla La Mancha), Schahram Dustdar (Technical Univ. Vienna), Keith Duddy (Queensland Univ. of Technology), Howard Foster (Imperial College London), Paul Greenfield (CSIRO), Peng Han (Fernuniv. Hagen), Birgit Hofreiter (Hochschule Lichtenstein), Dimka Karastoyanova (Univ. of Stuttgart), Rannia Khalaf (IBM Research), Axel Kieninger (KIT Karlsruhe), Agnes Koschmieder (KIT Karlsruhe), Heiko Ludwig (IBM Research), Leszek Maciaszek (Macquarie Univ.), Tiziana Margaria (Univ. of Potsdam), E. Michael Maximilien (IBM Research), Massimo Mecella (Univ. Roma LA SAPIENZA), Sooksathit Meesathit (Sakon Nakhon Rajabhat Univ.), Vojtech Merunka (Czech Univ. of Agriculture), Daniel Moldt (Univ. of Hamburg), Martin Molhanec (Czech Technical Univ. in Prague), Cesare Pautasso (Univ. of Lugano), Greg Pavlik (Oracle), Tomas

Pitner (Masaryk Univ.), Pierluigi Plebani (Politecnico di Milano), Franco Raimondi (Univ. College London), Thomas Risse (L3S Research Center), Norbert Ritter (Univ. of Hamburg), Colette Rolland (Univ. of Paris), Stefan Tai (KIT Karlsruhe), Willem-Jan van den Heuvel (Tilburg Univ.), Christian Werner (Univ. of Luebeck), Olaf Zimmermann (IBM Research Zurich). We would also like to thank the ICSOC organizers and especially the workshop chairs Asit Dan and Frédéric Gittler. We acknowledge the support of the Grant Agency of the Czech Republic: grant No. GAČR 201/07/0455/ and the Research Centre for Human Centred Technology Design at the University of Technology, Sydney.

# Introduction to the First International Workshop on User-Generated Services (UGS 2009)

Schahram Dustdar<sup>1</sup>, Manfred Hauswirth<sup>2</sup>, Juan José (Juanjo) Hierr<sup>3</sup>, Knud Möller<sup>2</sup>, Ismael Rivera<sup>2</sup>, Javier Soriano<sup>3</sup>, and Florian Urmetzer<sup>4</sup>

<sup>1</sup> TU Vienna, Austria

<sup>2</sup> DERI/NUI Galway, Ireland

<sup>3</sup> Telefónica I+D, Spain

<sup>4</sup> UPM, Spain

<sup>5</sup> SAP, Switzerland

Service-oriented architectures (SOA) have transformed the way software systems are being developed. However, the development of services is still service-centric rather than user-centric. The reuse and combination of such services requires the assistance of a skilled developer. To address this, UGS 2009 set out to explore research and development empowering end-users to participate in the generation, combination and adaptation of services to create functionality and solve problems in their work. This development in the service area mirrors that of *user-generated content* (UGC), which has become a major source of information on the World-Wide Web. Wikis, blogs, Web-based user forums and social networks have empowered end-users to collaboratively create content and share it. UGC is not only a phenomenon in the private domain, but has become a major source for technical solutions as well, as exemplified by the results of Web searches for technical problems: solutions are now increasingly found in sites providing UGC. Thus end-users have become a major source of knowledge, similarly leveraging the “resources at the edge of the network” as P2P systems have done on a technical level. The next logical step is that after supporting the creation and management of data, the same should be done at the level of services created and provided by end-users, i.e., *user-generated services* (UGS). UGS can cover a range of services, from ad-hoc, situational applications for personal use to more advanced enterprise mash-ups supporting a community of users. In order to facilitate UGS, tools and infrastructures to create, combine, reuse and execute possibly complex services in an easy manner are needed.

A range of issues have to be addressed in order to realize the vision of UGS. With the goal of tackling these issues and establishing a community around the topic of user-generated services, UGS 2009 brought together researchers and developers from both academia and industry, presenting and discussing a diversity of topics ranging from a user-study investigating the feasibility of UGS in general (Namoune et al.), front-ends for the visualization and composition of services (Gilles et al., Nestler et al., Silva et al.), over specific service domains such as personal information management (Grebner), the community-based



annotation of services (Loutas et al.) to problems of ontology mediation required in supporting end-users in combining services (Ambrus et al.). All papers presented here are based on the preliminary online proceedings made available at <http://CEUR-WS.org/Vol-540/>.

# Table of Contents

## Track 1: Business Models and Architecture

### The 4th Workshop on Trends in Enterprise Architecture Research (TEAR 2009)

Future Research Topics in Enterprise Architecture Management – A Knowledge Management Perspective . . . . .	1
<i>Sabine Buckl, Florian Matthes, and Christian M. Schweda</i>	
Enterprise Architecture Principles: Literature Review and Research Directions . . . . .	12
<i>Dirk Stelzer</i>	
Something Is Missing: Enterprise Architecture from a Systems Theory Perspective . . . . .	22
<i>Sebastian Kloeckner and Dominik Birkmeier</i>	
A Reassessment of Enterprise Architecture Implementation . . . . .	35
<i>Stephan Aier and Joachim Schelp</i>	
The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement . . . . .	48
<i>Marlies van Steenberghe, Jurjen Schipper, Rik Bos, and Sjaak Brinkkemper</i>	
Decoupling Models and Visualisations for Practical EA Tooling . . . . .	62
<i>Steffen Kruse, Jan Stefan Addicks, Matthias Postina, and Ulrike Steffens</i>	
Cross-Organizational Security – The Service-Oriented Difference . . . . .	72
<i>André Miede, Nedislav Nedyalkov, Dieter Schuller, Nicolas Repp, and Ralf Steinmetz</i>	
<b>1st International Workshop on SOA, Globalization, People, and Work (SG-PAW)</b>	
Enterprise Oriented Services (Position Paper) . . . . .	82
<i>Daniel Oppenheim, Krishna Ratakonda, and Yi-Min Chee</i>	
Automated Realization of Business Workflow Specification . . . . .	96
<i>Guohua Liu, Xi Liu, Haihuan Qin, Jianwen Su, Zhimin Yan, and Liang Zhang</i>	

PeopleCloud for the Globally Integrated Enterprise . . . . . 109  
*Maja Vukovic, Mariana Lopez, and Jim Laredo*

Public Disclosure versus Private Practice: Challenges in Business  
 Process Management (Position Paper) . . . . . 115  
*Stacy Hobson, Sameer Patil, and Xuan Liu*

**1st International Workshop on Service Oriented  
 Computing in Logistics (SOC-LOG)**

Analysing Dependencies in Service Compositions . . . . . 123  
*Matthias Winkler, Thomas Springer, Edmundo David Trigos, and  
 Alexander Schill*

Open Service-Oriented Computing for Logistics: A Case in Courier,  
 Express and Parcel Networks . . . . . 134  
*Marcel Kunkel, Christian Doppstadt, and Michael Schwind*

Gain in Transparency versus Investment in the EPC Network –  
 Analysis and Results of a Discrete Event Simulation Based on a Case  
 Study in the Fashion Industry . . . . . 145  
*Jürgen Müller, Ralph Tröger, Alexander Zeier, and Rainer Alt*

Using Automated Analysis of Temporal-Aware SLAs in Logistics . . . . . 156  
*Carlos Müller, Manuel Resinas, and Antonio Ruiz-Cortés*

Flexible SLA Negotiation Using Semantic Annotations . . . . . 165  
*Henar Muñoz, Ioannis Kotsiopoulos, Andrés Micsik,  
 Bastian Koller, and Juan Mora*

**Track 2: Service Quality and Service Level  
 Agreements Track**

**3rd Workshop on Non-functional Properties and  
 Service Level Agreements Management in Service  
 Oriented Computing (NFPSLAM-SOC 09)**

Runtime Prediction of Service Level Agreement Violations for  
 Composite Services . . . . . 176  
*Philipp Leitner, Branimir Wetzstein, Florian Rosenberg,  
 Anton Michlmayr, Schahram Dustdar, and Frank Leymann*

A Framework for Multi-level SLA Management . . . . . 187  
*Marco Comuzzi, Constantinos Kotsokalis, Christoph Rathfelder,  
 Wolfgang Theilmann, Ulrich Winkler, and Gabriele Zacco*

EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications . . . . .	197
<i>Jiang Dejun, Guillaume Pierre, and Chi-Hung Chi</i>	
On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management . . . . .	208
<i>Patrícia Silveira, Carlos Rodríguez, Fabio Casati, Florian Daniel, Vincenzo D'Andrea, Claire Worledge, and Zouhair Taheri</i>	
Transformation of Intermediate Nonfunctional Properties for Automatic Service Composition . . . . .	218
<i>Haruhiko Takada and Incheon Paik</i>	
Dealing with Fixable and Non-fixable Properties in Service Matchmaking . . . . .	228
<i>Octavio Martín-Díaz, Antonio Ruiz-Cortés, José M<sup>a</sup> García, and Miguel Toro</i>	
Using SLA Mapping to Increase Market Liquidity . . . . .	238
<i>Marcel Risch, Ivona Brandic, and Jörn Altmann</i>	
Translation of Service Level Agreements: A Generic Problem Definition . . . . .	248
<i>Constantinos Kotsokalis and Ulrich Winkler</i>	
Ontology-Based Feature Aggregation for Multi-valued Ranking . . . . .	258
<i>Nathalie Steinmetz and Holger Lausen</i>	
<b>2nd International Workshop on Service Monitoring, Adaptation and Beyond (MONA+)</b>	
Multi-level Monitoring and Analysis of Web-Scale Service Based Applications . . . . .	269
<i>Adrian Mos, Carlos Pedrinaci, Guillermo Alvaro Rey, Jose Manuel Gomez, Dong Liu, Guillaume Vaudaux-Ruth, and Samuel Quaireau</i>	
Calculating Service Fitness in Service Networks . . . . .	283
<i>Martin Treiber, Vasilios Andrikopoulos, and Schahram Dustdar</i>	
Applying Process Mining in SOA Environments . . . . .	293
<i>Ateeq Khan, Azeem Lodhi, Veit Köppen, Gamal Kassem, and Gunter Saake</i>	
Monitoring within an Autonomic Network: A GANA Based Network Monitoring Framework . . . . .	303
<i>Anastasios Zafeiropoulos, Athanassios Liakopoulos, Alan Davy, and Ranganai Chaparadza</i>	

An Extensible Monitoring and Adaptation Framework . . . . . 314  
*Razvan Popescu, Athanasios Staikopoulos, and Siobhán Clarke*

Cross-Layer Adaptation and Monitoring of Service-Based Applications . . . . . 325  
*Raman Kazhamiakin, Marco Pistore, and Asli Zengin*

Towards a Unified Architecture for Resilience, Survivability and Autonomic Fault-Management for Self-managing Networks . . . . . 335  
*Nikolay Tcholtchev, Monika Grajzer, and Bruno Vidalenc*

Replacement Policies for Service-Based Systems . . . . . 345  
*Khaled Mahbub and Andrea Zisman*

Retry Scopes to Enable Robust Workflow Execution in Pervasive Environments . . . . . 358  
*Hanna Eberle, Oliver Kopp, Tobias Unger, and Frank Leymann*

Integrating Complex Events for Collaborating and Dynamically Changing Business Processes . . . . . 370  
*Rainer von Ammon, Thomas Ertlmaier, Opher Etzion, Alexander Kofman, and Thomas Paulus*

Towards Business-Oriented Monitoring and Adaptation of Distributed Service-Based Applications from a Process Owner’s Viewpoint . . . . . 385  
*Krešimir Vidačković, Nico Weiner, Holger Kett, and Thomas Renner*

Adaptation of Service-Based Applications Based on Process Quality Factor Analysis . . . . . 395  
*Raman Kazhamiakin, Branimir Wetzstein, Dimka Karastoyanova, Marco Pistore, and Frank Leymann*

Delivering Multimedia in Autonomic Networking Environments . . . . . 405  
*Vassilios Kaldanis, Ranganai Chaparadza, Giannis Katsaros, and George Karantonis*

An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring . . . . . 414  
*Dragan Ivanović, Manuel Carro, and Manuel Hermenegildo*

**Track 3: Service Engineering Track**

**5th International Workshop on Engineering Service-Oriented Applications (WESOA09)**

Service Customization by Variability Modeling . . . . . 425  
*Michael Stollberg and Marcel Muth*

Towards a Quality Model for Choreography . . . . .	435
<i>Michele Mancioppi, Mikhail Perepletchikov, Caspar Ryan, Willem-Jan van den Heuvel, and Mike P. Papazoglou</i>	
Towards a Conceptual Framework for Legacy to SOA Migration . . . . .	445
<i>Maryam Razavian and Patricia Lago</i>	
MINERVA: Model drIVeN and sERvice oRIented Framework for the Continuous Business Process improvEment and rELated Tools . . . . .	456
<i>Andrea Delgado, Francisco Ruiz, Ignacio García-Rodríguez de Guzmán, and Mario Piattini</i>	
Design for Adaptation of Service-Based Applications: Main Issues and Requirements . . . . .	467
<i>Antonio Bucchiarone, Cinzia Cappiello, Elisabetta Di Nitto, Raman Kazhamiakin, Valentina Mazza, and Marco Pistore</i>	
Towards Runtime Migration of WS-BPEL Processes . . . . .	477
<i>Sonja Zaplata, Kristian Kottke, Matthias Meiners, and Winfried Lamersdorf</i>	
Encapsulating Multi-stepped Web Forms as Web Services . . . . .	488
<i>Tobias Vogel, Frank Kaufer, and Felix Naumann</i>	
Adapter Patterns for Resolving Mismatches in Service Discovery . . . . .	498
<i>Hyun Jung La and Soo Dong Kim</i>	
<b>1st International Workshop on User-Generated Services (UGS2009)</b>	
Lightweight Composition of Ad-Hoc Enterprise-Class Applications with Context-Aware Enterprise Mashups . . . . .	509
<i>Florian Gilles, Volker Hoyer, Till Janner, and Katarina Stanoevska-Slabeva</i>	
User-Centric Composition of Service Front-Ends at the Presentation Layer . . . . .	520
<i>Tobias Nestler, Lars Dannecker, and Andreas Pursche</i>	
On the Support of Dynamic Service Composition at Runtime . . . . .	530
<i>Eduardo Silva, Luís Ferreira Pires, and Marten van Sinderen</i>	
Rethinking the Semantic Annotation of Services . . . . .	540
<i>Nikolaos Loutas, Vassilios Peristeras, and Konstantinos Tarabanis</i>	
Service Composition for Everyone: A Study of Risks and Benefits . . . . .	550
<i>Abdallah Namoun, Usman Wajid, and Nikolay Mehandjiev</i>	

Using Personal Information Management Infrastructures to Facilitate User-Generated Services for Personal Use .....	560
<i>Olaf Grebner</i>	
Towards Ontology Matching for Intelligent Gadgets .....	570
<i>Oszkar Ambrus, Knud Möller, and Siegfried Handschuh</i>	
<b>Author Index</b> .....	581

# Future Research Topics in Enterprise Architecture Management – A Knowledge Management Perspective

Sabine Buckl, Florian Matthes, and Christian M. Schweda

Technische Universität München, Institute for Informatics,  
Boltzmannstr. 3, 85748 Garching, Germany  
{sabine.buckl,matthes,schweda}@in.tum.de

<http://www.systemcartography.info>

**Abstract.** Identifying, gathering, and maintaining information on the current, planned, and target states of the architecture of an enterprise is one major challenge of enterprise architecture (EA) management. A multitude of approaches towards EA management are proposed in literature greatly differing regarding the underlying perception of EA management and the description of the function for performing EA management. The aforementioned plurality of methods and models can be interpreted as an indicator for the low maturity of the research area or as an inevitable consequence of the diversity of the enterprises under consideration pointing to the enterprise-specificity of the topic. In this paper, we use a knowledge management perspective to analyze selected EA management approaches from literature. Thereby, we elicit constituents, which should be considered in every EA management function from the knowledge management cycle proposed by Probst. Based on the analysis results, we propose future research topics for the area of EA management.

**Keywords:** EA management function, knowledge management.

## 1 Motivation

Knowledge is often referred to as an competitive advantage for enterprises in to-days ever changing market environment. Thereby, this advantage does not only refer to knowledge about the environment, e.g. competitors, future trends and technologies, but also to knowledge about the internal make-up and processes of an enterprise. This internal make-up forms the management body of enterprise architecture (EA) management. EA is thereby understood as the "fundamental conception of a 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" [9]. The goal of EA management is to enable the enterprise to flexibly adapt via business/IT alignment [1].

Typical application scenarios of EA management are inter alia strategic IT planning, process optimization, and architecture reviews of projects [1]. Thereby, one major challenge of EA management is to foster the communication between



the involved stakeholders, e.g. the project director, the standards manager, and the enterprise architect in the case of an architecture review process. Thus, the task of EA management is to support decision making, via providing the required information in an appropriate form to the respective stakeholder. According to Matthes et al., EA management can be defined as "a continuous, iterative (and self maintaining) process seeking to improve the alignment of business and IT in an (virtual) enterprise. Based on a holistic perspective on the enterprise furnished with information from other enterprise level management processes [e.g. project portfolio management] it provides input to, exerts control over, and defines guidelines for other enterprise level management functions" [2]. The definition underlines the importance of information exchange for EA management. Likewise, typical tools providing support for EA management provide functionalities like *import, editing of data, creating visualizations, or communication and collaboration support* [10] also emphasize this aspect.

Similar to EA management, knowledge management (KM) is concerned with managing the "cooperation's knowledge through a systematically and organizationally specified process for acquiring, organizing, sustaining, applying, sharing, and renewing both the tacit and explicit knowledge of employees to enhance organizational performance and create value" [5]. Although the importance of information gathering, communication, and exchange for EA management is discussed repeatedly in literature about EA management (cf. [3,6,11,14]), no attempt has been performed to analyze and enhance existing EA management approaches from a KM perspective. Derived from this research gap, the article answers the following research questions:

How do existing EA management approaches address KM aspects of EA management? Which future research topics for EA management can be derived from a KM perspective?

The article firstly gives an overview on KM theories and selects the one of Probst (cf. [13]) as basis for future discussions (see Section 2). In Section 3 a KM perspective on EA management is established and used to assess prominent EA management approaches. The analyses' findings are used in Section 4 to outline areas for future development of EA management.

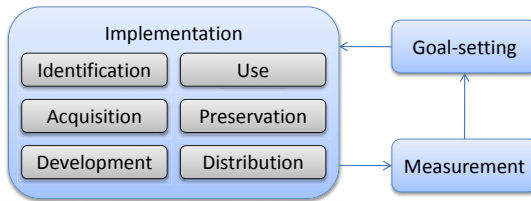
## 2 A Model for Knowledge Management

Academic research has brought up quite a few different models for knowledge management, which differ in respect to the perspective, they take on this management area. We revisit two prominent models for knowledge management and decide on the one most useful for answering above research questions. The criterion of usefulness and purposefulness is according to Probst [13] a simple but effective one for selecting an appropriate model of knowledge management, as the following quote of Probst subsumes:

While there is no single "right" model of KM, there is a simple criterion for evaluating any model: how useful is it [for] a chosen question?

Against above research questions, the KM models of Nonaka and Takeuchi (cf. [12]) and Probst (cf. [13]) are analyzed. In line with Holsapple and Joshi [8] we notice that these models, similar to most KM models, are *descriptive*, i.e., help to understand and explain KM phenomena. In this respect, they can be used in this paper, as the addressed research questions are concerned with understanding EA management from a KM perspective.

Nonaka and Takeuchi (cf. e.g. [12]) take an actor-centric perspective on KM. They identify four kinds of *knowledge conversion* activities that take place during knowledge creation in an organization. These are *socialization*, *externalization*, *combination*, and *internalization*. The activities are called *conversions* there, as they "convert" knowledge between different types, namely between *tacit* and *explicit* knowledge on the one hand, and between *individual* and *collective* knowledge on the other hand. In this framework of knowledge types, the activity of *socialization* converts knowledge of one entity to collective knowledge of a group. During *externalization* tacit knowledge is converted to explicit knowledge, codified in a knowledge representation. Explicit knowledge is in the *combination* activity combined by an individual into new knowledge. Finally, explicit knowledge is converted to tacit knowledge of an individual during the *internalization* activity. The model of Nonaka and Takeuchi (cf. [12]) can be used to understand how individuals act during knowledge creation in an organization and allows for a sociologic perspective on KM processes. This perspective is nevertheless only of minor interest in respect to the questions from Section 1.



**Fig. 1.** The KM cycle of Probst (cf. [13])

The KM cycle of Probst as presented in [13] consists of several *building blocks* for KM, reflecting typical activities that are carried out to avoid knowledge problems. As the cycle forms on the one hand a comprehensive model for KM and is on the other hand explained in very detail, it is subsequently sketched to provide the basis for the KM perspective on EA management. The KM cycle actually consists of the following two cycles, of which Figure 1 gives an overview:

- an *outer* cycle consisting of *goal setting*, *implementation* and *measurement*
- an *inner* cycle detailing the implementation activity into the sub-activities of *identification*, *acquisition*, *development*, *distribution*, *preservation*, and *use*

*Knowledge identification* is concerned with determining the knowledge that exists in an organization, and relating this knowledge to the one existing in the

organization's environment. The activity increases transparency of knowledge, and may help to identify redundant as well as missing knowledge. *Knowledge identification* can, if the number of knowledge sources to process is abundant, resort itself to *critical* knowledge as defined in the activity of *goal setting*.

*Knowledge acquisition* accounts for the fact that due to the growth of overall knowledge an organization is not capable to build up and maintain all needed know-how. Therefore, knowledge is imported over different *import channels*:

- acquisition of companies holding the corresponding knowledge
- stakeholder participation, e.g. by involving the customers of the organization
- counseling by experts that contribute to the organization's knowledge
- acquisition of knowledge products that foster the development of new knowledge (does not directly improve the organization's knowledge)

*Knowledge development* produces new knowledge on individual and collective level in a creative process, which can only to a very limited extent be discussed from a management perspective. Multiple sociological and psychological theories center around this activity and may be appropriate to study the process more in-depth. Linking back to the level of organizational KM and organizational development, e.g. an *atmosphere of trust* in the organization is regarded as a prerequisite to effective knowledge development.

*Knowledge distribution* means making knowledge available across the organization. Put in the words of Probst, as stated in [13], *knowledge distribution* is about the critical questions of **Who should know what, to what level of detail, and how can the organization support these processes of knowledge distribution?** These questions account for the fact that not everyone needs to know everything, as in contrast information overload might be as detrimental as a lack of information. Concerning the activity of knowledge distribution, the role of supporting tools and techniques should neither be underestimated nor overestimated. Useful and broadly accepted tools, and widely employed techniques can help to facilitate in the same ways as dysfunctional tools and not well adopted techniques can hamper effective *knowledge distribution*. As user acceptance is crucial for a tool or technique being an effective distribution facilitator, many organizational and non-technical issues have to be concerned regarding *knowledge distribution*.

*Knowledge use* forms the actual purpose of KM and refers to the application of knowledge in the production process of an organization. In respect to the later focus on EA management, which is no production process, the above statement can be reformulated as follows: *knowledge use* refers to the application of knowledge in the purpose-generating process of an organization. Here again, tools and techniques can be applied as facilitators; this is not surprising as especially in knowledge-intensive processes the borders between distribution and use are sometimes unclear. Notwithstanding, *knowledge use* should explicitly be accounted for, as the *goal setting* activity purposefully targets the use activity.

*Knowledge preservation* is concerned with avoiding the loss of valuable and purpose-relevant expertise in an organization. While tacit knowledge is more often subject to loss, e.g. due to an expert leaving, also explicit knowledge has to be preserved. Probst refers to outdated storage systems as *dead storage systems*,

colloquially stating that a storage system, which is not longer maintained, may cause knowledge loss as well as a leaving expert. Techniques and tools used for knowledge distribution can also be helpful for knowledge preservation.

Complementing the inner cycle of *knowledge implementation*, two more activities that constitute an embracing and sustainable KM are introduced below.

*Goal-setting*, i.e., the development of knowledge goals, establishes a conceptual framework for organization-specific KM. The knowledge goals determine which capabilities should be built on which level. Different levels of abstraction in respect to the formulation of goals can be distinguished. Most important for the subsequent considerations are the levels of *strategic knowledge goals* and *operational knowledge goals*. While the former goals describe a long-term vision of the knowledge portfolio of the organization, the latter goals operationalize the vision, i.e., translate it into action. Making the knowledge goals explicit is regarded highly important for controlling the evolution of the KM.

*Knowledge measurement* is concerned with measuring to which extent the knowledge goals have been fulfilled during the *implementation* activity. As knowledge is an intangible resource, indicators and measurement processes are hard to establish. To some extent the operational knowledge goals can be formalized that they can help to objectively assess certain aspects of KM. Nevertheless, a commonly accepted way to measure knowledge has yet not been established, such that managers concerned with KM activities have to rely on their subjective perception of goal fulfillment. Additionally, surveys on user satisfaction with knowledge access in distinct areas, which reflect certain knowledge goals, may be helpful during *knowledge measurement*.

### 3 Analyzing Existing EA Management Approaches from a Knowledge Management Perspective

Preparing the subsequent analyses of prominent EA management approaches from a KM perspective, the KM model of Probst [13], more precisely its building blocks, are mapped to the application domain of EA management. To ground the mapping solidly in the application domain of EA management, the outer cycle's activities of KM are mapped first, starting with the *implementation* activity. This activity can be identified with the core of EA management, i.e., with the "continuous process seeking to improve the alignment of business and IT in a (virtual) enterprise". This part of the definitional statement towards EA management (cf. Section II) sketches the main goal of the implementation of EA management, but does not provide further details on the implementation. These are later discussed along the activities from the inner cycle. Continuing with the activities from the outer cycle, both *knowledge measurement* and *goal-setting* can be identified with the aspect of "self maintenance" of the EA management process. More precisely, an effective and continuous EA management, established as a management function within an enterprise, must define the share of the overall architecture of the enterprise that it covers. This can be understood as goal-setting, i.e., defining which knowledge about the architecture is needed;

multiple EA management approaches target this topic. The knowledge measurement closes a feedback loop by assessing to which extent the knowledge goals could be attained. Put in the EA management terminology, the measurement activity assesses, if the architecture concepts defined during goal-setting have adequately been considered during EA management. This provides input for revisiting the knowledge goals, if albeit a good coverage of relevant architecture concepts, an increased alignment between business and IT could not be achieved.

Above considerations on EA management from a KM perspective partially neglect process-related aspects of EA management. To some extent this narrow focus is broadened by diving into the details of *implementation* activity, but the focus in this paper lays on the knowledge and information aspect of EA management not on the process aspect thereof. The sub-activities of the building block *implementation* can be mapped as follows to the domain of EA management. During *knowledge identification* possible sources of information about the EA are identified. These sources may be both people, as e.g. business or enterprise architects, but also documentation tools. *Knowledge acquisition* relates to activities as EA management counseling by consultancies, more detailed with incorporating best-of-breed EA-related solutions into the EA knowledge of the company. In the context of EA management, *knowledge development* can refer to planning and decision activities, where additional knowledge about the EA is created. *Knowledge distribution* maps to the EA management activity of communicating architectural knowledge, i.e., as information on current and planned architectures, to people involved in other enterprise level management functions, as e.g. project portfolio management. In this vein, *knowledge preservation* can be understood as storing this architecture knowledge in a way that interested stakeholders can access it. Additionally, preservation is also concerned with making accessible not only the most recent architectures, but also former plans and documentations. Finally, *knowledge use* can be identified with management activities in the enterprise-level management functions that access the architecture knowledge for deciding, planning, executing, or measuring. Based on the KM perspective on EA management existing approaches to EA management originating from academia and practice are detailed and discussed subsequently.

A well-known approach to EA management is *The Open Group Architecture Framework (TOGAF)* [14], whose main constituent is the *Architecture Development Method (ADM)*, which describes a cyclic project-oriented process for EA management. In the ADM each EA management project starts with the *preliminary* phase, which defines the project's scope and reach (**knowledge goal-setting**) and decides on other frameworks and tools to be utilized (**knowledge acquisition**). The preliminary phase is followed by the *architecture vision* phase in which future states of the EA are developed (**knowledge development**). The current state of the EA is documented in three distinct phases, which focus on different parts of the architecture – the *business architecture* phase, the *information systems architecture* phase, and the *technology architecture* phase. Although information has to be gathered and consolidated in these phases, TOGAF only addresses the challenge of **knowledge identification** via a *stakeholder*

*management*. Means and methods how to draw knowledge e.g. from tools, already in use, are not referred to. Based on the current and future states of the EA, the *opportunities and solutions* phase develops plans for the evolution, which are decided upon and detailed during the *migration planning* phase. The migration plans are subsequently realized in the *implementation governance* phase, in which other management functions, e.g. project portfolio management, are provided with knowledge to support decision making (**knowledge use**). Finally, the phase *architecture change management* assesses the quality of the developed architecture and handles change requests. Although this phase partially incorporates **knowledge measurement**, important aspects of this KM activity are not considered, e.g. a continuous improvement of the overall process. Whereas the task of **knowledge distribution** is indirectly mentioned in the some phases of the ADM, see e.g. the objective "confirm the transition architectures [...] with relevant stakeholder" [14], methods and means how to conduct this task are not further detailed. Similarly, the challenge of **knowledge preservation** does not form a focal point of TOGAF. Viewpoints to communicate architectural knowledge are textually described but no further explanation how a specific stakeholder can access and use the information are given.

The *Enterprise Architecture Management Pattern Catalog (EAMPC)* [4] was developed at the Technische Universität München and contains a collection of best practice methods, visualizations, and information models for EA management. The intent of the EAMPC is to support EA practitioners in the concern-driven development of an enterprise-specific EA management function. Concerns represent typical problems, which occur in the context of managing an EA, for instance, "Which business processes, if any, are suitable candidates for outsourcing?" [4] The concerns contained in the EAMPC address the different areas, e.g. business process support management and application landscape management. The topic of application landscape management is concerned with evolution aspects (**knowledge development**). In order to use the EAMPC, the enterprise under consideration has to select the appropriate concerns (**goal-setting**). Based from the selected concerns, the according *methodology patterns (M-Patterns)* addressing the concerns can be derived. Within each M-Pattern one or more *viewpoint patterns (V-Patterns)* are referenced. These V-Patterns describe how the information, which is necessary to address the concern, can be visualized and presented to the involved stakeholders (**knowledge preservation**). A description of the corresponding information including the types of elements, their attributes, and relationships to each other is given in the *information model patterns (I-Patterns)*, which are referenced by V-Patterns. Methods and means to gather the according information are described as part of the solution description of the M-Pattern (**knowledge identification**). The M-Pattern further described required governance structures, roles, and responsibilities. Thereby, the links to other enterprise-level management functions, as e.g. the project portfolio management, are discussed and the type of relationships, ranging from information provision to enforcing, is described (**knowledge distribution and use**). Methods for assessing the performance of

an EA management function (**knowledge measurement**) and for **knowledge acquisition** are not described in the EAMPC.

Niemann presents an approach to EA management organized in the phases *document*, *analyze*, *plan*, *act*, and *check* [11]. According to Niemann, the objective of EA management is to support an enterprise in "doing the right thing right, with minimal risk" [11]. The approach provides a standard information model and does hence not account for enterprise-specific **goal-setting**. The model consists of three submodels for the business, application, and systems architecture. Information about the current state of these architectures is gathered in the *document* phase. Whereas the description of the document phase emphasizes on *what* should be documented and *how* it should be documented (**knowledge preservation**, the question *where* to gather the respective data from (**knowledge identification**) is only briefly sketched. Based on the results of the *document* phase, the *analyze* phase assesses certain architectural properties, e.g. heterogeneity, complexity, or costs. Based on the the analyses' results, future plans for the EA are derived, evaluated, and decided upon in the *plan* phase (**knowledge development**). The developed roadmap is realized in the *act* phase, in which EA management influences demand and portfolio management as well as program and service management functions (**knowledge usage**). The *check* phase analyzes key performance indicators for the EA that may influence marketing for EA management, which is according to Niemann one key success factor. Marketing methods are described on a very abstract level (**knowledge distribution**) [11]. The performance measurements described by Niemann mostly target the EA, whereas only one measurement – the *architecture management scorecard* – measures the EA management function itself [11]. Although other frameworks and tools for EA management are mentioned in [11], a combination with such approaches is not described (**knowledge acquisition**).

In [7] Hafner and Winter derive a process model for architecture management from three case studies. The model consists of four phases: *architecture planning*, *architecture development*, *architecture communication*, and *architecture lobbying*. One activity of the *architecture planning* phase is the identification of strategic requirements. This activity may in line with [6] be understood as defining the share of the enterprise that should be considered by the EA management (**goal-setting**). For each area-of-interest in the EA the corresponding stakeholders should be identified (**knowledge preservation**). A process and involved roles for **knowledge identification** in the architecture planning phase are described in [6]. During planning, the current architecture is further assessed, architecture principles are revised, and the future states of the EA are updated (**knowledge development**). Whereas the *architecture planning* phase focuses on strategic aspects, *architecture development* focuses on operational aspects. Main activities of this phase are to identify and manage further requirements as well as piloting, developing, and integrating architecture artifacts. *Architecture communication* is concerned with identifying relevant stakeholders and communicating architecture artifacts (**architecture distribution**). The phase *architecture lobbying* targets aspects like assistance for running projects via consultancy, which

is a part of knowledge dissemination (**knowledge use**) aiming to influence and control projects. Whereas assessment and analyses of different states of the EA are discussed in the process model of Hafner and Winter [7], a process phase to analyze the EA management function itself (**knowledge measurement**) is not described. The possibility to complement the process model with other external resources, e.g. frameworks, (**knowledge acquisition**) is also not discussed.

## 4 Proposing Topics for Future EA Management Research

Table 4 summarizes the results of the literature analysis from Section 3, preparing a discussion on common strengths and weaknesses of the analyzed approaches. Three of the four approaches analyzed provide a "standard" reference method for EA management (cf. [7][11][14]). These "one-size-fits-it-all" methods contain generic goals for EA management, as architecture roadmapping and transformation planning. TOGAF additionally mentions the importance of enterprise-specific goals, but does not provide exemplary ones [14]. The EAMPC in contrast lists typical EA management concerns, which can be used to support the **goal-setting** for an enterprise-specific EA management function. The absence of concrete goals might explain the lack of methods for assessing and **measuring** the EA management function itself, which is a common weakness of most of the analyzed approaches. From this weakness, we derive a first topic for future research: *Operationalizing knowledge goals for EA management*. While existing approaches currently focus on general tasks of EA management, typical goals are of interest in order to derive the necessary knowledge demands. Via the selection of the relevant goals, an enterprise can configure the reference method according to its specific demands. Accordingly, methods and means for assessing and measuring the achievement of these goals can be developed on explicit goals. They lay the foundation for an EA management governance method.

**Identifying**, gathering, and maintaining knowledge about the EA is a challenge, which is only recently addressed by isolated approaches (cf. [6]). As the analyzed EA management methods do not detail on how to **acquire** and incorporate knowledge from other sources, is limited. Therefore, future research should focus on the *integration of existing EA management approaches* instead

**Table 1.** A KM perspective on existing EA management approaches

	[14]	[3,4]	[11]	[6,7]
Goal-setting	●	●	●	●
Measurement	○	○	○	○
Identification	●	●	○	●
Acquisition	●	●	○	○
Development	●	●	●	●
Use	○	●	●	○
Preservation	○	●	●	●
Distribution	●	●	●	●



of developing the wheel over and over again. Additional guidance on how to accomplish this integration, e.g. via openly configurable EA management reference methods needs to be developed and researched. From a KM perspective, common strengths of the analyzed EA management approaches are the **development** and **use** of knowledge. All these approaches provide means and methods to develop future states of the EA and evolution roadmaps. Nevertheless, these means and methods are mainly approach-specific and cannot be reused in other approaches. Future research should be focused on interoperability of the methods.

Although the analyzed approaches agree that the enterprise is a complex socio-technical system, only one approach [14] details on the aspect of human stakeholders and their involvement in EA management. Therefore, the **distribution** of knowledge is often discussed by referring to the related management processes, e.g. project portfolio management, without explicating stakeholders involved. Similarly, the **preservation** of knowledge is only mentioned as a challenge, which should be addressed via EA management tools. Future research could target the establishment of a more systematic stakeholder model for EA management together with a structured approach to describe the corresponding viewpoints. Additionally, the topic of knowledge preservation class for techniques that help to access and compare past (planning) states of the EA.

Above analyses showed that some KM activities are only partially addressed by current EA management approaches. Future research may concentrate on these activities in two ways, namely by *theorizing* explanations for the lower importance of the activities or by *improving* the support for the activities. In-depth analyses of successful EA management approaches from a KM perspective help to pursue both directions. The analyses may show that the activities are actually not considered relevant, as companies perceive them adequately addressed. In this case, proven practice methods for distributing and preserving EA knowledge could be documented to complement the existing EA management approaches in literature. If in contrary, a lack of adequate support is discovered, KM models may be used to improve existing EA management approaches. Especially the *operationalization of knowledge goals* seems to be a promising way to improved stakeholder-specific knowledge distribution and preservation.

## References

1. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – literaturüberblick stand der praxis. *Wirtschaftsinformatik* 50(4), 292–304 (2008)
2. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M.: State of the art in enterprise architecture management 2009. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2009)
3. Buckl, S., Ernst, A.M., Matthes, F., Ramacher, R., Schweda, C.M.: Using enterprise architecture management patterns to complement togef. In: The 13<sup>th</sup> IEEE International EDOC Conference (EDOC 2009), Auckland, New Zealand. IEEE Computer Society, Los Alamitos (2009)
4. Chair for Informatics 19 (sebis), Technische Universität München. Eam pattern catalog wiki (2009), <http://eampc-wiki.systemcartography.info> (cited 2010-02-25)

5. Davenport, T.H., Prusak, L.: *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston (1998)
6. Fischer, R., Aier, S., Winter, R.: A federated approach to enterprise architecture model maintenance. In: *Proceedings of the 2<sup>nd</sup> International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007)*, St. Goar, Germany, October 8-9, pp. 9–22 (2007)
7. Hafner, M., Winter, R.: Processes for enterprise application architecture management. In: *Proceedings of 41<sup>st</sup> Hawaii International International Conference on Systems Science (HICSS-41 2008)*, Waikoloa, Big Island, HI, USA, January 7-10, p. 396. IEEE Computer Society, Los Alamitos (2008)
8. Holsapple, C.W., Joshi, K.D.: Description and analysis of existing knowledge management frameworks. In: *32<sup>nd</sup> Hawaii International International Conference on Systems Science (HICSS-32 1999)*, Waikoloa, Big Island, HI, USA, vol. 1, p. 1072. IEEE Computer Society, Los Alamitos (1999)
9. International Organization for Standardization. *Iso/iec 42010:2007 systems and software engineering – recommended practice for architectural description of software-intensive systems* (2007)
10. Matthes, F., Buckl, S., Leitel, J., Schweda, C.M.: *Enterprise Architecture Management Tool Survey 2008*. Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2008)
11. Niemann, K.D.: *From Enterprise Architecture to IT Governance – Elements of Effective IT Management*. Vieweg+Teubner, Wiesbaden (2006)
12. Nonaka, I., Takeuchi, H.: *The Knowledge-Creating Company*. Oxford University Press, Oxford (1995)
13. Probst, G.J.B.: *Practical knowledge management: A model that works*. Arthur D Little PRISM (1998)
14. The Open Group. *TOGAF "Enterprise Edition" Version 9* (2009), <http://www.togaf.org> (cited 2010-02-25)

# Enterprise Architecture Principles: Literature Review and Research Directions

Dirk Stelzer

Fachgebiet Informations- und Wissensmanagement, Technische Universität Ilmenau,  
Postfach 100565, 98684 Ilmenau, Germany  
dirk.stelzer@tu-ilmenau.de

**Abstract.** This paper presents a literature review on enterprise architecture principles. The results show that there are various gaps in the research literature: No accepted definition of enterprise architecture principles has emerged yet. A detailed conceptual framework is still lacking. Business principles, IT principles and enterprise architecture principles are often mixed up. Research into generic design principles is still in its infancy. Our review illustrates the necessity to conduct more research on enterprise architecture principles. We describe conceptual foundations and provide guidance for further research in this field.

**Keywords:** enterprise architecture, architecture principles, design principles, representation principles, design rules, literature review.

## 1 Introduction

The term architecture is defined as the “fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution” [1], [2]. Accordingly, we define enterprise architecture as the fundamental organization of an enterprise embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

According to the Oxford Dictionary of English a principle – among other explanations – is (1) a fundamental truth or proposition serving as the foundation for belief or action, (2) a rule or belief governing one’s personal behaviour, (3) a general scientific theorem or natural law, (4) a fundamental source or basis of something.

In the context of enterprise architecture, however, “a precise definition of the concept of principles as well as the mechanisms and procedures needed to turn them into an effective regulatory means still lacks” (p. 49) as van Bommel et al. point out [3]. As a matter of fact, when conducting an initial examination of publications on enterprise architecture principles we found various interpretations of the concept. Some authors take individual views leading to inconsistencies in research findings.

Compared to the literature on enterprise architecture in general, the number of publications on enterprise architecture principles is limited. This is surprising as various authors [4], [5], [6], [7], [8], [9] reckon architecture principles as pivotal elements of enterprise architectures. Hoogervorst actually equates architecture with principles.

He defines architecture “as a consistent set of design principles and standards that guide design” (p. 215) [10]. Richardson, Jackson, and Dickson call principles “the most stable element of an architecture” (p. 389) [6]. Aside from that, architecture principles are central elements of enterprise architecture frameworks such as TOGAF [7].

The aim of this paper is to conceptualize the research area of enterprise architecture principles, to examine prior research, and to identify research options.

The remainder of the paper is organized as follows. Section 2 contains conceptual foundations of enterprise architecture principles. Section 3 describes results of our literature review. In section 4 we summarize our findings and provide directions for further research.

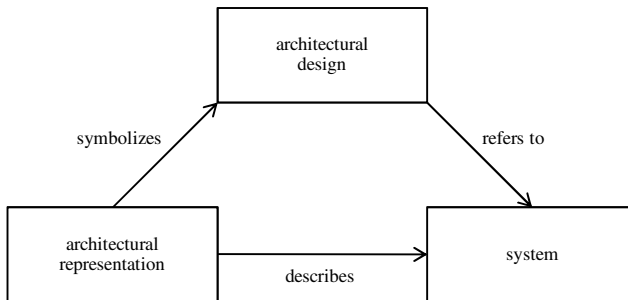
## 2 Conceptual Foundations

In the following sections we describe conceptual foundations of enterprise architecture principles.

### 2.1 Architectural Triangle

In the architecture framework proposed by The Open Group [7] architecture “has two meanings depending upon the context: 1. A formal description of a system, or a detailed plan of the system at component level to guide its implementation [and] 2. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.” [7] In other words, the term architecture may denote both, the inherent structure of a system and its representation. Hence, architecture and architectural representation should be distinguished. The architecture is a conceptual model of the system of interest. The architectural representation is a more or less formal description of the architecture. Figure 1 illustrates the associations of a system, its architecture, and architectural representation arranged parallel to the so-called semiotic triangle [11], [12].

Architecture principles may refer either to the design or to the representation of architectures. We label the first as design principles and the latter as representation principles. Design principles are fundamental propositions guiding the construction and



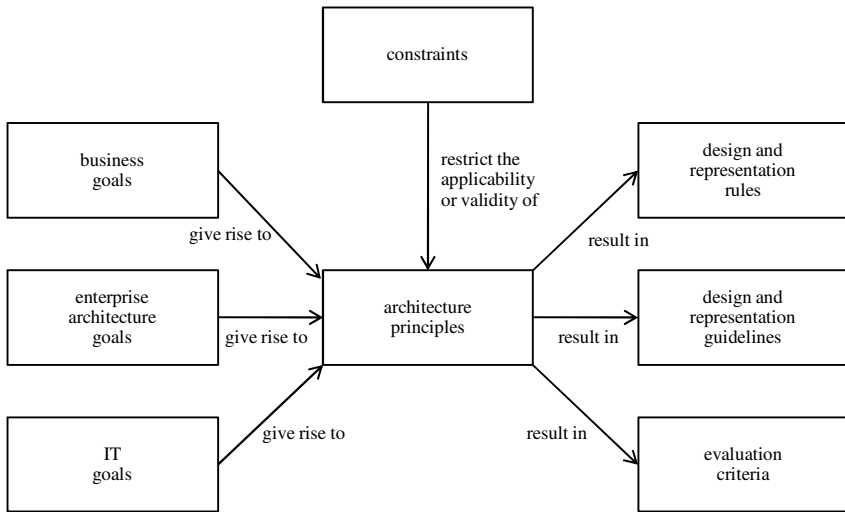
**Fig. 1.** Architectural Triangle

evaluation of architectures, e.g. separation of concerns, modularity, or loose coupling. Representation principles are fundamental propositions for describing and modeling architectures, as well as for evaluating architectural representations. Examples for representation principles are understandability, consistency, and completeness.

### 2.2 Context of Architecture Principles

Principles are means to achieve certain ends. When designing enterprise architectures principles serve to accomplish business, IT, or architecture goals. Constraints (e.g. strategic, financial or technological limitations) may restrict the applicability or validity of architecture principles.

Since principles are usually abstract, high-level propositions they need to be specified in order to guide the development or evaluation of a system. This is often realized by providing rules or guidelines for the development of architectures and evaluation criteria for quality assessment. Schekkerman defines a rule as “a prescription on how something has to be done” (p. 34) [5]. Guidelines are less rigorous. They provide guidance for behavior but do not call for strict obedience. Evaluation criteria are quality characteristics for the assessment of architectural designs or representations. Figure 2 shows the context of architecture principles.

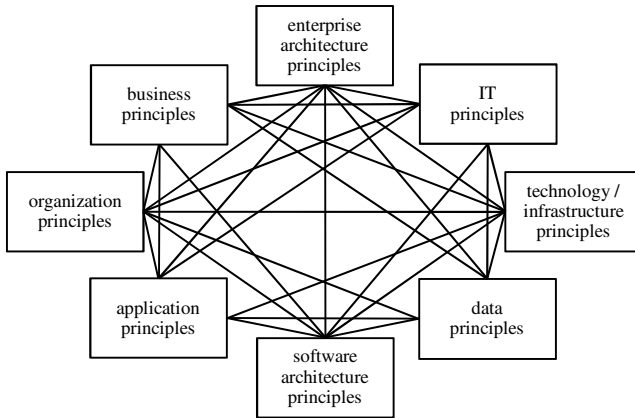


**Fig. 2.** Context of Architecture Principles

Rules, guidelines, and criteria for designing or evaluating architectures should be derived from architecture principles, which in turn should be derived from relevant goals [13].

### 2.3 Network of Principles

Architecture principles are usually embedded in a network of associated principles, for example, business and IT principles as well as principles that refer to elements of



**Fig. 3.** Network of Principles

enterprise architectures such as organization, application, software architecture, data, or infrastructure principles. Figure 3 exemplifies a network of principles.

## 2.4 Level of Universality of Architecture Principles

Architecture principles in related arenas, such as software engineering or organizational design, are generic propositions that are largely independent from mission, strategy, objectives, constraints, or conditions of a particular enterprise, organizational unit, or project. Examples for software architecture principles are separation of concerns, modularity, loose coupling, and tight cohesion [14].

Principles described in the literature on enterprise architecture are often enterprise-specific, i.e. tailored to the needs of the enterprise. For example: “Star Enterprise IT areas will need to collaborate to provide the best service in application development and support, and to eliminate artificial internal competition” (p. 389) [6].

## 3 Literature Review

### 3.1 Method

We used a structured approach recommended by Webster and Watson [15] to identify relevant publications for the review. As a first step, we examined IS journals and IS conference proceedings using the EBSCO database and the Web of Science. We conducted electronic searches in titles and abstracts on the following keywords: “enterprise architecture” and “principle” or “design” or “rule” or “guideline”.

In a second step, we extended our search to IS journals and conference proceedings that were not covered by our original search. These searches identified a total of 42 articles. After analyzing each article’s abstract, keywords, or the full article when necessary, we excluded 27 articles that did not appear to be concerned with or relevant to enterprise architecture principles. This process provided 15 articles for in-depth review.

In a third step we reviewed the citations in the articles identified in the previous steps to determine prior articles on enterprise architecture principles. A further set of four articles from journals and conference proceedings other than those formally searched was collected and a subset of 19 articles was read in full and coded.

We did not include articles on architecture principles from related research areas such as software engineering or organizational design and engineering. We limited our review to articles focusing on enterprise architecture principles. We also did not include articles discussing principles for designing or evaluating enterprise architecture frameworks or principles for service oriented architectures [16], [17]. We excluded all publications that only mentioned the terms architecture principles, design rules, or guidelines without elaborating on these concepts. Out of the 19 coded articles, eleven include passages of interest. They are compiled in the analysis.

### 3.2 Overview of the Literature

Compared to the considerable amount of publications on enterprise architecture in general, the number of articles presenting research findings on enterprise architecture principles is rather low. This is surprising because architecture principles are considered to be essential elements of architectures [1], [2], [7]. We identified no more than eleven publications that analyze enterprise architecture principles [3], [4], [6], [8], [9], [10], [13], [18], [19], [20], and [21]. Only four of these articles [3], [6], [9], and [13] have their main emphasis on principles. The other seven articles discuss principles among other topics.

Prevalent methodologies used are case studies [4], [6], [21], [13] and conceptual descriptions [3], [8], [9], [10], [19], and [20]. One article [18] uses mathematical representation. We did not find any survey that explores development, use, or evaluation of architecture principles in multiple enterprises.

### 3.3 Definitions of Enterprise Architecture Principles

No accepted definition of the term enterprise architecture principles has emerged yet. Table 1 presents definitions of architecture principles covered by our review. Authors of articles [4], [18], [20], and [21] do not define the term.

It is remarkable that the term architecture principle is defined in six articles only. One article [10] equates a collection of design principles with enterprise architecture. It is also interesting that only one definition [6] emphasizes that principles may guide design *and* evaluation of architectures. The other definitions focus on the design purpose of principles.

Furthermore, some definitions focus on selected layers of enterprise architectures. They do not seem to consider all layers of enterprise architectures. Some definitions focus on IT (“simple, direct statements of how an enterprise wants to use IT” [8]; “rules and guidelines for the use and deployment of all IT resources and assets” [13]), others on business (“rules and guidelines ... that inform and support the way in which an organization sets about fulfilling its mission” [3] and [9]). Only two definitions [6] and [19] comprise enterprise architecture in its entirety.

**Table 1.** Definitions of Enterprise Architecture Principles

References	Definitions
[3] referring to [7]	“Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.” (p. 49)
[6]	“Principles are an organization’s basic philosophies that guide the development of the architecture. ... Principles provide guidelines and rationales for the constant examination and re-evaluation of technology plans.” (p. 389)
[8]	“... simple, direct statements of how an enterprise wants to use IT. These statements establish a context for architecture design decisions by translating business criteria into language and specifications that technology managers can understand and use. Architecture principles put boundaries around decisions about system architecture.” (p. 38)
[9] referring to [7]	“Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.” (p. 1139)
[10]	(no explicit definition); “collectively the design principles are identified as enterprise architecture” (p. 217)
[13]	“Architectural principles define the underlying general rules and guidelines for the use and deployment of all IT resources and assets across the enterprise ...” (p. 2)
[19]	“Architecting principles are rules to use when elaborating enterprise architectures.” (p. 1214)

### 3.4 Context of Architecture Principles

The context of architecture principles as outlined in section 2.2 is often structured in rationales and implications [6], [8] or in goals and rules [9], [13], [20], [21]. A rationale gives an explanation for the principle. It states underlying reasons, in most cases by explaining the principle’s contribution to achieving architectural or business goals. Implications describe potential consequences for those in charge of developing, evaluating or deploying the architecture or elements of the architecture [6]. Accordingly, Hoogervorst claims: “All principles should have a three-fold context: the rationale of the principle, (2) the implications, and (3) key actions necessary for making the principle operational” (p. 229) [10]. Richardson, Jackson, and Dickson propose the following structure for describing enterprise architecture principles: (1) principle statement, (2) rationale, and (3) implications [6].

Constraints are neglected by nine articles included in our review. This is astonishing because constraints may help researchers and practitioners to correctly assess the principle’s scope and validity. Van Bommel et al. [9] suggest using constraints when formalizing architecture principles. Chen and Lillehagen [19] point out that architecture principles should be embedded in goals and objectives, constraints, conditions, and challenges.

Similar to the situation in service design [16] there is no empirical validation of successful architecture principles. We did not find any publication that explores the



relationship between deploying architecture principles and attaining architecture goals. Lessons learned when deploying enterprise architecture principles [6], [8], [13], [21] provide anecdotal evidence at best. Tested knowledge on which architecture principles facilitate the achievement of specific architecture goals would be highly beneficial for the enterprise architecture community.

### 3.5 Network of Principles

Interdependencies of enterprise architecture principles and other principles, e.g. IT principles or business principles, are not mentioned in ten of the articles included in our review [3], [4], [6], [8], [9], [10], [18], [19], [20], and [21].

Lindström [13] points out, that architectural principles should be derived from business principles which in turn should be derived from business strategy. IT governance and IT strategy principles should be derived from architectural principles. However, the distinction of architecture principles and other principles remains ambiguous. Lindström [13] cites the following architectural principles of Vattenfall, a major European energy provider: “IS/IT Strategy development shall be an integral part of business strategy development.” “Control of development and implementation of IS/IT projects must comply with a corporate common project management model.” Most of the so-called enterprise architecture principles reported in her article resemble IT principles. Lindström does neither distinguish IT principles from enterprise architecture principles nor does she explain criteria of how to tell between IT and architecture principles. The same is true for principles compiled by Richardson, Jackson, and Dickson [6].

### 3.6 Level of Universality of Architecture Principles

In our review we found articles that examine generic principles and other articles that report about enterprise specific principles. In a second step we analyzed which level of universality is addressed by the articles focusing on design principles and on representation principles respectively. Table 2 shows the level of universality and the nature of the architecture principles discussed in the articles included in our review.

**Table 2.** Level of Universality and Nature of Architecture Principles

	<b>design principles</b>	<b>representation principles</b>
<b>generic</b>	[19], [21]	[3], [4], [9], [18], [20]
<b>enterprise-specific</b>	[6], [8], [10], [13]	

The majority of the articles focus either on enterprise-specific design principles or on generic representation principles. Only two articles elaborate on generic design principles. No article describes enterprise-specific representation principles.

We were astonished about the fact that only two article describe generic design principles for enterprise architectures. We had expected that two decades of research on enterprise architectures would have yielded more knowledge on design principles that are independent of the specific circumstances of a particular enterprise.

## 4 Summary and Research Directions

The results of our review show that there are various gaps in the research literature: No accepted definition of enterprise architecture principles has emerged yet. Design and representation principles often are not explicitly distinguished. A detailed conceptual framework that could serve as a basis for conducting quantitative research is still lacking. Business principles, IT principles and enterprise architecture principles are often mixed up. Research into generic design principles is still in its infancy.

Our literature review illustrates the necessity to conduct more in-depth research on enterprise architecture principles. We suggest the following options for future research:

1. Identifying an appropriate definition of enterprise architecture principles. An acceptable definition should cover all layers of enterprise architecture and should not be restricted to particular layers. It should also account for the three major purposes of architecture principles: design, description, and evaluation of systems. We propose the following definition: Enterprise architecture principles are fundamental propositions that guide the description, construction, and evaluation of enterprise architectures. Enterprise architecture principles fall into two classes: Design principles guide the construction and evaluation of architectures. Representation principles guide the description and modeling of architectures, as well as the evaluation of architectural representations.
2. Obviously it is difficult to clearly distinguish enterprise architecture principles from IT principles or business principles. We do not know whether and how companies distinguish these categories of principles. More research is needed to answer the question whether this distinction is helpful and how it can be achieved.
3. Exploring the issues of enterprise architecture principles from more theoretical perspectives. Adjacent research areas may provide helpful insights to answer this question. Research into software engineering and software architecture has yielded considerable findings on architecture principles [14], [22]. Organizational design and engineering explore principles of how to design and to describe enterprises [23], [24]. Research on service oriented architectures has produced valuable knowledge on architecture principles [16], [17]. Systems architecting [25] may also provide helpful insights of how to design architecture principles.
4. Investigating generic enterprise architecture design principles. Findings on generic design principles are meager in the field of enterprise architecture. Adjacent research areas, such as software architecture or organizational design and engineering, have produced valuable knowledge on generic design principles. It would be highly interesting to explore whether there are generic design principles that are applicable to all layers of enterprise architectures. In a second step, scholars could address the question under which circumstances specific principles may contribute to the achievement of particular enterprise architecture goals.
5. Extending the basis of case studies. There are only few publications that describe practical experience with enterprise architecture principles. Since this research field has not yet been explored in detail and theoretical foundations are meager we need more explorative research. More case studies might help to shed light on key issues and success factors when formulating and deploying architecture principles.

6. Conducting quantitative research. When a detailed conceptual framework for exploring enterprise architecture principles is elaborated quantitative research should be conducted. Surveys covering multiple enterprises in various industries could help to assess whether enterprise architecture principles converge to a coherent set of generic principles or whether these principles need to be tailored to the specific needs of the particular enterprise.

## References

1. IEEE 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems – Description (2000)
2. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems (2007)
3. van Bommel, P., Buitenhuis, P.G., Stijn, J.B., Hoppenbrouwers, A., Proper, E.H.A.: Architecture Principles – A Regulative Perspective on Enterprise Architecture. In: Reichert, M., Strecker, S., Turowski, K. (eds.) EMISA 2007, pp. 47–60. Gesellschaft fuer Informatik, Bonn (2007)
4. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. JEA 3(2), 7–18 (2007)
5. Schekkerman, J.: Enterprise Architecture Good Practices Guide: How to Manage the Enterprise Architecture Practice, Trafford, Victoria (2008)
6. Richardson, G.L., Jackson, B.M., Dickson, G.W.: A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. MISQ 14(4), 385–403 (1990)
7. The Open Group: TOGAF Version 9. The Open Group Architecture Framework (TOGAF) (2009), <http://www.opengroup.org>
8. Armour, F.J., Kaisler, S.H., Liu, S.Y.: A Big-Picture Look at Enterprise Architectures. IEEE IT Professional 1(1/2), 35–42 (1999)
9. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, E.H.A., van der Weide, T.P.: Giving Meaning to Enterprise Architectures - Architecture Principles with ORM and ORC. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1138–1147. Springer, Heidelberg (2006)
10. Hoogervorst, J.: Enterprise Architecture: Enabling Integration, Agility and Change. IJCIS 13(3), 213–233 (2004)
11. Lyons, J.: Semantics. University Press, Cambridge (1977)
12. Ogden, C.K., Richards, I.A.: The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Harcourt & Brace, New York (1923)
13. Lindström, Å.: On the Syntax and Semantics of Architectural Principles. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, p. 10. Computer Society Press, Washington (2006)
14. Witt, B.I., Baker, F.T., Merritt, E.W.: Software Architecture and Design. Principles, Models, and Methods. Van Nostrand Reinhold, New York (1994)
15. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MISQ 26(2), xiii–xxiii (2002)
16. Aier, S., Gleichauf, B.: Towards a Sophisticated Understanding of Service Design for Enterprise Architecture. In: Feuerlicht, G., Lamersdorf, W. (eds.) ICSSOC 2008. LNCS, vol. 5472, pp. 316–326. Springer, Berlin (2009)
17. Erl, T.: SOA: Principles of Service Design. Prentice Hall, Upper Saddle River (2008)

18. Goikoetxea, A.: A Mathematical Framework for Enterprise Architecture Representation and Design. *IJITDM* 3(1), 5–32 (2004)
19. Chen, D., Lillehagen, F.: Enterprise Architectures - Review on Concepts, Principles and Approaches. In: Sobolewski, M.W., Cha, J. (eds.) *Proceedings of the 10th International Conference on Concurrent Engineering (ISPE CE 2004)*, pp. 1211–1216. Tsinghua University Press, Beijing (2004)
20. Balabko, P., Wegmann, A.: Systemic Classification of Concern-Based Design Methods in the Context of Enterprise Architecture. *ISF* 8(2), 115–131 (2006)
21. Wilkinson, M.: Designing an ‘Adaptive’ Enterprise Architecture. *BT Technology Journal* 24(4), 81–92 (2006)
22. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*, 2nd edn. Addison-Wesley Longman, Reading (2003)
23. Romme, A.G.L., Endenburg, G.: Construction Principles and Design Rules in the Case of Circular Design. *Organization Science* 17(2), 287–297 (2006)
24. Goold, M., Campbell, A.: *Designing Effective Organizations. How to Create Structured Networks*. Jossey-Bass, San Francisco (2002)
25. Reichtin, E.: The Art of Systems Architecting. *IEEE Spectrum* 29(10), 66–69 (1992)

# Something Is Missing: Enterprise Architecture from a Systems Theory Perspective

Sebastian Kloeckner and Dominik Birkmeier

University of Augsburg, Universitaetsstr. 16, 86159 Augsburg  
{sebastian.kloeckner,dominik.birkmeier}@wiwi.uni-augsburg.de

**Abstract.** Enterprise modeling has been an area of significant research in the information systems discipline throughout the last decade. Mainly developed by IT-practitioners, enterprise architectures (EA) became a promising and comprehensive approach to model either the current or desired state of enterprises. Existing approaches are, however, often criticized for paying too little attention to the business side of enterprises. In this paper, we interpret an enterprise as socio-technical system and analyze from a systems theory perspective which features are necessary for a comprehensive model. From there, we deduce if, why and how additional aspects of enterprises should be included into EA. Amongst others, it becomes obvious that especially human actors, as most flexible and agile elements of enterprises, are not adequately included in current architectures. We therefore present first ideas for integrating this important aspect into EA, the corresponding implications of such an inclusion, as well as several areas of further research.

**Keywords:** enterprise architecture, systems theory, socio-technical systems.

## 1 Introduction

Enterprise modeling in general and enterprise architectures (EA) in particular have gained significant momentum in research (e.g. [1, 2]) and practice (e.g. [3]) during the last years. Especially the promise to make the important elements of an enterprise and their relations visible makes it an interesting concept for the analysis and design of complex business systems. A comprehensive enterprise architecture therefore specifies, amongst others, the goals and strategies of an enterprise, its business processes as well as the associated resources like production systems, information systems and humans [4]. While the former aspects are often included in current concepts of EA, especially humans, as integral parts of enterprises, are often not taken into consideration. But only such a complete picture would essentially support necessary transformations of organizations in a flexible and agile way.

First being discussed in the 1970s, several enterprise architecture concepts, often under different names, were proposed over the time (e.g. [2, 5-7]). Today the Zachmann-Framework [8], The Open Group Architecture Framework [9] and the Federal Enterprise Architecture [10] can be seen as the most widespread frameworks for enterprise architectures. But as most of these concepts are rooted in the IT-departments of today's enterprises, they are oftentimes strongly focused on IT-related aspects.

Although business issues are slowly moving into research focus [11, 12], few attentions have so far been paid by the information systems science community to organizational aspects. When looking at the complete picture, enterprises are socio-technical systems and therefore do not only contain technical components, but also humans and the organizational context [13]. Especially this organizational context can have a significant impact on the overall success and is one of the main reasons of budget overruns or even complete failings, as stakeholders are resistant to change or do not adopt new technologies. Dietz and Hoogervorst [14] consider the traditional *black-box* thinking based knowledge, i.e., knowledge about the function and the behavior of enterprises, as one of the reasons for such problems. While being sufficient for managing an enterprise within its current range of control, this kind of thinking is inadequate when an enterprise has to change. For such cases, a *white-box* approach, describing the construction and operation of enterprises, is needed.

In this paper, we therefore take an argumentative-deductive approach to analyze from the holistic perspective of systems theory, in particular Ropohl's *Theory of General Technology* [15], where certain aspects are missing in the current concept of enterprise architectures. Furthermore, we present first ideas how these missing parts can be integrated and which benefits could be realized by such an integration.

In the remainder of the paper we will proceed as follows: after presenting related work and motivating the research gap in the next section, we introduce relevant theories and concepts. In detail, we present the common understanding of enterprise in the field of enterprise architectures, as well as the basic concepts of systems theory in general and the Theory of General Technology in particular. By interpreting enterprises as socio-technical systems, different aspects in regard to including human factors into enterprise architecture are deduced in the synthesis. Taking these aspects into account we draw the conclusion that especially an integration of human beings into the lower layers of EA is necessary. In the following section on implications for practice and academia we present that this will further disclose new optimization approaches and cost-saving opportunities. Finally, we sum up our findings, provide an outlook and raise several possible trends for EA.

## 2 Related Work

The evaluation and comparison of EA frameworks in general and their completeness in particular has frequently been addressed in literature. Besides publications focusing on a single framework (e.g. [16]) there are also several extensive comparisons between existing frameworks such as the ones from Leist et al. [17], Bernus et al. [18], and Schekkerman [19]. All of them elaborate on the individual strengths and weaknesses of the common frameworks and try to identify their gaps as well as possibilities for further improvement. Noran [20] furthermore examines the mapping of classical frameworks onto the Generalized Reference Architecture and Methodology (GERAM) Framework [12], which tries to provide a common, but rather abstract, regulation framework for the former ones.

In addition to those comparisons between frameworks, Aier et al. [21] provide a literature survey on established contributions from academia and practice, as well as an empirical examination on comprehension, model building and utilization of enterprise

architectures. They develop a systematic overview on the current understanding and the state of the art of EA. From there, they identify discrepancies between research and practice and discuss corresponding implications for both. Through an empirical survey among practitioners and researchers in the area of EA, they identify, amongst others, which design artifacts have to be considered in an EA and the degree of their realization in practice. From this survey, it can be seen that the interaction with customers and suppliers, roles and responsibilities, as well as organizational units are all considered mainly important, but their degree of realization is generally lower than the average of all design artifacts. On the other side, many of the aspects considered as the most important ones, with the highest degree of realization, are purely technical. Those are, amongst others, interfaces, applications, data structures, software-, hardware- and network-components, etc. It can be seen from the study that, while organizational aspects of enterprises slowly come into focus, enterprise architectures are predominantly shaped by IT-departments and their view of the enterprise.

Furthermore, an analysis of the “impact of service-oriented architecture on enterprise systems” was carried out by Bieberstein et al. [22]. Based on it, Schroth [23] presented his view of a Service-Oriented Enterprise. Both elaborate on the alignment of service-oriented architectures (SOA) with first aspects of existing organizational theories. Bieberstein et al. state that “the SOA paradigm also needs to be extended to transmute organizational structures and behavioral practices” [22] and they thus “propose the Human Services Bus (HSB), a new organizational structure that optimizes the workflow and streamlines cross-unit processes to leverage the new IT systems” [22]. Schroth [23] proposes an approach of mapping the major underlying principles of SOA, namely decentralization, agility, as well as composition and coordination of building blocks, to upcoming forms of organizations. He uses the HSB to allow advertising human services within an enterprise and provide a means for workflow monitoring. While both works consider aspects from organizational theory, they originate from a rather technical SOA context and focus on changes that an integration of SOA implicates for enterprises and so do not extend their findings to a general EA.

In summary it can be said that the issue of evaluating EA approaches has been faced from different perspectives. An extensive analysis and comparison of current frameworks, as well as a critical reflection of differences in research and practice has been utilized to identify goals for the future development. On the other hand, the idea of including aspects from organizational theory has been raised, but was limited to the field of service-oriented architectures. Though, to the best knowledge of the authors there is so far no comprehensive analysis of enterprise architecture in regard to systems theory from a scientific perspective.

### 3 Conceptual Foundations

As the review of the related work has illustrated, current concepts of EA are often focused on IT-related aspects. While understandable as most concepts have their origins in IT-departments, the call for a scientific backing of these concepts and an inclusion of organizational aspects gets louder. In order to get an insight into existing concepts and possible scientific theories, we firstly present the current understanding of enterprise architecture and the basic elements of systems theory.

### 3.1 Enterprise Architecture

As shown before, enterprise modeling is a field of significant research and is widely accepted in science and practice [4, 11]. Over time several different names, perspectives and definitions for enterprise architectures were proposed (e.g. [2, 5-7]). A comprehensive overview of different enterprise modeling approaches with sometimes different perspectives can be found in [21]. For a common understanding of the used terms, we will rely on the following definitions: In the context of EA, The Open Group defines an enterprise as “Any collection of organizations that has a common set of goals and/or a single bottom line. In that sense, an enterprise can be a government agency, a whole corporation, a single department, or a chain of geographically distant organizations linked together by common ownership” [9]. On the other hand, the ANSI/IEEE Standard 1471-2000 defines architecture as the “fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution” [24].

While most of the available modeling approaches conform to these definitions, they diverge in certain aspects due to their respective goals and complexity. Winter and Fischer criticize that “Traditionally, architecture in the information systems is focusing on IT related artifacts [...]” and suggest that “Only when ‘purely’ business related artifacts are covered by EA, important management activities like business continuity planning, change impact analysis, risk analysis and compliance can be supported.” [11]. Based on a literature analysis, they deduce five essential layers and related artifacts of enterprise architectures, which were adapted by Aier, Riege and Winter [21] and are shown in figure 1. These proposed layers are a first step to incorporate organizational aspects into the concept of enterprise architecture, as they include artifacts like organizational goals, organizational units and responsibilities. However, they still do not consider these aspects on the lower layers. When taking the history of enterprise architecture and its roots in the IT-departments of enterprises into account, it becomes understandable that it is, mainly on the lower layers, still strongly focused on IT-related issues. But, this perspective only examines one side of the coin, as it ignores the human contribution to the overall performance.

Going further back into history, it becomes obvious that enterprises and their architectures existed before IT came into play. At this time information systems, in their classical definition as systems for the communication and processing of information, without any or only little technology (letter shoot, dockets, etc.) were already in place. The most important components of such systems and the corresponding architectures were, and probably are, human beings. While literature often states flexibility and agility as primary goals of enterprise architectures (e.g. [25]), many approaches do not take these extraordinary flexible and agile “components” of an enterprise into account. Again, this might be owed to the fact that, until recently, practitioners of IT-departments have been leading in the development of the EA discipline. It therefore seems to be advisable to examine from a scientific perspective *if*, *why* and *how* these important providers of services and functionality should be included into enterprise architecture in general. And as enterprises are often described as systems, the holistic perspective of systems theory in combination with the concept of socio-technical systems seems to be a good starting point for such an examination.



Business Architecture	<ul style="list-style-type: none"> <li>•Products/Services</li> <li>•Market segments</li> <li>•Organizational goals</li> </ul>	<ul style="list-style-type: none"> <li>•Strategic projects</li> <li>•Relationship to customers</li> <li>•Relationship to suppliers</li> </ul>
Process Architecture	<ul style="list-style-type: none"> <li>•Business processes</li> <li>•Organizational units</li> </ul>	<ul style="list-style-type: none"> <li>•Responsibilities</li> <li>•Information Flows</li> </ul>
Integration Architecture	<ul style="list-style-type: none"> <li>•Applications</li> <li>•Application clusters</li> <li>•Enterprise services</li> </ul>	<ul style="list-style-type: none"> <li>•Integration systems</li> <li>•Data flows</li> </ul>
Software Architecture	<ul style="list-style-type: none"> <li>•Software services/components</li> <li>•Data structures</li> </ul>	
Infrastructure Architecture	<ul style="list-style-type: none"> <li>•Hardware components</li> <li>•Network components</li> </ul>	<ul style="list-style-type: none"> <li>•Software-Platforms</li> </ul>

**Fig. 1.** Proposed layers and artifacts of Enterprise Architecture [21]

### 3.2 Socio-technical Systems

Systems theory is an interdisciplinary theory, which focuses on the description and analysis of any system as an integral whole that consists of interconnected elements (parts) and their relations. The roots of modern systems theory can be traced to Bertalanffy's General Systems Theory [26]. Mainly based on these works, Ropohl developed a Theory of General Technology [15] for explaining and analyzing elements and relationships of socio-technical systems. Socio-technical in this context refers to the interrelatedness of social and technical aspects of a system, e.g. human beings and technical artifacts in an organization. In his depictions, Ropohl first distinguishes between three different system perspectives: the functional, the structural and the hierarchical view. The functional system concept regards the system as a black-box, which is characterized by certain relations between in- and output properties that can be observed from outside. The structural system concept views the system as a whole, consisting of interrelated elements. And the hierarchical system concept finally emphasizes the fact that parts of a system can be considered as systems themselves and that the system itself is part of a larger system.

Based on these definitions, Ropohl develops a general model of an action system, which is characterized by three sub systems: the goal setting system (GS), the information system (IS) and the execution system (ES). The ES obtains material and energetic attributes and performs the basic work. In the context of business information systems, the mentioned material has to be interpreted as data and information, which have to be processed. The IS handles informational attributes. It absorbs, processes, and passes information and deduces instructions for the execution system from this information. The GS creates the system's internal goals as maxim of action. By introducing the concept of division of labor, Ropohl then further decomposes the initially "monolithic" system from a different perspective. As each action can consist of more than one sub-action, united in one virtual or real system, these sub-actions can also be disassembled into own systems. These new subsystems then have to be linked and coordinated in order to fulfill the formerly united action. The combination of the functional decomposition of a single action and the division of labor (chain of action, resp. workflow) is shown in figure 2.

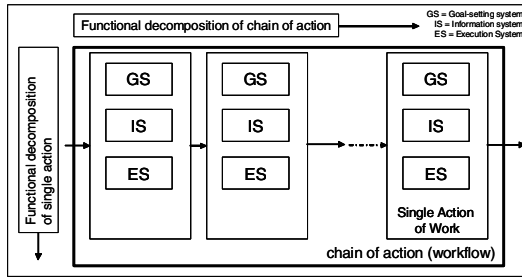


Fig. 2. Functional decomposition of single action and chain of action

Based on these two perspectives of decomposition, Ropohl then develops the concept of socio-technical division of work, shown in figure 3. Ropohl considers Adam Smith [27] to be the first, who said, while in a different context, that machines are similar to humans in many ways and that machines are small systems, created to cause certain movement effects.

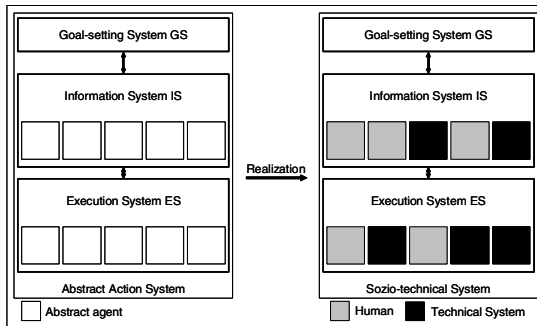


Fig. 3. Socio-technical division of work (adapted from [15])

The abstract action system, depicted on the left side of figure 3, contains an abstract agent as the agent of action. This system can be interpreted as a human being which incorporates all subsystems or as an organization, where several actors are in charge of certain sub-actions. When interpreted as an organization and transferred into reality the single sub-actions of the respective subsystems can be either assigned to human actors or to technical artifacts. Therefore, a socio-technical system is defined as an action or work system, which is composed of human beings and technical systems (artifacts). But while technical artifacts have clearly defined system boundaries, human individuals have additional properties, exceeding their predefined roles in the system, due to their social and cognitive capabilities (e.g. social networks, creativity, problem solving, etc.).

## 4 Synthesis

A comparison of the basic structures of enterprises and the theory of socio-technical systems shows that many similarities exist. An organization composed of humans and

artifacts, like an enterprise, must therefore be interpreted as a socio-technical system, whose structure, as stated in ANSI/IEEE Standard 1471-2000 [24], is described by its architecture. Its actions are realized by humans and/or technical artifacts and from an abstract point of view, the different management and execution layers of enterprises can directly be mapped to the different subsystems of the functional decomposition: the goal-setting system is realized by the executive managers, the information system by the middle managers and/or information systems and the execution systems is materialized by the workers and/or production systems. Due to the recursive structure of systems theory, each of these three subsystems can again be interpreted as system and therefore consists of the three subsystems itself, which are realized by humans and/or artifacts.

Looking at the previously described layers and artifacts of EA, many of the proposed elements also fit with the concept of socio-technical systems: The business architecture with its goals, strategic projects, desired products, targeted market segments, etc., models the goal-setting system of the enterprise. The process architecture, with its business processes, information flows and responsibilities, etc., as well as the integration architecture, containing applications, enterprise services, integration systems, data flows, etc., mirrors the idea of the information system. The software and infrastructure architectures can finally be mapped onto the execution system.

As shown, there are obvious similarities between EA and socio-technical systems and as enterprises are often characterized as socio-technical systems, it seems to be reasonable to further analyze the concept of enterprise architecture under the perspective of the Theory of General Technology. Especially, as it also becomes evident that the respectively considered elements, subsystems and relations seem to be different.

Considering the literature on EA, it first becomes apparent that EA models are used for two different purposes: Many publications about enterprise architectures differentiate between descriptive and design models (e.g. [3, 4, 16]). Descriptive models, often called “As-Is-Models”, illustrate the state of an enterprise in its current situation and how business is executed. Design models, regularly labeled as “To-Be-Models”, envision a future state of the enterprise and how it should be shaped in the future. As the purpose of these models significantly differs, it becomes obvious that they are in fact the outcome of two different types of systems. The first system, which has the descriptive model as result, can be named *run-time system*, the second, producing the design model, should be called *design system*. These two systems are not independent of each other, though. The design system pictures a future state, which shall be implemented in the run-time system. The existing run-time system on the other side is usually the basis for a new design, as complete green field approaches are typically not possible in practice. These two systems themselves contain socio-technical sub-systems. Each of these subsystems realizes certain actions, which can or cannot be fulfilled by humans and/or technical artifacts as shown in table 1. In addition, each type of agent (human being and technical artifact) has distinct properties, which make them better suitable for a certain task.

During design time, the goal-setting-subsystem can only be realized by humans, since the creation of goals, targets, decisions and strategies needs creativity, which cannot be realized in technical artifacts. While a support by technical artifacts is possible, e.g. decision support systems, a replacement of humans by technical artifacts is not possible. The information subsystem of the design system has to define the necessary flows and tasks in order to make a realization of the defined goals and targets possible.

While first approaches are made to at least partly automate the action of this subsystem (e.g. [28]), it is unlikely that it can be fully automated in the near future. Finally, the execution subsystem of the design system creates (make) or acquires (buy) the necessary resources to fulfill the tasks defined by the information subsystem. While this action is usually realized by humans, several approaches, like code generators or automated market places, try to computerize it.

**Table 1.** Different actions of sub systems at design and execution time

	Design System		Run-Time System	
	Action	Agent	Action	Agent
<b>GS</b>	(Re-)Define targets of system and actions	Humans only	Control and/or set goal of executed action	Humans and technical artifacts (goals are implicitly set)
<b>IS</b>	(Re-)Define necessary flows and tasks to achieve goals	Mostly humans, sometimes technical artifacts	Execute control flow or function templates, call necessary functions	Humans and technical artifacts
<b>ES</b>	Create and acquire necessary flow and function templates	Humans and technical artifacts	Realize and execute certain function	Humans and technical artifacts

The goal-setting subsystem of the run-time system can be realized by humans or by technical artifacts. In the case of humans, they will follow the plans, which were defined during design time. If certain situations were not considered during design time, they are able to (re-)define what has to be done and how to react to certain influences from the environments in a flexible as well as agile way. For technical artifacts the goal-setting subsystem exists only implicitly. By defining its functionality during design time, its goals are irrevocably set. If certain system-external or -internal influences or states were not considered at design time, e.g. by catching errors, a technical artifact will malfunction, e.g. an error occurs. An independent redefinition of goals and a corresponding reaction is, while being researched, not possible yet. According to the defined goals, the information subsystem executes the created or acquired flow and function templates. It controls, depending on the input and based on these templates, which agents, humans or technical artifacts, have to be called to fulfill a certain action. While humans can independently switch to comparable functions if a needed subsystem is not available (e.g. different supplier of comparable pre-products), this is, due to the normally missing capability of comparing unspecified possibilities, usually not possible for technical artifacts. The best chance for technical artifacts, if considered during design time, is to keep state until the required subsystem becomes available. The execution subsystem finally executes the defined tasks. It takes the input and transfers it into the output. The actions of the execution subsystem can, due to the hierarchical concept of systems theory, themselves be complex systems. They can be realized by an enterprise, like a sub-contractor, by technical artifacts, like web services or by human beings. Now, taking into account that the results of the information subsystem of the design system are used by the goal-setting subsystem of the run-time system and that the outcomes of the execution subsystem at design time are utilized by the information subsystem at run time, the different subsystems can be connected as shown in table 2. In addition, it can again be shown that the different subsystems can also be mapped onto the proposed layers of EA.

**Table 2.** Mapping of socio-technical subsystems to proposed layers of enterprise architecture

Layers of EA	Design System	Run-Time System
Business Architecture	(Re-)Define targets of system and actions	---
Process Architecture	(Re-)Define necessary flows and tasks to achieve goals	Control and/or set goal of executed action
Integration Architecture	Create and acquire necessary flow and function templates	Execute control flow or function templates, call necessary functions
Software Architecture	---	Realize and execute certain function
Infrastructure Architecture	---	---

## 5 Consequences of the Synthesis

Taking this into account, several consequences for and gaps of enterprise architecture can be deduced. While the higher layers still contain the human aspect of socio-technical systems, they disappear on the lower layers like integration, software and infrastructure architecture. Although it is understandable for the infrastructure level, where offices and buildings, walks and streets would correspond to hardware and network components, it has to be questioned, if this is comprehensible for the integration and software layer. When looking into today's enterprises, human beings regularly execute functions of these layers: For example, whenever a direct information system integration is not possible, due to technical restrictions, human beings or teams connect the two distinct application systems over their graphical user interfaces and act as human connectors between these systems. But, as these human connectors are usually very costly compared to technical interfaces, information about them and their integration into the enterprise architecture would help to optimize the enterprise. In this context, human teams could be interpreted as enterprise services which offer certain business and integration functionality to other elements of the system.

When looking at the layer of software architecture with its software components and data structures, again, the general tasks are sometimes carried out by human beings. E.g., humans often carry specialized data about certain customer preferences or realize tasks which require a certain portion of creativity. Both are characteristics which are important for successful enterprises and cannot easily be realized by technical artifacts. When these qualities are overseen, just because they were not part of the enterprise architecture, for example when replacing sales clerks by web front ends or call centers, the overall performance of the enterprise can take severe damage. In this context, human actors could be understood as software or elementary services which carry data or realize certain specialized functionality. The breakdown or loss of such system elements and the corresponding implications can be compared with those of software services, although the latter are often easier to fix.

It therefore seems to be more than reasonable to integrate human beings or aggregations of those (e.g. teams) as components or service providers with certain interfaces into the lower layers of enterprise architecture, as shown in figure 4. Only if human beings, as integral part of the socio-technical system "enterprise" are included, profound management and alignment decisions become possible.

Business Architecture	<ul style="list-style-type: none"> <li>•Products/Services</li> <li>•Market segments</li> <li>•Organizational goals</li> </ul>	<ul style="list-style-type: none"> <li>•Strategic projects</li> <li>•Relationship to customers</li> <li>•Relationship to suppliers</li> </ul>
Process Architecture	<ul style="list-style-type: none"> <li>•Business processes</li> <li>•Organizational units</li> </ul>	<ul style="list-style-type: none"> <li>•Responsibilities</li> <li>•Information flows</li> </ul>
Interaction Architecture	<ul style="list-style-type: none"> <li>•Application (clusters)</li> <li>•Enterprise services</li> <li>•Human teams</li> </ul>	<ul style="list-style-type: none"> <li>•Integration systems</li> <li>•Human connectors</li> <li>•Data flows</li> </ul>
Execution Architecture	<ul style="list-style-type: none"> <li>•Software services</li> <li>•Human Service</li> </ul>	<ul style="list-style-type: none"> <li>•Data structures</li> <li>•Human Experts</li> </ul>
Infrastructure Architecture	<ul style="list-style-type: none"> <li>•Hardware components</li> <li>•Network components</li> </ul>	<ul style="list-style-type: none"> <li>•Software-Platforms</li> </ul>

Fig. 4. Integration of human factors into proposed layers and artifacts of EA

In consequence and in order to mirror the inclusion of human beings and technical artifacts properly, the name of the integration and software architecture should be changed: Instead of using the word *integration*, the word *interaction* would better reflect the interplay between humans and technical artifacts. And *execution* architecture better characterizes the realization of basic services, regardless of whether they are provided by humans or technical artifacts.

## 6 Implications for EA in Practice and Research

The integration of the human or social aspect into enterprise architecture has several implications for the successful management of enterprises and the focus of future research. First, it would allow the concept of enterprise architecture to become a comprehensive model of all relevant aspects and elements of the enterprise. The white-box perspective of systems theory has shown that each subsystem of the supersystem enterprise is usually a socio-technical system and omitting one important element of this system would be negligent and carries the risk of wrong decisions.

Second, as an increase of agility (adapt to unexpected changes) and flexibility (adapt to expected changes) is one of the primary objectives of the concept of enterprise architecture [25], it seems to be reasonable to include the most agile and flexible components of the socio-technical system enterprise, human beings and organizational groups, into the concept. Especially due to their flexible goal-setting and information system, human beings can quickly adapt to unexpected as well as to expected changes of the environment. A comparably quick adaption of technical artifacts has to be viewed as unlikely. Furthermore, an a priori integration of solutions for unexpected changes is impossible, as they can, by definition, not be foreseen. Therefore, the inclusion of human beings into the concept of enterprise architecture would significantly increase the level of usable resources.

Third, the inclusion of human aspects into enterprise architecture would be the first step to the often demanded integration of concepts from organizational science (e.g. [13]). In addition, an integration of human and organizational aspects would reveal areas, where organizational science or information systems science tries to solve comparable questions, which were already answered by the neighbor discipline. For example, while the component and service identification community (e.g. [29, 30]) still

tries to find the best-suited solution for the grouping of technical artifacts (granularity), the organizational science already gave answers to these questions in the early 1970s (e.g. [31, 32]). Under the name “departmentalization” the authors utilize affinity analysis to create units with maximal cohesion and minimal dependencies.

Forth, if humans are considered as part of the system, the role of information system technology can change: Instead of information technology being part of human-controlled systems, humans can become part of technical-controlled systems, e.g. in the case where the process flow (information system of the run-time system) is controlled by the IT-system and humans fulfill certain tasks as part of the execution system (at execution time). While this will be bought at the cost of an asynchronous execution, as humans are slower than usual time-outs of IT-systems, it will offer new functionality and realization potentials. And perhaps this shift in perspective could even result in a change of existing programming paradigms.

Finally, only the integration of humans into the concept of enterprise architecture will shed light on spots, where new technical artifacts would allow significant cost reductions. Based on activity-based costing [33] for formerly manual tasks, the amount of cost-savings can be exactly calculated and used as argument for additional investments in information technology. On the other side, there may also occur situations, where technical artifacts should be replaced by humans; for example due to their higher agility and flexibility or when the total costs of ownership (TCO) exceed the costs of labor. In order to make such decisions, a new measure like *Total cost of IT usage*, as an aggregate of TCO and costs of labor, might become reasonable.

## 7 Conclusions and Further Research

In this paper we analyzed the concepts of enterprise architecture from a systems theory perspective. By interpreting enterprises as socio-technical systems and under usage of the Theory of General Technology, we deduced that the concept and use of enterprise architectures implicitly includes two different systems: The design system and the run-time system. These two systems fulfill different purposes and therefore require different capabilities. Due to these requirements, it becomes obvious that humans and groupings of them play an important role in today’s enterprises and the negligence of their actions and properties results in incomplete models. Especially, as humans are the only components in the system “enterprise”, which are able to act in an agile way. While we did not explicitly name service-oriented architectures, the interpretation of humans as service providers would also perfectly integrate into the paradigm of SOA and e.g. in fact supports the realization of a Human Service Bus [22]. But the inclusion of the human factor into the concept of EA also creates several open questions:

- What shift in perspective is necessary, if humans, as active and passive elements of enterprises, in opposition to technical artifacts, as solely passive components, are included into enterprise architectures?
- How can humans and technical artifacts be included into identification approaches?
- How can additional capabilities of human individuals (e.g. social networks), not directly connected with their role in the organization, be reflected in models?
- What side-effects are created by the inclusion of the human factor?

- What measures are necessary to decide if the replacement of humans by technical artifacts or vice versa is reasonable?
- Do situations exist, where the replacement of technical artifacts by humans is reasonable and how can the proportion between humans and technical artifacts be optimized?

When further applying the white-box perspective of systems theory, many other questions could also be taken into consideration. However, already the inclusion of human actors into EA models will shed light onto those actually unobserved areas of the socio-technical system enterprise. In order to answer these questions, further research, especially from a holistic, scientific perspective is needed. Only if the needs of today's practice and the systematic procedures of science are combined, really comprehensive models and methods can be created.

## References

1. Arbab, F., Boer, F.d., Bonsangue, M., Lankhorst, M., Proper, E., Torre, L.v.d.: Integrating Architectural Models - Symbolic, Semantic and Subjective Models in Enterprise Architecture. *Enterprise Modelling and Information Systems Architectures 2* (2007)
2. Frank, U.: Multi-perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: *HICSS 2002*, vol. 3, p. 72. IEEE Computer Society, Hawaii (2002)
3. Gaertner, W.: Ansatz für eine erfolgreiche Enterprise Architecture im Bereich Global Banking Division/Global Transaction Banking IT and Operations der Deutschen Bank. *Wirtschaftsinformatik 46*, 311–313 (2004)
4. Sinz, E.: Unternehmensarchitekturen in der Praxis – Architekturdesign am Reißbrett vs. situationsbedingte Realisierung von Informationssystemen. *Wirtschaftsinformatik 46*, 315–316 (2004)
5. Scheer, A.-W., Schneider, K.: ARIS — Architecture of Integrated Information Systems. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) *Handbook on Architectures of Information Systems*, pp. 605–623. Springer, Heidelberg (2006)
6. Johnson, P., Ekstedt, M.: Enterprise Architecture: Models and Analyses for Information Systems Decision Making. *Studentlitteratur*, Pozkal (2007)
7. Ferstl, O.K., Sinz, E.J.: Modeling of Business Systems Using SOM. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) *Handbook on Architectures of Information Systems*, pp. 347–367. Springer, Heidelberg (2006)
8. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal 26*, 277–293 (1987)
9. The Open Group: TOGAF "Enterprise Edition" Version 8.1 (2002)
10. United States Office of Management and Budget: FEA Consolidated Reference Model Document Version 2.3 (2007)
11. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. *Journal of Enterprise Architecture* (2007)
12. IFIPFAC Task Force: GERAM: Generalised Enterprise Reference Architecture and Methodology Version 1.6.3 (1999)
13. Picot, A., Baumann, O.: The Relevance of Organisation Theory to the Field of Business and Information Systems Engineering. *Business & Information Systems Engineering 1* (2009)



14. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise ontology in enterprise engineering. In: ACM Symposium on Applied Computing. ACM, Fortaleza (2008)
15. Ropohl, G.: Allgemeine Technologie. Eine Systemtheorie der Technik, Hanser, München/Wien (1999)
16. Noran, O.: An analysis of the Zachman framework for enterprise architecture from the GERAM perspective. *Annual Reviews in Control* 27, 163–183 (2003)
17. Leist, S., Zellner, G.: Evaluation of current architecture frameworks. In: 2006 ACM symposium on Applied computing, Dijon, France, pp. 1546–1553 (2006)
18. Bernus, P., Nemes, L., Schmidt, G. (eds.): Handbook on Enterprise Architecture. Springer, Heidelberg (2003)
19. Schekkerman, J.: How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework, Trafford, Victoria, BC (2004)
20. Noran, O.: A Meta-Methodology for Collaborative Networked Organisations. Griffith University, Brisbane (2004)
21. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. *Wirtschaftsinformatik* 50, 292–304 (2008)
22. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. *IBM Systems Journal* 44, 691–708 (2005)
23. Schroth, C.: The Service-Oriented Enterprise. In: TEAR 2007, St. Gallen, Switzerland (2007)
24. IEEE Computer Society: Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Std 1471-2000, New York (2000)
25. Schelp, J., Aier, S.: SOA and EA - Sustainable Contributions for Increasing Corporate Agility. In: HICSS 2009. IEEE Computer Society, Waikoloa (2009)
26. Bertalanffy, K.L.v.: General System theory: Foundations, Development, Applications. George Braziller, New York (1976)
27. Smith, A.: An inquiry into the nature and causes of the wealth of nations (1776)
28. Heinrich, B., Bewernik, M.-A., Henneberger, M., Krammer, A., Lautenbacher, F.: SEMPA – Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen. *Wirtschaftsinformatik* 50, 445–460 (2008)
29. Albani, A., Overhage, S., Birkmeier, D.: Towards a Systematic Method for Identifying Business Components. In: Chaudron, M.R.V., Szyperski, C., Reussner, R. (eds.) CBSE 2008. LNCS, vol. 5282, pp. 262–277. Springer, Heidelberg (2008)
30. Aier, S., Winter, R.: Virtuelle Entkopplung von fachlichen und IT-Strukturen für das IT/Business Alignment – Grundlagen, Architekturgestaltung und Umsetzung am Beispiel der Domänenbildung. *Wirtschaftsinformatik* 51 (2009)
31. Müller-Merbach, H.: OR-Ansätze zur optimalen Abteilungsgliederung in Institutionen. In: Kirsch, W. (ed.) Unternehmensführung und Organisation, Wiesbaden, pp. 93–124 (1973)
32. Kieser, A.: Abteilungsbildung. In: Frese, E. (ed.) Handwörterbuch der Organisation, Poeschel, Stuttgart, vol. 3, pp. 57–72 (1992)
33. Peacock, E., Tanniru, M.: Activity-based justification of IT investments. *Information and Management* 42, 415–424 (2005)

# A Reassessment of Enterprise Architecture Implementation

Stephan Aier and Joachim Schelp

Institute of Information Management, University of St. Gallen  
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland  
{stephan.aier, joachim.schelp}@unisg.ch

**Abstract.** Aside of day-to-day business in some organizations Enterprise Architecture (EA) seems to be successful while it is not in others that also have notations, models, methods, and even dedicated EA tools. In order to understand these differences we have analyzed the development of EA in six companies over the last eight years. Our analyses showed, that apart from formal structure and processes (i) training and education of architects and non-architects, (ii) improving architects' communication skills, (iii) intensifying EA representation in projects, and (iv) tool support (not replacements with tools), significantly contribute to long term EA success.

## 1 Introduction

Enterprise Architecture (EA) is a still developing discipline. It includes rich modeling approaches as well as procedure models and tools. Both contributions from practitioners [see e.g. 1] and research [see e.g. 2] are numerous and show both progress and growing maturity. From a research point of view the scope of EA and EA management seems to be well-defined by now [2], although terminology is still varying. For practitioners it seems to be a unique opportunity to get a grip on the ever growing complexity of their application landscapes.

Standardization efforts like TOGAF [3] offer means to establish EA processes. Corresponding efforts e.g. like CobIT [4] increase governance maturity and ensure well defined interfaces to related IT management fields—governance, IT strategy, business/IT alignment etc. However, although EA is spreading among practitioners, it still seems to be immature in practice. With all processes set up according to the standards and modeling and maintaining most artifacts as recommended by EA management approaches: does EA work and will it survive in any company? Some of the companies, which established EA and EA management processes in the past, are now facing obstacles when practicing EA and are struggling to redefine their EA management in a way making it more sustainable.

Therefore the research question in this contribution is: What are factor combinations for successful EA implementation beyond the mere notion of maturity? As a basis of our analysis we will employ a description framework which has been developed in the course of various practitioners' workshops over the last eight years. Based on this description framework we will analyze six cases and discuss why certain companies have been rather successful in implementing EA while others did not leverage

their EA invest. The analysis will show that EA success is not necessarily a matter of maturity of a number of EA functions but a complex set of factors that have to be observed for implementing EA. Also there is no perfect set of EA factor combinations guaranteeing successful EA because EA always is part of a complex socio-technical network. However, we will identify successful factor combinations as well as common patterns prohibiting EA success.

## 2 State of the Art

Following the architecture definition of ANSI/IEEE Std 1471-2000 EA can be understood as the “fundamental organization of a government agency or a corporation, either as a whole, or together with partners, suppliers and/or customers (‘extended enterprise’), or in part (e.g. a division, a department, etc.) as well as [...] the principles governing its design and evolution”[5]. EA is positioned ‘between’ business and IT and has to serve both; and it is regarded as a mechanism for contributing to agility, consistency, compliance, efficiency, and sustainability [6].

In their extensive literature review covering both academic and practitioner EA frameworks [2] identified seven major EA research groups with four of them covering nearly all EA artifacts, providing procedure models, having their framework implemented in a tool, and have cases at hand where companies used their framework to configure their EA frameworks. See [2] for a detailed discussion of all approaches and a tabular overview.

Four approaches can be considered to represent the state of the art in EA research, which stem from Telematica Institute Enschede (The Netherlands), University of St.Gallen (Switzerland), TU Berlin (Germany), and KTH Stockholm (Sweden). The remaining three projects are from University of Lisboa (Portugal) [e.g. 7], Federal Technical University in Lausanne (Switzerland) [e.g. 8], and TU Munich (Germany) [9]. These three are less evolved than the other four approaches in terms of EA artifact coverage, procedural models, tool support, and application in practice.

The ArchiMate project [e.g. 10] is a Dutch research initiative comprising participants both from academia and practice. It is led by Telematica Institute with links to the University of Twente and the University of Nijmegen, The Netherlands. ArchiMate is a method to manage and communicate EA issues between the different involved stakeholders. It is based on an analysis of existing architecture frameworks and uses views and viewpoints as its core method, closely following the IEEE 1471 standard [11]. The derived modeling language is clearly defined and is used to develop an EA toolset. A procedure model is developed in order to employ a modeling language and a modeling tool in EA communication and EA decision processes. Its roots being conceptual, several case studies have been documented [e.g. in 12] which show the applicability of the different aspects of the method.

At University of St.Gallen, the Business Engineering Framework was developed since the early 1990ies [13]. While the focus was on business process engineering in the beginning, in 2003 publications start to extend the Business Engineering Framework to EA. Explicit meta-models and procedure models exist. The method is developed using conceptual, case study and empirical foundations. The method components are developed in cooperation with companies which are employing the resulting method chunks. A toolset based on the meta-model has been developed and deployed in practice.

Another EA approach has been proposed by a group at the TU Berlin [e.g. 14]. Based in the enterprise application integration (EAI) discussion, they develop a method to establish an EA model. The documentation of the meta-model and the procedure model in the publications is sparse, but the meta-model is used to develop an EA modeling tool. The development approach is conceptual, but uses case studies and some surveys as well.

At KTH Stockholm, an EA framework has been developed which is rooted in views and viewpoints [e.g. 15]. The language work is made explicit very thoroughly, as well as the procedure model. Most of the publications are conceptual with extensive literature analysis, but case studies can be found as well.

With a set of publications from each group, the terminology with these approaches has been widely standardized. Discussions with representatives from these groups show that glossaries are defined or are currently in development stage in some groups. Some deviations in terminology between individual publications of a group can be assigned to the progress of knowledge gained in the ongoing research process, some minor deviations to the usual clutter introduced by review processes forcing the authors to apply e.g. terminology changes. Regarding scope, most groups have their EA language (i.e. architectural layers, elements included in models, relationships etc.) defined, some to an extent that a corresponding tool could be developed.

All analyzed approaches present EA frameworks comprising EA description languages; some include procedure models for EA management as well. The research methodology employed is essentially design research.

Both meta-models and procedure models define an important part of the terminology used in EA research projects. At least six of the seven approaches covered in this study, sufficiently define their language to make it adaptable by others.

Unfortunately, none of the existing (major) EA approaches stemming from academia is providing a framework to assess the long term success of EA within a company. Therefore we cannot build upon existing literature on this topic, but have to set up a description and research framework.

### **3 Structure of Analysis**

The terminology of our discussion framework is based on the St. Gallen approach, which is rather comprehensive and covers all architectural levels and artifacts that can be identified in companies. Furthermore some of these companies observed are using the framework to structure their architecture and glossaries. Therefore the translation loss can be reduced by adhering to this framework.

The description framework we use to structure our analysis has been developed in a series of research projects and has been challenged by EA practitioners involved in these projects. We conduct three workshops per year with participants from all companies involved in the research projects. In these workshops EA characteristics and factors influencing EA success have been discussed or could be derived as a side result of other EA topics discussed. The participants are employees from the companies involved, most of them either enterprise architects or business analysts, or at least partially involved in corporate architecture topics by having a partial architecture role.

Four groups of factors could be identified over time: (a) Contextual factors describe the general corporate environment. (b) Structural factors describe the architectural power and its impact in the companies. (c) Process characteristics describe the working mode of architectural influence in the companies. (d) Finally we discuss factors influencing the architectural leverage over time.

From the *contextual factors* the size of the enterprise and the resulting number and size of the architecture models is the most obvious. Bigger companies require more and larger models to be described, which translates in larger and more complex EA activities. The market orientation of the IS department—being either a cost center or a profit center—results in a different stance on supplemental activities like architecture. The economic pressure the company is facing adds to this: Cost cutting exercises often influenced the growth of EA activities, or their shrinking respectively. Having a dedicated architecture budget was considered to be relevant as well: it determines if corrections in architecture can be driven by the architecture group or if they are dependent on finding a sponsor within the organization. Furthermore overall strategic alignment, e.g. as discussed in [16], was expected to be relevant for the standing of (enterprise) architecture with the company. Finally overall corporate culture was considered important, especially when change occurs.

*Structural factors* describe the architectural power and its impact within a company. Parts of the formal power can be derived from overall governance structures. But the architectural power can be influenced by some more architect centric perspectives as well: Architects may have a formal network. For example in a matrix structure central lead architects may define (strong) or coordinate (weak) what domain architects do. Furthermore they have their personal network in a company, which gives them additional insight in ongoing developments relevant for architecture. The EA organization alone does not describe the formal mode of “selling” EA sufficiently. Enterprise architects can be characterized as EA evangelists or EA police in any organizational structure. Their impact depends on their recognition as an ivory tower or on being credible. This may depend on the individual architect’s history, their peers within the organization, and the architecture education. The later can be distinguished in the architectural knowledge and the architecture skills of architects [17] and non-architects. The willingness to adapt to architecture depends on the visibility and perception of architecture outside the EA department—or the IS department respectively. To structural factors we add both tool support and coverage. EA tools are helpful to foster EA communication by providing (potentially) easy to use models, which are consistent and up-to-date. EA coverage—which architectural levels and artifacts are covered—influences range and number of architecture contacts in an organization.

*Process factors* describe the kind of enterprise architects’ involvement in projects as well as their impact. Changes in architecture are considered to be implemented by projects—either architectural projects or any other kind of projects. Architects’ involvement may be determined by the number of projects, their size and duration and the amount and type of architects’ activities within a project: being involved in some or all projects, being involved in quality gates only or permanently as a project member (with architectural and/or other activities assigned). These factors result in a different impact in projects, which is supported by the instruments available to enact architecture. This factor does not address overall or EA governance instruments (see structural factors), but architectural rules and EA processes defined.

Finally factors could be identified which are relevant for the *architectural influence and impact over time*. First of all amount and frequency of architectural training (further education and training) is to be mentioned. This again can be differentiated in training and education of architects and non-architects. Furthermore EA marketing has to be considered. This includes all activities, which aim at raising awareness for architectural issues, side effects on architecture and effects of architectural deficits.

Table 1 sums up potential EA factors which are used as the description framework when analyzing the case studies in the next section.

**Table 1.** Potential EA factors for EA implementation success

Factor group	Individual factor	Description
<b>Contextual factors</b>	size of company/architecture	size of company; number and size of resulting architecture models used
	market orientation	cost center or profit center
	economic pressure	Are there cost cutting exercises?
	budget	Is there a dedicated (E)A budget?
	strategic alignment	What kind of business/IT alignment exists?
	culture	How does the corporate culture influence change?
<b>Structure</b>	governance	Is there an EA governance and how is it anchored?
	architectural power	How strong are formal and informal architectural power and the resulting impact?
	skills of architects	What skills do architects have?
	skills of non-architects	What are the architectural knowledge and architectural skills of non-architects?
	EA visibility outside the EA department	Are any architectural efforts visible outside the architecture department?
	tools	Is there any EA tool support?
	coverage	What is covered by EA?
<b>Process</b>	project support	How are architects involved in projects in general?
	impact in projects	How do they contribute to projects?
	rules and EA processes	What are the instruments to enact architecture within projects?
<b>EA over Time</b>	training of architects	frequency and amount of architectural training and further education
	training of non-architects	
	EA marketing	“marketing” measures to raise architecture attention and architecture sensibilization

## 4 Case Studies

In the following subsections the case studies of six companies are outlined. These companies have introduced EA functions several years ago and made experiences with the evolution of these architectures. In each case the general situation as well as an outline of the position regarding our structure of analysis is presented. If necessary comparability is provided by translating the individual companies’ terms into the architectural levels defined in [5] covering the entire “business-to-IT” stack.

Data for the case studies have been collected with three of these companies since 2002/2003 and with the remaining three since 2006. Key stakeholders in IT management, architecture management (i.e. IS and business architects), and business/IT relationship management have been interviewed. In addition to the interviews regular review meetings have been set up to observe state, development, and architectural

issues in the companies involved. Two of the companies described participated in long term collaborative research projects in IS integration and enterprise architecture involving ten companies in the period of 2002–2009. The companies chosen for this study have a long term EA experience and have mature architecture management structures in place. Data presented in the case studies below aggregate research results gained with these companies until summer 2009. Due to company request case studies have been made anonymous.

#### **4.1 Company A**

Company A is a major financial service provider in Switzerland primarily focusing on standardized retail banking and transaction processing. Regarding architectural levels all levels mentioned in [5] can be found with broad, defined architecture management processes on IS side. All business related architecture artifacts are managed by an organizational unit directly reporting to the CEO. Alignment of business and IS architectures is explicit and facilitated by personal interweavement by having former IS architects included in the business architecture unit.

Due to the “experimental” positioning of EA on business side, the EA function had a passive role. Their main task was to host the EA repository and to support the integration of existing partial enterprise models (e.g. process models, application landscapes etc.). Also the EA meta model was strictly focused on stakeholder needs and thus was very lean. However, over time this passive set-up also revealed its weaknesses, namely poor coordinative power on interfaces of different stakeholders as well as poor performance in leveraging synergies among various business and IT projects. Therefore the EA function developed a more and more active role, e.g. by being involved in all major change projects by design. Especially the relationship between the EA department and the still existing IT architecture however, became an issue. Both departments address overlapping parts of the EA. While they may have different concerns they redundantly start to define EA processes, functions and also tools.

#### **4.2 Company B**

Company B is one of the top five globally operating pharmaceutical companies. Its structure is dominated by its divisions and partially influenced by the respective countries. Some of the divisions are big and thus very powerful in terms of financial power and resources and are thus independent. Other divisions are rather small and often have to buy resources/services from bigger divisions. In order to leverage the benefits of potential reuse of services or the standardization of platforms and processes an international unit spanning all divisions has been introduced.

EA management understood as leveraging synergies by reusing services, processes and/or platforms is not well implemented on enterprise level, since at least for big divisions there is hardly any economic pressure forcing these divisions into enterprise wide consolidation programs. While big divisions have a fundamental EA management for certain countries in place, the smaller divisions hardly have enough capacity to invest in “housekeeping” projects.

While formal governance structures exist on enterprise level they hardly have any impact on inter-division consolidation. This is mostly due to the fact that budgets are

earned in and allocated to the divisions instead of the global headquarter. As a consequence the entire company has a high level of redundancy in processes and systems resulting in a highly complex EA which makes change projects (which rarely influence only one division) risky and expensive. However, the company has started a number of reengineering projects consolidating at least typical support processes.

### 4.3 Company C

Company C is a globally operating telecommunications service provider with a large, complex and heterogeneous application landscape. At the end of the last century, corporate management decided to structure the group into four rather independent divisions representing the four strategic business areas. The new structure reduced the overall complexity by reducing the interdependencies between the four divisions on a business layer as well as on a technology layer by definition. At the same time, however, the heterogeneity as well as redundancy between the divisions grew as a result of their new independence. This independence resulted in e.g. inconsistent information about customers where different divisions served the same customer segments with different, division-specific products. As a consequence, divisions have been continually integrated again in order to leverage synergies among them.

Company C primarily focuses the entire stack from business models to questions of low-level technologies (e.g. network infrastructure). As a control instance company C has implemented an EA function on group level. The primary means of alignment is a capability model with about 90 top-level capability definitions which are structured in twelve group-wide domains. The EA function provides governance services with rules and processes for the introduction of EA as well as EA compliance assessments for projects. Projects are guided by EA information and expertise to improve compliance. Delivery processes support the actual IT implementation of business concepts. The group wide EA board ensures EA consistency on very aggregated level addressing major change and/or infrastructure projects.

### 4.4 Company D

Company D is a Swiss financial IS solution provider focusing on both developing and running banking software. The company's application portfolio is in a transition from mainframe to a client/server-based service oriented architecture.

Company D has to differentiate two distinct EA perspectives. On the one hand side there is company D's EA ranging from business to infrastructure architecture. On the other hand there are the business and organization architectures of the customers, which have to be mapped to the application architecture provided by company D and the corresponding software and infrastructure architectures.

In the previous year efforts have been started to redesign both architecture teams and architecture processes. Now there is a dedicated EA team defined and business and organization architectures are being rebuilt. The key members of the redesigned architecture teams (both EA and domain architecture teams) have a common (team) background. This is helpful during the organizational redesign phase, because there is a common understanding of the architectural goals and the underlying software and infrastructure architecture. During the organization's redesign EA won a more prominent position with the chief architect being a member of the executive board.



Although organizational changes are not completed, EA is receiving more attention. Furthermore tool support is being rebuilt as well, changing the architecture tool landscape from being more software development focused to an EA oriented tool approach, which is focusing less on software details, but on supporting cross architectural questions—as recommended in [e.g. 18, 19].

#### 4.5 Company E

Company E is a major transportation and logistics service provider in Switzerland. It offers both cargo and passenger transportation and provides rail infrastructure.

A couple of years ago the inauguration of a new CIO resulted in renewed architecture efforts including the foundation of a central EA team. The EA team is complemented by domain architecture teams, which are changing their focus from a domain and software centered perspective to an EA perspective. EA processes have been set up altering existing development processes to reflect architectural issues, e.g. by defining quality gates, which projects cannot surpass without fulfilling strict architectural requirements. This change in processes is fostered by a broad range of efforts to enhance architecture attention, knowledge, and skills throughout the company. Therefore a broad training program was set up. This program spans over twelve days and is focused on EA. More than 60 architecture stakeholders participated in this program by now. From these only parts are central or domain architects. The others are business analysts, (internal) customer, development, or infrastructure representatives.

In addition to the training program further measures were set up. For example (i) architecture communication has been advanced by an EA tool providing a broad set of EA artifacts in an easy-to-use web interface, (ii) all information required to meet architectural guidelines in the quality gates is available through a well-organized intranet web application.

#### 4.6 Company F

Company F is a medium sized insurance company in Switzerland. The market conditions are still comfortable, although competition is increasing, as the local market is saturated. Architecture has a long history in this company, therefore one of the oldest EA teams can be found with company F. EA team size is about 20 architects; with an overall IS staff of about 200.

Architecture is well received in the company. Architects have to spend half of their time in IS projects, and can only spend the other half on pure architectural tasks. All IS projects are facing strict architectural quality gates. However, they are supported from the early design stages on by an architecture bureau, which offers any support required to enhance architecture in a project. Furthermore all projects have to spend a certain percentage of the project budget on architectural issues. But they are free to choose the architect to participate in their project. Company F is investing in architectural training as well. On the business side the reception of business architects is very high: Although being part of IS some business architects are considered to be part of the business side. Although the quality gates are strict, overall formal architectural power is not that strong. With company F they are very influential due to personal expertise and focus on communication skills. Furthermore EA team members are creative in initiating and spinning architectural concepts outside the EA department. The focus is on keeping architectural attention and awareness high.

## 5 Analysis of Cases

Cases A and D have to be handled carefully in the subsequent analysis. These companies participated actively in the research project, in which the factors were identified. Although companies A and D joined the project after the factors have been shaped, their contributions had influence in the adjacent discussions within the research project. Using cases A and D in this paper could be seen critical, because the framework could be considered not to be independent of these two cases. However, it is, and we will respect this issue in the discussion below.

**Table 2.** Contextual factors influencing the EA implementation

Contextual factors	Company A	Company B	Company C	Company D	Company E	Company F
size of company/ architecture	medium	large	large	medium	large	medium
market orientation	profit center	cost center	cost center	moving to profit-center	cost-center	profit-center
economic pressure	medium	low	medium	high	medium	medium
budget	available with business case	distributed over divisions	fixed budget + project related	fixed budget + project related	project-related	project-related
strategic alignment	explicit coverage of business and IT	only with limited focus on divisions, products, countries	explicit coverage of business and IT	changing due to market changes	weak	business driven architecture
culture	strong business focus on EA	strong focus on marketing of products	affinity to IT	strong affinity to IT	strong technical affinity	Strong acceptance

From the contextual factors (table 2) the combination of strategic alignment and culture seems interesting. Those companies with higher (more explicit) alignment and either an EA aware culture or some technical affinity in the overall staff, EA awareness seems to be higher. A high economic pressure as with company D is, as usual, fostering changes; in this case in favor of EA.

The impact of EA in the organization seems mostly influenced by architectural skills of non-architects and their reception of architects. Central governance for instance is important for aligning the different EA activities; however, governance only seems to work effectively if EA becomes more than a central service of the headquarter. High credibility of architects and continuous training of and/or communication to non-architects proved to be important for EA impact with companies E and F.

Tool support per se is not helpful, but making access easy and selecting EA artifacts carefully for web publication with stakeholders in mind proved to be very successful with company E. Company D learned this lesson by using tools targeted to meet software engineering requirements and had failed with the business side. Now they are making huge progress by using a dedicated EA tool.

**Table 3.** Attributes of structure describing the EA implementation

Structure	Company A	Company B	Company C	Company D	Company E	Company F
<b>governance</b>	well established, but misalignment in EA/IT arch.	central governance is ineffective	established with growing impact	is being re-established	established with growing impact	established with high impact
<b>architectural power</b>	high credibility, informal network no formal power	business driven projects are dominating	different impact among bus. units	high with established architects	growing credibility and thrust	high credibility
<b>skills of architects</b>	well established	available in large divisions	well established	partly high and growing	partly high and overall growing	High and well-established
<b>skills of non-architects</b>	well established	hardly available	very heterogeneous	heterogeneous	heterogeneous, heterogeneous growing	heterogeneous
<b>EA outside unit tools</b>	well established various tools consolidated into EA repository	hardly available heterogeneous tool landscape	very heterogeneous central EA platform + specialized tools	heterogeneous central EA tool with growing impact	growing central EA tool with high impact	established different tools, but high impact
<b>coverage</b>	few relevant artifact types ranging from business to IT	very heterogeneous level of EA coverage in the divisions	all major artifact types covered; strong capability model	small but growing	major artifact types and growing	major artifact types and growing

**Table 4.** Attributes of Processes describing the EA implementation

Process	Company A	Company B	Company C	Company D	Company E	Company F
<b>project support</b>	introduced only recently	introduced in a major division	available as an EA service	available via domain architects	medium, growing	high
<b>impact in projects</b>	introduced only recently	in major divisions	in most major projects	being re-established	strong	strong
<b>rules and EA processes</b>	established	in major divisions	established	being re-established	established	established

For all cases active support of projects is a major success factor for EA. This becomes especially visible in case B where the central architecture group is hardly connected with the change projects happening almost exclusively within the divisions. When looking into details EA architects have the biggest impact in company F. Although the benefit of easier communication in a medium sized company has to be discounted, the local arrangements foster both EA acceptance and impact: The inventive incentive structure—projects have to spend on architecture, but can choose the architect, architects have to earn half of their budget in projects—pays off.

Again the education of employees outside the EA department plays a major role for EA success. This becomes obvious in company E where a large EA education program has been started. Company F is stressing communication skills of EA architects, which is addressed in company D as well.

**Table 5.** Factors concerning time describing the EA implementation

<b>EA over Time</b>	<b>Company A</b>	<b>Company B</b>	<b>Company C</b>	<b>Company D</b>	<b>Company E</b>	<b>Company F</b>
<b>training of architects</b>	done occasionally	in major divisions only	done occasionally	done occasionally	regular and extensive	regular
<b>training of non-architects</b>	implicitly only	hardly any	implicitly only	implicitly only	regular and extensive	implicitly only
<b>EA marketing</b>	done informally	hardly any	regularly done	informally	formally and informally	formally and informally

When assessing long term success of individual EA efforts in companies A to F, company F has clearly the best record: Steady education of architects, inventive incentive structures, and ongoing communication has made EA successful for a couple of years now. Therefore it might be realistic to expect—*ceteris paribus*—continuous EA success in company F. With the other companies an outlook is less easy: Company E invested in training and education, which can lay the foundation for future EA success, but efforts are too young, to be judged. With company D changing market conditions are reviving EA again, but similar to company E future success is difficult to judge. With both companies' additional EA efforts—forming/extending of EA groups, (re)building EA processes and governance—are important. With company E training and education will be beneficiary. With company D this cannot be done directly due to company D being a software vendor. Company C as a large international company has a very complex EA. However, they have managed to establish a group wide domain and capability model which have become a central reference for all kinds of major change projects. Their challenge is to continuously promote the benefits of additional models (since they represent abstract elements only) like these. Company B is also a large international company that, however, does not have a comparable culture in EA. The dominant reason seems to be the low economic pressure which did not put the strong divisional separation into question. Company A finally has been a very successful case for several years, because they have been one of the first and still are one of the few companies that have located EA on the business side of the organization. Therefore they have implemented some rather innovative EA use cases like project assessment and achieved a high acceptance rate outside the IT department. However, today they struggle with exactly this separation from IT since IT has developed overlapping architecture services.

To sum up observations regarding long term success of EA, we recognize that additional efforts beyond formal structures and processes are required. Especially it seems to be important to stress communication and training/education: Communication skills of architects have to be enhanced and regular training and education of architects is most important. Training and education of non-architects fosters the acceptance of architectural issues and reduces barriers.

To improve EA impact a continuous representation in projects is required. Especially incentive structures fostering this on both architecture and project side might be helpful. In addition, tool support can enhance communication, but it cannot replace it. For an EA being appreciated by business and IT additional abstract models like domain models and capability models provide a strong benefit, since these are the only models that have relations to either side and that are very stable compared to e.g. business process models. Finally all cases show, that there is hardly any “best way” to

implement EA but rather a number of successful combinations of factors describing EA implementation. These successful factor combinations, however, are not unchanging, but they have to be continually adapted to the organizations needs and maturity. This on the other hand means, an organization should not necessarily start with an “optimal” EA implementation but with maybe less powerful options like company A which in the beginning has chosen a very passive and therefore not very powerful approach. Now, however, they slowly expanded EA’s active impact as the organization learned to understand the value und associated costs of EA.

## 6 Conclusion and Further Research

Over the previous years we have developed a research framework to observe EA success in companies. When applying this framework to the six cases at hand, we can see some support for the hypothesis that EA success can only partly be assigned to formal EA structures and processes. We assume that—especially with long term success in mind—(a) training and education of architects and non-architects, (b) improving architects’ communication skills, (c) intensifying EA representation in projects, and (d) tool support (not replacements with tools), can significantly contribute to long term EA success. Based on these results the next step in our research is to design a questionnaire and distribute it among EA practitioners on a bigger scale to test these preliminary results empirically.

## References

1. Schönherr, M.: Towards a Common Terminology in the Discipline of Enterprise Architecture. In: Service-Oriented Computing – ICSOC 2008 Workshops, pp. 400–413. Springer, Berlin (2009)
2. Schelp, J., Winter, R.: Language Communities in Enterprise Architecture Research. In: Diversity in Design Science – Proceedings of DESRIST 2009, May 7-9. ACM, Philadelphia (2009)
3. The Open Group: TOGAF (The Open Group Architecture Framework) Version 9 (2009)
4. T. Governance Institute, <http://www.isaca.org>
5. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In: Proceedings of the 10th IEEE EDOC Workshops on Trends in Enterprise Architecture Research (TEAR 2006), Hong Kong, October 17. IEEE Computer Society Press, Los Alamitos (2006)
6. Schelp, J., Aier, S.: SOA and EA – Sustainable Contributions for Increasing Corporate Agility. In: Proceedings of HICSS-42. IEEE Computer Society, Los Alamitos (2009)
7. Magalhaes, R., Zacarias, M., Tribolet, J.: Making Sense of Enterprise Architectures as Tools of Organizational Self-Awareness (OSA). In: Second Workshop on Trends in Enterprise Architecture Research (TEAR 2007), Nova Architectura, St. Gallen, pp. 61–70 (2007)
8. Wegmann, A.: Theory and practice behind the course designing enterprisewide IT systems. IEEE Transactions on Education 47, 490–496 (2004)

9. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog. Software Engineering for Business Information Systems (sebis), TU Munich, Munich (2008)
10. Arbab, F., de Boer, F., Bonsangue, M., Lankhorst, M., Proper, E., van der Torre, L.: Integrating Architectural Models. Symbolic, Semantic and Subjective Models in Enterprise Architecture. In: EMISA, vol. 2, pp. 40–56 (2007)
11. IEEE: IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE 1471-2000). The Institute of Electrical and Electronics Engineers, Inc., New York (2000)
12. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2005)
13. Österle, H.: Business Engineering in the Information Age – Heading for New Processes. Springer, New York (1995)
14. Aier, S., Schönherr, M.: Integrating an Enterprise Architecture using Domain Clustering. In: Second Workshop on Trends in Enterprise Architecture Research (TEAR 2007), Nova Architectura, St. Gallen, pp. 23–30 (2007)
15. Johnson, P., Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making. Studentlitteratur, Pozkal (2007)
16. Henderson, J.C., Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal 32, 4–16 (1993)
17. Wieringa, R.J.: Competenties van de ICT-architect. In: Informatie, SDU 2006, pp. 34–40 (2006)
18. Schelp, J., Winter, R.: Business Application Design and Enterprise Service Design: A Comparison. International Journal of Service Sciences 1, 206–224 (2008)
19. Aier, S., Winter, R.: Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. BISE 51, 150–163 (2009)

# The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement

Marlies van Steenberg<sup>1</sup>, Jurjen Schipper<sup>2</sup>, Rik Bos<sup>2</sup>, and Sjaak Brinkkemper<sup>2</sup>

<sup>1</sup> Sogeti Netherlands, Wildenborch 3, 1112 XB Diemen, The Netherlands

Marlies.van.Steenbergen@sogeti.nl

<sup>2</sup> Department of Information and Computing Sciences,

Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands

{J.Schipper,R.Bos,S.Brinkkemper}@cs.uu.nl

**Abstract.** The field of enterprise architecture is still very much in development. Many architecture teams are looking to improve their effectiveness. One of the instruments to do so is the Dynamic Architecture Maturity Matrix. In the past the DyAMM has been applied to many architecture practices to assess their architecture maturity level. In this paper we present an analysis of these assessments. This provides us with an overview of common strengths and weaknesses in existing architecture practices. In addition, we use the set of assessments to analyze the DyAMM instrument for four types of anomalies.

**Keywords:** enterprise architecture, maturity models, architecture maturity matrix.

## 1 Introduction

Enterprise architecture, the application of principles and models to guide the design and realization of processes, information systems and technological infrastructure, is seen by many as a means to make complexity in IS manageable [1, 2, 3]. For this promise to come true, sound architectural practices, by which we mean the whole of activities, responsibilities and actors involved in the development and application of enterprise architecture, have to be implemented [4, 5]. As an aid in developing these architectural practices architecture maturity models have been introduced in the past. These maturity models are used to assess the maturity of architecture practices and to suggest improvements to these practices. In [5] three types of maturity models are distinguished. The main distinction is between *fixed-level models* that distinguish a fixed number of maturity levels, usually five, like in the well-known CMM [6, 7, 8, 9, 10, 11, 12] and *focus area oriented models* that depart from the idea that there is a fixed number of generic maturity levels and instead define for each focus area its own number of specific maturity levels. To still be able to show incremental growth, the overall maturity of an organization is expressed in terms of combinations of the maturity levels of these focus areas. Focus area oriented models have been applied to testing [13] and software product management [14].

As enterprise architecture is still a field in development, a focus area oriented model is the most appropriate as it allows for a more fine-grained approach [5].

The Dynamic Architecture Maturity Matrix (DyAMM) is such a focus area oriented model. The DyAMM is developed as part of the DyA program in which an approach to enterprise architecture is developed, called Dynamic Architecture (DyA), that focuses on a goal-oriented, evolutionary development of the architectural function [15, 16]. The first version of the DyAMM, version 1.0, was developed in 2002 by a group of experts on the basis of many years of practical experience in the field of enterprise architecture. The format of the DyAMM was taken from the Test Process Improvement (TPI) Model [13]. Based on the first few applications, the DyAMM underwent one update in 2004, which consisted of the repositioning of some of the maturity levels, resulting in DyAMM 1.1. DyAMM 1.1 has been qualitatively validated by applying it to a number of cases [5].

The DyAMM has been applied to many organizations in the last couple of years from many sectors in various countries in Europe as well as in the US. Thus a substantial dataset of assessment results has been accumulated. This dataset not only provides an insight into the state of the architecture practice, but it may also be used to quantitatively analyze and refine the DyAMM as is presented in this paper. We defined four types of instrument anomalies that might occur in focus area oriented models and used the dataset to investigate the extent to which the DyAMM exhibits these four types of potential anomalies. We found only few actual anomalies, thus supporting the validity of the DyAMM instrument.

The contribution of this paper is twofold. On the one hand it provides a view on the state of the architecture practice. On the other hand it provides a way to analyze and fine-tune focus area oriented maturity models. In the next section we will present the DyAMM in more detail. This will be followed in section 3 by an overview of the assessment results collected in the last few years. In section 4 we further analyze the assessment results with the purpose of fine-tuning the DyAMM. In section 5 we provide our conclusions.

## 2 The Dynamic Architecture Maturity Matrix

In this section we briefly discuss the DyAMM. For a full description of the DyAMM we refer to [15].

### 2.1 Structure of the DyAMM

The DyAMM is an instrument to incrementally build an architecture function. It distinguishes 18 architecture practice focus areas that have to be implemented. These focus areas are derived from practical experience in the field of enterprise architecture. The definitions of the architecture practice focus areas are provided in table 1.

Each focus area can be divided into a number of maturity levels. By positioning these maturity levels against each other in a matrix, as shown in figure 1, the DyAMM presents the order in which the different aspects of the architecture function should be implemented. The maturity levels of each focus area are depicted by the letters A to D, indicating increasing levels of maturity. As each focus area has its own specific maturity levels, the number of maturity levels may differ for each focus area, varying from two to four. Most focus areas distinguish three levels. For example the



**Table 1.** The architecture practice focus areas of the DyAMM

Focus area	Definition
Development of architecture	The approach to architecture development, varying from isolated, autonomous projects to an interactive process of continuous facilitation.
Use of architecture	The way architecture is used: merely as a conduit for information, as a means of governing individual projects or even as a tool for managing the entire organization.
Alignment with business	The extent to which the architectural processes and deliverables are in tune with what the business wants and is capable of.
Alignment with the development process	The extent to which architecture is embedded in the existing (business and IT) development processes of the organization.
Alignment with operations	The extent to which architecture is both used in and built on the operations and maintenance discipline.
Relationship to the as-is state	The extent to which the existing situation is taken into account by the architecture processes and deliverables.
Roles and responsibilities	The distribution of responsibilities concerning both architecture processes and deliverables within the organization.
Coordination of developments	The extent to which architecture is used as a steering instrument to coordinate the content of the many developments that usually take place concurrently.
Monitoring	The extent to which and the manner in which compliance of projects with the architecture is guaranteed.
Quality management	The extent to which quality management is applied to the architecture practice.
Maintenance of the architectural process	The extent to which the architectural process is actively maintained and improved.
Maintenance of the architectural deliverables	The extent to which and the manner in which the architectural deliverables are kept up to date.
Commitment and motivation	The extent to which commitment is attained from and shown by the organization.
Architectural roles and training	The acknowledgement and support of the architectural roles and the extent to which architects can educate themselves.
Use of an architectural method	The extent to which a (common) architectural method is used.
Consultation	The extent to which communication among architects and between architects and their stakeholders takes place on a structural basis.
Architectural tools	The extent to which architects are supported by tools.
Budgeting and planning	The extent to which architectural activities are budgeted and planned.

focus area *Use of architecture* has three maturity levels A: *architecture used informatively*, B: *architecture used to steer content* and C: *architecture integrated into the organization*. The position of the letters in the matrix indicates the order in which the focus areas must be implemented to incrementally build an architecture practice in a balanced manner. The thirteen columns define progressive overall maturity scales, scale 0 being the lowest and scale 13 being the highest scale achievable. If an organization has achieved all focus area maturity levels positioned in a column and in all columns to its left, it is at that maturity scale. This is depicted by coloring the cells in the matrix up to and including the maturity level that has been achieved, for each of the focus areas.

The organization depicted in figure 1 for illustrative purposes shows an unbalance in that some focus areas, like *Alignment with the development process*, are quite advanced, while others, like *Use of architecture*, are not yet developed at all. Thus despite the development of some of the focus areas, on the whole the organization in figure 1 is still only at scale 1. Its first step would be to develop *Use of architecture* to maturity level A.

Area	Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Development of architecture			A			B			C						
Use of architecture				A			B				C				
Alignment with business			A				B				C				
Alignment with the development process				A				B		C					
Alignment with operations						A			B			C			
Relation to the as-is state						A				B					
Roles and responsibilities					A		B					C			
Coordination of developments								A			B				
Monitoring					A		B		C		D				
Quality management									A		B			C	
Maintenance of the architectural process								A		B		C			
Maintenance of architectural deliverables						A			B					C	
Commitment and motivation			A					B		C					
Architecture roles and training					A		B			C				D	
Use of an architectural method					A						B				C
Consultation				A		B				C					
Architectural tools								A				B			C
Budgeting and planning					A							B		C	

Fig. 1. The Dynamic Architecture Maturity Matrix

An organization that still has to build its architecture practice, starts with developing the focus areas positioned in scale 1 to their first maturity levels: *Development of architecture*, *Alignment with business* and *Commitment and motivation* (the A's in column 1). To get to the next stage, the first maturity levels of the focus areas *Use of architecture*, *Alignment with the development process* and *Consultation* have to be achieved (the A's in column 2). And so on. Once all A's in columns 1 to 3 have been achieved, is it time to develop the focus area *Development of architecture* to the next level (the B in column 4). In this way the matrix can be used to set priorities in developing the architectural practice.

### 2.2 Use of the DyAMM

Each maturity level of each focus area is associated with one to four yes/no questions. Focus area level determination is done by answering these questions. Only if all questions associated with a maturity level can be answered confirmatively for an organization, the associated maturity level can be said to be achieved. Table 2 shows as an example the questions associated with level A of the focus area *Use of architecture*. In total there are 137 questions associated with the matrix.

Table 2. Questions to measure maturity level A of focus area Use of architecture

Nr.	Question
9	Is there an architecture that management recognizes as such?
10	Does the architecture give a clear indication of what the organization wants?
11	Is the architecture accessible to all employees?

The DyAMM can be applied in two distinct ways: as an *independent assessment* or as a *self assessment*. The primary use of the DyAMM is as an assessment instrument to be used by independent assessors. Usually, an assessment is commissioned by the person responsible for the architectural function, most often the head of the architects, the head of the IT department or the CIO. The assessment may be the first step in an

improvement process. The assessors, usually as a team of two, complete the matrix by answering all 137 questions. They base their answers primarily on interviews with relevant stakeholders, like senior managers, business managers, project managers, system developers, operations personnel and architects. In addition, documentation is studied to support the findings from the interviews and to establish width and depth of the architectural artifacts. The second use of the DyAMM is as a self assessment to be completed by individuals for their own organization. Architects can answer the 137 questions for themselves, which leads to a completed matrix. This latter use of the DyAMM is offered as a service to the architectural field [17].

### 3 Analysis of DyAMM Assessments

We collected 56 assessments conducted in the context of the DyA program over the period 2005 – 2008. This set includes independent assessments performed by DyA experts as well as self assessments executed in the context of courses or workshops. The assessments were collected directly from the DyA experts involved, which enabled us to establish their origins. In some assessments the authors were involved. The assessments are distributed over various industry sectors. Financial intermediation (23.2%), public administration (21.4%), transport, storage and communications (16.0%) and manufacturing (14.3%) are best represented. The high representation of financial intermediation in our set corresponds with a general high enterprise architecture awareness that can be noticed in this sector and that is evidenced for instance by the frequent representation of this sector at enterprise architecture conferences. The high percentage of public administration is in line with an increasing demand by government to apply enterprise architecture practices.

#### 3.1 Overall Maturity Distribution

After establishing the dataset we conducted a number of analyses. First we determined the distribution of the 56 cases over the 13 maturity scales (table 3). We looked both at the minimum scale and at the average scale. The minimum scale is the scale for which an organization has achieved all focus area maturity levels positioned in that column and in all columns to its left. The average scale was calculated for each case by adding the scales for each focus area and dividing this by 18, e.g. for the case in figure 1 this is 1 for *Development of architecture*, 0 for *Use of architecture*, etc.

**Table 3.** Overall maturity distribution

		Minimum		Average		
Scale	Frequency	Percentage	Scale	Frequency	Percentage	
0	50	89.3	0	8	14.3	
1	4	7.1	1	18	32.1	
2	2	3.6	2	16	28.6	
≥3	0	0.0	3	7	12.5	
<b>Total</b>	<b>56</b>	<b>100</b>	4	5	8.9	
			5	1	1.8	
			6	0	0.0	
			7	1	1.8	
			≥8	0	0.0	
			<b>Total</b>	<b>56</b>	<b>100</b>	

The distribution shows that the vast majority of organizations is still at scale 0 and that none has a scale of 3 or higher. Looking at the average scores, we see more spread. This indicates that various organizations score high on some focus areas, while at least one of the focus areas needed for the first few scales is underdeveloped. This is indicative of an unbalance in the development of the 18 focus areas, some focus areas being well developed, while some essential areas seem to be neglected.

### 3.2 Distribution of Focus Area Maturity Levels

Drilling down, we next investigated the maturity level distribution of the 56 cases on each of the 18 focus areas of the DyAMM. This is presented in table 4.

**Table 4.** Distribution of organizations over focus area maturity levels

Focus area	0	A	B	C	D	Total
Development of architecture	60.7	26.8	3.6	8.9	-	100
Use of architecture	82.2	7.1	10.7	0	-	100
Alignment with business	75	10.7	8.9	5.4	-	100
Alignment with the development process	23.2	41.0	32.2	3.6	-	100
Alignment with operations	66.1	19.6	12.5	1.8	-	100
Relationship to the as-is state	66.1	21.4	12.5	-	-	100
Roles and responsibilities	42.8	5.4	46.4	5.4	-	100
Coordination of developments	51.8	30.4	17.8	-	-	100
Monitoring	89.2	1.8	5.4	1.8	1.8	100
Quality management	92.8	5.4	1.8	0	-	100
Maintenance of the architectural process	76.8	14.3	7.1	1.8	-	100
Maintenance of the architectural deliverables	73.2	21.4	1.8	3.6	-	100
Commitment and motivation	34.0	57.1	7.1	1.8	-	100
Architectural roles and training	10.7	34.0	46.4	7.1	1.8	100
Use of an architectural method	67.8	30.4	1.8	0	-	100
Consultation	42.9	48.2	7.1	1.8	-	100
Architectural tools	48.2	37.5	12.5	1.8	-	100
Budgeting and planning	53.6	41.0	5.4	0	-	100

The focus area maturity level distribution in table 4 shows for each of the maturity levels of each focus area what percentage of the organizations in the dataset score that particular level. Thus, for the focus area *Use of architecture* it is shown that 82.2% of the organizations score level 0, 7.1% scores level A, 10.7% scores level B, and 0% scores level C. It is clear from the focus area maturity level distribution that level 0 has a relatively high score on most of the focus areas. This implies that on many aspects the maturity of the architecture practices assessed is rather low. A few focus areas have a better distribution of organizations over the maturity levels, like *Alignment with development* and *Architectural tools*. Apparently there is a difference between organizations in maturity for these aspects of the architecture practice. This may indicate that some organizations pay more attention to them, either because they deem them more important or because they are relatively easy to achieve.

Translating the focus area maturity level distribution into the average maturity score for each focus area enables us to rank the focus areas according to maturity level found in the dataset (figure 2). The average maturity score is calculated by attaching a score of 0 to 4 to the maturity levels 0, A, B, C and D respectively. The fact that not all focus areas distinguish exactly five levels is corrected for in calculating



**Fig. 2.** The average score for each of the focus areas

the average score, by translating all focus areas to a number of four levels. This gives a range of potential scores between 0 and 3.

The average maturity scores show that there is a clear difference in maturity between the focus area scoring lowest, *Quality management* and the focus area scoring highest, *Architectural roles and training*. If we relate the average score to the positioning of the focus areas in the DyAMM we see that the following focus areas that, according to the DyAMM, should be addressed at an early stage (scales 1 and 2), score relatively low on actual maturity: *Development of architecture* (scale 1), *Alignment with business* (scale 1) and *Use of architecture* (scale 2).

If we look at the focus areas of scale 3, we see the following focus areas scoring low: *Monitoring*, *Use of an architectural method* and *Budgeting and planning*. Of these, *Monitoring* scores especially low. *Monitoring* seems to be a common weakness in the architecture practices analyzed. Maturity level A of the focus area *Monitoring* is positioned at scale 3 in the matrix. The low score on this focus area apparently prevents organizations from attaining scale 3. This explains the fact that none of the assessed organizations score scale 3 or higher.

Looking at the top 3 focus areas in figure 2, we see that the following focus areas score highest: *Architectural roles and training* (scale 3), *Alignment with the development process* (scale 2) and *Roles and responsibilities* (scale 3). Of all three, maturity level A is positioned within the first three scales, i.e. relatively early in the maturity development process. Thus, the attention to these focus areas is justified. The focus areas *Architectural roles and training* and *Roles and responsibilities* are both concerned with structural aspects of the architecture practice. It seems that it is common to have these structural aspects in place. The high score of *Alignment with the development*

*process* is striking, especially when compared with the much lower scores of *Alignment with operations* and *Alignment with business*. It seems that most organizations are more mature in the relation between architecture and projects, than in the relation between architecture and operations or business. The immaturity of the alignment with the business may be a consequence of the fact that architectural thinking most often originates in the IT department. This does not explain, however, the low score on alignment with operations.

On the whole we may conclude that the architecture practices assessed are still in the early stages of architecture maturity. Architecture as an instrument for providing guidance to projects is relatively well developed, though follow up in the sense of compliance monitoring is strikingly lacking. The alignment of the architectural choices with the business goals and the existence of an interactive dialogue between architects and business is still underdeveloped.

## 4 Analysis and Refinement of the DyAMM Instrument

The dataset not only gives us a picture of the EA practice, it can also be used to analyze and fine-tune the DyAMM, by using it to detect anomalies that might point to flaws in the DyAMM.

### 4.1 Approach

To fine-tune the DyAMM we defined four kinds of potential instrument anomalies, which we then searched the dataset for. The anomalies found we further explored in an expert panel, after which we decided on whether the anomaly should lead to an adaptation of the DyAMM. Thus our approach consisted of four steps: (1) Definition of instrument anomalies, (2) Quantitative analysis of the dataset, (3) Discussion in Expert Panel, (4) Decision making.

The first step was to define instrument anomalies that might distort the result of an assessment. What we looked for were elements that did not fit the concept of incremental growth, elements that were superfluous and elements that showed interdependency. With this in mind, we identified four kinds of potential instrument anomalies: blocking questions, blocking levels, undifferentiating questions and correlations (table 5).

**Table 5.** Potential instrument anomalies

Anomaly	Definition
Blocking question	A question that in at least 10% of the cases was answered with “No”, while if it were answered with “Yes” by these organizations, they would move up at least two levels for the focus area concerned.
Blocking level	A focus area maturity level that is not achieved by at least 10% of the cases, while if these organizations would achieve the level, their overall score would be 2 scales higher.
Undifferentiating question	A question that at least 85% of the assessments answered with “Yes” and that thus does not differentiate between organizations.
Correlation	A dependency between two focus areas with a significance of 0.05.

Blocking questions may indicate that a question or level should be moved to a higher maturity level. Blocking levels may indicate that the level concerned should be moved to the right in the matrix. Undifferentiating questions seem to be superfluous and might possibly be removed, making the use of the matrix more efficient. Correlations indicate a dependency between two focus areas. This may indicate that focus areas should be combined.

Quantitative analysis of the dataset provided a few anomalies that required further investigation. These anomalies were discussed in an expert panel consisting of seven experts with 4 to 17 years of experience in the field of enterprise architecture, from organizations differing in size from 600 to 65.000 employees. The expert session consisted of three parts: (1) discussion of the blocking questions found, (2) rating the importance of the focus areas and (3) discussing the correlations found. In the session we checked whether the blocking questions were understandable, whether they were relevant to the architecture practice and, if so, at what level of maturity they are most relevant. We also asked the participants to rate the importance of each of the focus areas for an architecture practice in the starting-up phase. Finally, we asked whether they had any explanations for the correlations found. The expert session was organized in a group decision room at Utrecht University [18, 19].

Based on the discussion in the expert panel, we made a final decision on how to deal with the anomalies found.

## 4.2 Blocking Questions

**Quantitative analysis.** Quantitative analysis provided three blocking questions (table 6). There are three possible responses to a blocking question: (1) the question should be moved to a higher level, (2) the question should be rephrased or (3) the question represents a genuine weakness in today's architecture practices and should remain as it is.

**Table 6.** Blocking questions

Nr.	Question	Focus area	Percentage blocked
18	Is there a clear relationship between the architecture and the organization's business goals?	Alignment with business	10.7
44	Has a policy been formulated concerning the as-is state (existing processes, organizational structures, information, applications and technical infrastructure)?	Relationship to the as-is state	12.5
48	Does the architecture have an official status in the organization?	Roles and responsibilities	14.3

**Expert panel.** To determine our response to the blocking questions we used the opinion of the experts on the blocking questions, as presented in table 7. To determine whether questions should be rephrased we asked for the understandability and relevance of the questions. To determine whether the questions should be moved to a higher level, we asked for the maturity phase in which the questions become relevant.

**Table 7.** Expert opinion on blocking questions

Nr	Question	Understandable?	Relevant?	Phase?
18	Is there a clear relationship between the architecture and the organization's business goals?	Yes (6)	Yes (7)	1 (4)
		No (1)	No (0)	2 (3)
				3 (0)
44	Has a policy been formulated concerning the as-is state (existing processes, organizational structures, information, applications and technical infrastructure)?	Yes (7)	Yes (7)	1 (5)
		No (0)	No (0)	2 (2)
				3 (0)
48	Does the architecture have an official status in the organization?	Yes (7)	Yes (6)	1 (3)
		No (0)	No (1)	2 (3)
				3 (1)

The numbers in brackets in table 7 show the number of respondents giving the answer indicated. The rightmost column shows the maturity phase in which according to the experts the question mentioned becomes relevant. Phase 1 translates to scale 0-3 in the matrix, phase 2 to scale 4-8 and phase 3 to scale 9-13. Table 7 shows that the questions are well-understood and relevant. This indicates that the questions need not be removed or rephrased in order to be understood. Regarding the position of the question, there is more deviation between the experts. Most consensus is about question 44. Most participants agree that the question is relevant in the early stages of an architecture practice. For the other two questions about half of the participants place them in the early stages, and the other half place them in the middle stage, when architecture is more or less on its way. In the matrix this would mean between scale 4 and 8 which would indicate a move of the questions to a next level. The difference in opinion regarding question 48 concentrated on the fact that in some organizations a formal approval at an early stage is important, while in others it is not. This seems organization dependent. Question 18 is one of the two questions that are associated with level A of the focus area *Alignment with business*. Interestingly, the other question, question 19, 'Is the architecture evaluated in terms of the business goals?', serves as a blocking question for 7% of the assessments. This suggests two possibilities: (1) level A might be positioned too early in the matrix and should be moved to a higher scale or (2) the questions are not appropriate to level A and must be reconsidered. As the experts rank *Alignment with business* as the second most important focus area in the early stages, we may conclude that the positioning of level A of *Alignment with business* at scale 1 needs no reconsideration.

**Decision making.** Question 44 is well-understood and deemed relevant to the early stages of an architecture practice, thus we leave this question as it is (response option 3). Question 48 is well-understood and deemed relevant to the early stages of an architecture practice for formally oriented organizations. As the present version of the DyAMM is generic for all organizations, we opt to leave this question too as it is (option 3). Question 18 is well-understood, but the discussion indicates that it might be too advanced for the early stages. This goes for its companion question 19 as well. Which leaves us to consider whether other questions would be more appropriate to this level. We put reconsideration of questions 18 and 19 as an item on the improvement list for version 1.2 of the DyAMM to be further investigated (option 2).



### 4.3 Blocking Levels

**Quantitative analysis.** Quantitative analysis provided three cases that contained a level preventing them from moving up two scales. Two organizations scored 0 on the focus area *Development of architecture*, preventing them from moving from scale 0 to scale 2. One organization scored 0 on the focus area *Alignment with business*, preventing it also to move up from scale 0 to scale 2. These numbers are too small to consider these levels blocking levels, as the threshold of 10% is not reached by far. Thus we may conclude that the DyAMM does not contain any blocking levels.

### 4.4 Undifferentiating Questions

**Quantitative analysis.** Two undifferentiating questions were found.

**Table 8.** Undifferentiating questions

Nr.	Question	Focus area	Percentage Yes
95	Are money and time allocated to architecture?	Commitment and motivation - A	87.5
102	Does the role of architect exist in the organization?	Architectural roles and training - A	87.5

**Decision making.** The fact that these two questions have such a high percentage of Yes score, can be partly explained by the fact that not many organizations will perform an architecture assessment if they do not allocate any time or money to architecture, or in some way or other recognize the role of architect. It might be worthwhile, however, to reconsider these two questions. As question 2 is the only question for level A of *Architectural roles and training*, this also explains the high percentage of level A for this focus area. We decided to put the reconsideration of questions 95 and 102 on the improvement list for DyAMM 1.2.

### 4.5 Correlations

**Quantitative analysis.** Analyzing all relationships between the 18 items, we found one (Pearson, two-sided) correlation that complies with a significance level of 5%: between *Alignment with the development process* and *Commitment and motivation* ( $r=.37$ ;  $p=.04$ ), and two correlations that meet a significance level of 1%: between *Commitment and motivation* and *Architectural roles and training* ( $r=.55$ ;  $p=.00$ ) and between *Architectural roles and training* and *Alignment with the development process* ( $r=.43$ ;  $p=.01$ ).

**Expert panel.** We presented the three significant correlations found to the experts, asking them whether they could explain the correlations. This generated some suggestions, like the idea that when management shows commitment to architecture, projects will be more inclined to take architecture into account and more attention will be paid to the function of architect. Or, when architects are better trained, they will generate commitment more easily. An explanation offered for the third correlation was that as functions and roles are important in many development processes, alignment is easier

when the function of architect is formally established. Not one explanation emerged as being the most likely, however.

**Decision making.** Further research is needed to investigate why these correlations were found, and no significant correlation between other items. The potential application of factor analysis or scale analysis to explore this, however, is not feasible due to the number of observations (N=56).

#### 4.5 Overall Conclusion

Taking all into account, not many instrument anomalies were found (table 9).

**Table 9.** Instrument anomalies found with the dataset

Type of anomaly	Anomalies found	Action
Blocking question	Questions 18, 44, 48	Reconsider questions 18, 19
Blocking level	None	None
Undifferentiating question	Questions 95, 102	Reconsider questions 95, 102
Correlation	Three significant correlating focus areas	Further investigate explanation of significant correlations

Though the analyses lead to the reconsideration of a few questions, they do not lead to a shifting of maturity levels within the matrix. The correlations found do not lead to immediate changes in the DyAMM, but they do ask for further investigation on the interdependencies between various aspects of the architecture practice.

## 5 Conclusions and Further Research

This paper presents the analysis of 56 assessments of architecture practices executed with the DyAMM. From this analysis we get, on the one hand, a view of the current strengths and weaknesses of architecture practices and, on the other hand, a quantitative validation of the assessment instrument used, the DyAMM, a focus area oriented maturity model. For the latter purpose, we defined four types of potential instrument anomalies that might occur in focus area oriented maturity models.

As for the current status of the architecture practice, we may conclude that most practices are still in the early stages of maturity, and that on the whole there is much room for improvement. The aspects most underdeveloped but of great importance to young architecture practices, are the alignment of the architecture with the business goals, the use of architecture as a guiding principle to design and the monitoring of projects on compliance with the architecture. These are aspects architects should definitely work on. Better developed are the structural aspects of architectural function descriptions and education and the assignment of architectural responsibility. Also, the alignment with projects is well developed. Commitment and motivation are reasonably well realized within the organizations assessed.

As far as the DyAMM itself is concerned, the analysis does not give rise to fundamental changes in the matrix, though it provides some input to fine-tune the matrix. This fine-tuning consists primarily of reformulating some of the questions. The fact that so few anomalies were found is additional support for the validity of the

DyAMM and quantitatively strengthens previous qualitative validation. We intend to use the results of our analysis in formulating version 1.2 of the DyAMM. We will also retain the dataset to test future suggestions for improvement from other assessors.

The expert panel discussion indicates that some of the questions of the DyAMM are organization-dependent, like the importance of the architecture being formally approved. It might be interesting to investigate whether there might be cause for different matrices for different types of organizations.

The correlations found need further investigation into interdependencies between various aspects of the architecture practice. Of course, this may in turn lead to further refinement of the DyAMM in future.

An interesting area for future research is the cause of the discrepancy in maturity between alignment with the development process and the consequent monitoring of this same development process.

There are some limitations to the research presented here. In our dataset we combined assessments that were conducted in various ways, independent assessments as well as self assessments being included, to increase the number of assessments contained in the set. Preliminary analysis of the assessment results, comparing the average focus area scores, gives no indication that these two types of assessments differ substantially, though. A second limitation is that all cases score in the first three scales. This means that we cannot draw definite conclusions yet on the validity of the higher scales in the DyAMM. As the execution of assessments continues, we hope to repeat the analyses in the future with a larger set.

**Acknowledgments.** The authors wish to thank the participants of the expert panel for their contribution in discussing the anomalies found. Thanks also to Ronald Batenburg, Nico Brand, Wiel Bruls, Ralph Foorhuis and Ilja Heitlager for their valuable comments on an earlier draft of this paper.

## References

1. Lankhorst, et al.: Enterprise Architecture at Work. Springer, Heidelberg (2005)
2. Ross, J.W., Weill, P., Robertson, D.: Enterprise Architecture as Strategy. Harvard Business School Press, Boston (2006)
3. Versteeg, G., Bouwman, H.: Business architecture: a new paradigm to relate business strategy to ICT. *Information Systems Frontier* 8, 91–102 (2006)
4. Van der Raadt, B., Slot, R., Van Vliet, H.: Experience report: assessing a global financial services company on its enterprise architecture effectiveness using NAOMI. In: Proceedings of the 40<sup>th</sup> Annual Hawaii International Conference on System Sciences (2007)
5. Van Steenberg, M., Van den Berg, M., Brinkkemper, S.: A Balanced Approach to Developing the Enterprise Architecture Practice. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) *Enterprise Information Systems. LNBIP*, vol. 12, pp. 240–253 (2007)
6. GAO: A framework for assessing and improving enterprise architecture management (2003)
7. CMMI: CMMI<sup>SM</sup> for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1) Staged Representation; CMU/SEI-2002-TR-012; ESC-TR-2002-012 (2002)

8. Appel, W.: Architecture Capability Assessment. *Enterprise Planning & Architecture Strategies*. METAGroup 4(7) (2000)
9. METAGroup: Diagnostic for Enterprise Architecture, META Practice (2001)
10. NASCIO: NASCIO enterprise architecture maturity model (2003)
11. Westbrook, T.: Architecture Process Maturity Revisited and Revised. METAGroup Delta 2902 (2004)
12. Luftman, J.: Assessing business-IT alignment maturity. *Communications of AIS* 4, Article 14, 1–49 (2000)
13. Koomen, T., Pol, M.: *Test Process Improvement, a practical step-by-step guide to structured testing*. Addison-Wesley, Boston (1999)
14. Van de Weerd, I., Bekkers, W., Brinkkemper, S.: *Developing a Maturity Matrix for Software Product Management*. Institute of Computing and Information Sciences, Utrecht University. Technical report UU-CS-2009-015 (2009)
15. Van den Berg, M., Van Steenberghe, M.: *Building an Enterprise Architecture Practice*. Springer, Dordrecht (2006)
16. Wagter, R., Van den Berg, M., Luijpers, L., Van Steenberghe, M.: *Dynamic Enterprise Architecture: how to make it work*. Wiley, Hoboken (2001)
17. Information site DYA: <http://www.dya.info>
18. DeSanctis, G., Gallupe, R.B.: A Foundation for the Study of Group Decision Support Systems. *Management Science* 33(5), 589–609 (1987)
19. GroupSystems MeetingRoom: *Concepts Guide*, Tucson, Arizona (1990 – 2001)

# Decoupling Models and Visualisations for Practical EA Tooling

Steffen Kruse, Jan Stefan Addicks, Matthias Postina, and Ulrike Steffens

Software Engineering for Business Information Systems  
OFFIS - Institute for Information Technology  
Escherweg 2  
26121 Oldenburg, Germany  
{[steffen.kruse](mailto:steffen.kruse@offis.de), [jan.stefan.addicks](mailto:jan.stefan.addicks@offis.de),  
[matthias.postina](mailto:matthias.postina@offis.de), [ulrike.steffens](mailto:ulrike.steffens@offis.de)}@offis.de  
<http://www.st.offis.de>

**Abstract.** Rigorous modelling techniques and specialised analysis methods support enterprise architects when embarking on enterprise architecture management (EAM). Yet, while customised modelling solutions provide scalability, adaptability and flexibility they are often in conflict with generic or reusable visualisations. We present an approach to augment customised modelling with the techniques of model transformations and higher-order transformations to provide flexible and adaptable visualisations with a minimum of requirements for the underlying enterprise models. We detail our approach with a proof-of-concept implementation and show how a decoupling can ease EAM approaches and provide appropriate tooling in practice.

## 1 Motivation

An adequate visualisation of interrelations is one important foundation of enterprise architecture management (EAM) [1] and often considered particularly useful for decision support by different stakeholders. However, there is still a number of obstacles to be overcome in order to support the flexible and effective creation of such visualisations.

EAM enterprise models may result from EA frameworks like described in [2,3] or more general in [4]. These frameworks provide initial samples of enterprise models which can be refined for specific enterprises or organisations. Changing markets, legal requirements, or changes in the organisational structure force enterprises to adapt accordingly. These changing business needs again have to be reflected by the underlying IT structure [5] and the respective enterprise model. The same is true for the managed evolution of enterprises' IT architectures as described e.g. in [6] which leads to new stakeholder concerns [7] and in turn call for an adjustment of enterprise models. Hence, enterprise models can be expected to be unique for each enterprise (at least at a certain degree of detail) and to mature over time. Following this line of argument, we conform to [8,9] who call for flexible and extensible enterprise models.

The required flexibility of enterprise models has to be expanded to the types of analyses performed on them. Although common EA concerns can be identified [10], they still vary depending on an enterprise's specific characteristics and stage of development. Furthermore, EA research develops new concepts for further exploration of EA information for new possibilities of analyses to suit diversifying kinds of stakeholders [11,12] and more generic techniques for executing these kinds of analyses have been proposed [13,14,15].

Combining flexible enterprise models with flexible analyses is a challenging task, since each type of analysis has to rely on some basic assumptions with regard to the enterprise model it is used on. However, for EA visualisations we can mitigate this problem by decoupling models from visualisations and by enabling the design of new visualisations on detached models. This is achieved by the use of model transformations and higher-order transformations (HOTs). These techniques allow a flexible way of visualising EA interrelations and provides adequate visualisations for different stakeholders and different concerns.

The remainder of the paper is structured as follows: In the next section, we give an overview of our approach, detailing the separate concerns in employing flexible visualisations. Section 3 covers the technical detail of prototyping our approach; starting with transforming models at run-time to produce visualisations from arbitrary enterprise models and going on to the use of HOTs for the configuration of visualisation cases. We conclude with a description of the next challenges at hand. <sup>1</sup>

## 2 Overview of Our Approach

EAM visualisations can be structured according to views and viewpoints in the software engineering technology space [16]. In this regard, a visualisation covers one or more stakeholder *concerns* by selecting suitable enterprise model elements and bundling this selection as a *viewpoint*. When the visualisation is displayed, being populated by live data (as instances of the selected enterprise model entities) a *view* conforming to the defined viewpoint is created.

Our approach can be structured according to four areas of interest, each covering different aspects of creating and utilising EAM visualisations. This separation of concerns is made possible by the decoupled use of models and viewpoints and potentially allows different experts and stakeholders to participate in the process. The areas are:

*Enterprise Modelling.* An enterprise model identifies domain specific and enterprise specific information objects and processes which are relevant for the EA activities of the different stakeholders. We build our prototype around the Eclipse Modeling Framework Project (EMF) <sup>2</sup>. We expect enterprise models to be unique for each enterprise to cater for individual characteristics. None the

---

<sup>1</sup> The research described in this paper was partially accomplished in the IF-ModE project funded by the German Federal Ministry of Education and Research.

<sup>2</sup> <http://www.eclipse.org/modeling/emf/>

less, modelling does not have to be started "from scratch" as enterprise models can be derived from domain reference models (e.g. [17], [18]) and by employing accepted EA frameworks like TOGAF [2] or the Zachman Framework [3].

We use an excerpt from an exemplary enterprise model to illustrate our approach (see figure 1). In essence it covers the relevant organisational units, processes, and software components and how these interact in a given enterprise. For this purpose, the *Support*-entity defines which component supports which organisational unit in which process. To order processes and organisational units, these can be organised in trees: sub-entities reference their parent entities via the respective *parent*-relation. When the corresponding information has been collected (i.e. an instance of this model is available), questions like "Which part of my organisation needs software X for which purpose?" can be answered. These answers are best given as suitable EA visualisations.

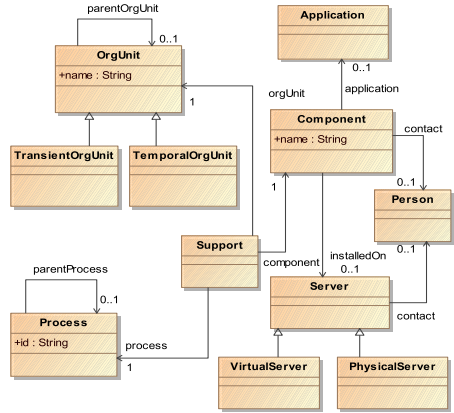
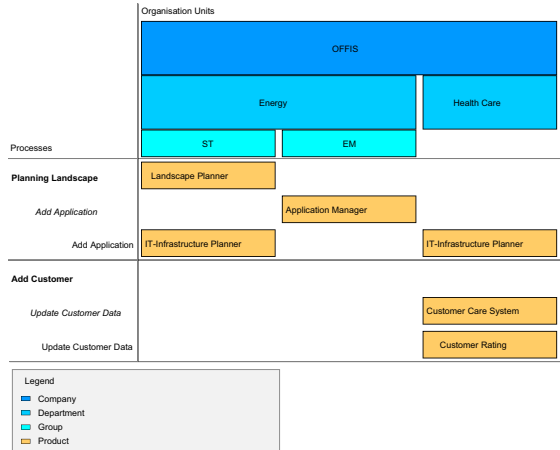


Fig. 1. Enterprise Model Example

*EA Visualisation Design.* For different kinds of information (and questions), appropriate forms of visualisation are needed. In general, the suitability of one type of visualisation over another does not only depend on the informational content but rather on the way the information is structured and how data entities relate to each other. A trivial example is the use of tables to display large amounts of sets of textual information with reoccurring attributes, while graph-like structures with textual annotations and directed edges are an accepted means to visualise processes. In essence, the design of the visualisation is independent of domain or enterprise context and only depends on the characteristics of the visual data. Our approach provides the means to specify visualisations in a domain-independent manner and thereby to allow visualisations to be bundled and reused in different contexts. In this way, visualisations can either be designed up-front by tool vendors and then applied to individual enterprise models or developed during an actual EA project and later reused.

Here we chose the matrix-map as an example of our work (see figure 2). The matrix-map is a suitable visualisation when looking at how one set of entities is distributed by relation over two other entity sets. The related entities make up the x- and y-axes and can be organised in sets of trees, while the entities of interest fill the matrix between the axes. Our example shows how software components of a fictional enterprise are distributed between processes and organisational units. This relationship is defined by the *Support*-entity in our exemplary enterprise model (see figure 1).

*EA Visualisation Configuration.* Once EA has been introduced within the enterprise and both, the enterprise model and the domain-independent visualisations are available; our approach provides the means to bring these two parts together. For this purpose, an enterprise architect selects a suitable type of visualisation and the model elements to be visualised. These are then mapped to configure a viewpoint of the required information to answer a specific question at hand. It is up to the enterprise architect to select the appropriate visualisation and informational content for the different stakeholder concerns.



**Fig. 2.** Matrix-map

*EA-Information Procurement and Evaluation.* Once viewpoints are configured, they can be filled with live data from a repository and updated at will to produce views. Stakeholders can use the resulting views (visualisations) for their informational needs. Depending on the kind of visualisation, further configuration mechanisms may be available at usage-time, such as filters or individual highlighting of information.

These four areas of interest allow us to employ our approach within different EA scenarios. For example, tool vendors can realise visualisation design as part of their tool implementation. An enterprise relying on the respective tool can then specify its individual enterprise model and configure the viewpoints to provide its stakeholders with views. Alternatively, an enterprise pursuing its own EA implementation can realise all four concerns by itself. This flexibility better enables enterprises to select the best strategy in fulfilling their EA concerns.

### 3 Overview of the Generation Process

As Buckl et al. [19] have shown, it is possible to use model to model transformations to extract the information needed for a specific viewpoint from an enterprise model and produce a model suited for processing by an algorithm to create the layout of an EA visualisation. This is quite obvious when models conformant to metamodels exist or can be created for both sides and can be related by a transformation. Yet, we deem the specification of model transformations to be a difficult task for people with little expertise in the area and inexpedient in practice, as a new transformation is needed for each viewpoint. We show how the transformations can be generated by higher order transformations (HOTs).



This eliminates the need for knowledge of model transformations for end-users and enables a high potential for the re-use of visualisations.

In essence, our approach relies on separating the semantics and structure of the information to be displayed from how it is displayed. This is achieved by defining the minimal set of characteristics<sup>3</sup> which a set of information objects is required to exhibit to be displayable as a given type of visualisation. For this purpose we create visualisation-specific models (ViMs) for each visualisation type (such as the matrix-map).

### 3.1 The Visualisation-specific Model (ViM)

The central entity of a visualisation is a visualisation-specific model (ViM). It defines the model elements and structure needed to configure a concrete viewpoint. Views (instances of this model) are to be processed by a layout engine to produce the final visualisation. The ViM describes valid models of a visual domain; it is independent of the semantics and architectural domain of the information to be visualised (enterprise model). In the case of the matrix-map (see the left-hand side of figure 4), the key entities are first the nodes making up the x- and y-axes, second the entities displayed in the matrix, and third the relationships between these. We refer to the entities shown on the matrix as *Items*. These have a title and a set of *Attachments*, which can provide further information on the properties *Items* possess. Each *Item* relates to exactly one *Node* on the x- and y-axes. The *Nodes* are organised in trees along the axes. We generated a Java object-model for the ViM using tooling of the Eclipse Modeling Framework Project (EMF) and developed a layouter to create a resulting diagram from the ViM as shown in figure 2.

### 3.2 The Transformation Process

To generate views at run-time, we make use of model transformations [20]. The model transformations describe the rules of producing views from a set of enterprise data in terms of the respective viewpoints. (Instances of the enterprise model on the left-hand side (LHS) are transformed into instances of the ViM on the right-hand side (RHS).) One transformation description covers exactly one viewpoint; it maps entities of the enterprise domain to those of the visual domain (see figure 3).

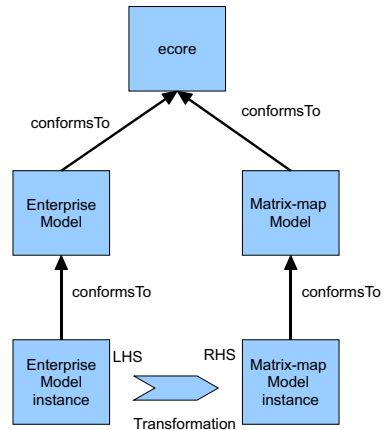


Fig. 3. Runtime Transformation

<sup>3</sup> We refrain from using the term "pattern" at this stage, as characteristics which are difficult to express in terms of patterns (like data dispersion or order) may become important in special types of visualisations. During the course of our future work on visualisation types we hope to sharpen the term.

In our prototype, we use QVT Relations (QVT-RL) [21] to describe transformations and the MediniQVT engine<sup>4</sup> for the execution. The enterprise meta-model is EMF.ecore based. EMF.ecore enjoys far reaching use in modelling in general and an appropriate model can be derived by transformation from other formats. We use Teneo of the Eclipse Modeling Framework Technology (EMFT) project<sup>5</sup> to produce an enterprise model from an EAM repository at runtime. Teneo provides automated persistency of EMF.ecore based models. Since Teneo in turn utilises Hibernate<sup>6</sup>, mappings to a wide variety of databases and data schemas are possible. Teneo has the added benefit of handling model queries transparently and loading entities only as they are required by the transformation process, which keeps data access to a minimum. In our example, the transformation maps *OrgUnits*, *Processes* and *Components* to *MatrixNodes* and *Items* and fills the corresponding titles. As each transformation covers one viewpoint, it means that a new transformation is needed when a different combination of entities is to be visualised (a different viewpoint is needed). For example, if we wanted to show in a matrix map who (*Person*) is responsible for which *Component* running on which *Server*, we would have to map these entities to the matrix map ViM again using a transformation description although the type of visualisation remains the same. Here we see a major drawback in relying solely on transformation descriptions for this task, as the user has to be adept at programming transformations. For this reason our approach generates transformations for viewpoints using HOTS.

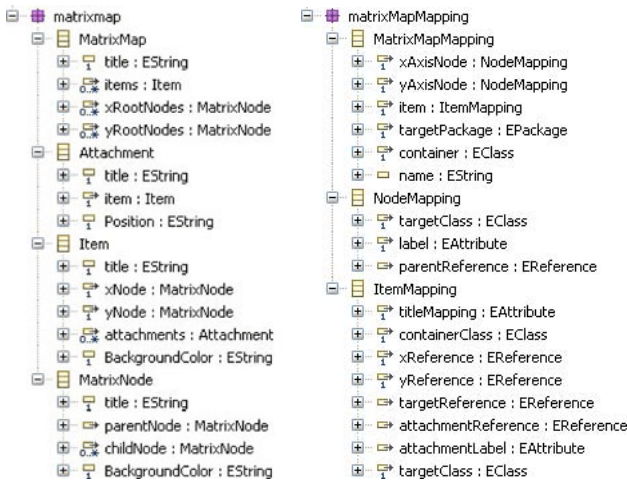


Fig. 4. Matrix-map and mapping model

<sup>4</sup> <http://www.ikv.de/>

<sup>5</sup> <http://www.eclipse.org/modeling/emft/>

<sup>6</sup> <http://www.hibernate.org/>

### 3.3 Generating Transformations: Higher-Order Transformations

To configure a viewpoint, we use a mapping model (MM), which maps elements of the enterprise model to elements of the ViM (see the right-hand side of figure 4). The MM is specific to the given visualisation (with constructs like x- and y-axis and nodes for the matrix-map), but specifies the required elements from the enterprise model in terms of ecore types. Thereby any enterprise model based on ecore (or any model based on ecore to be exact) can potentially serve as input and can be visualised using the given approach - without adaptation. The mapping models for the different visualisation types are augmented with a set of constraints detailing further characteristics the selected set of enterprise classes must fulfil to qualify for a viewpoint. As a simple example, consider that the *EAttribute* in a *NodeMapping* has to belong to the given *targetClass* *EClass* in order for the mapping to be correct (figure 4).

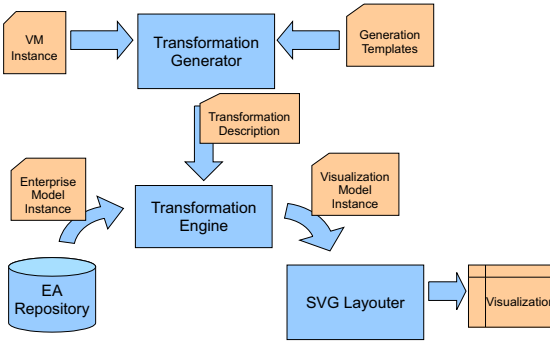


Fig. 5. Overview of the Component Interaction

defined in terms of the mapping model, which in turn makes no assumptions on a specific enterprise model, beyond the minimal set of characteristics needed to visualise enterprise information in a visualisation type, and that the enterprise model is ecore based. In conclusion, no further requirements are imposed.

Listing 1 shows an excerpt of the Matrix-map HOT template where the QVT-RL transformation relation *XRootNodesRL* is created for the root *NodeMapping*-elements of the x-axis in the mapping model. Expressions in guillemets are part of the Xpand language. The resulting transformation rule states that a *MatrixMap*-element must exist in the resulting matrix-map ViM instance, with a *MatrixNode*-element in the list of x-axis root nodes, when the class *c* of the *targetClass* type (the type of the enterprise entity to appear on the x-axis) exists, under the conditions:

- the given parent reference of the enterprise element is empty (it is a root node),
- the enterprise model element is mapped to the ViM instance by the rule *ModelToMap()* and

To produce an executable transformation from a given mapping model instance, we make use of higher order transformations (HOTs). The overall interaction of the different components of our approach is sketched out in figure 5. The HOTs are expressed as oAW Xpand templates (see Listing 1) and create the QVT-RL transformation file required for the transformation process described above. The HOTA templates are

```

«DEFINE xRootNodes (Mapping mapping) FOR NodeMapping»
top relation XRootNodesRL {
  checkonly domain tar model:
    «fullyQualifiedName(mapping.container)»;
  checkonly domain tar c:
    «fullyQualifiedName(targetClass)»;

  enforce domain mm map : matrixmap::MatrixMap
  {
    xRootNodes = node : matrixmap::MatrixNode{}
  };

  when {
    «IF parentReference!=null»
    c.«parentReference.name».oclIsUndefined();
    «ENDIF»
    ModelToMap(model, map);
    EClassToXNode(c, node);
  }
}
«ENDDEFINE»

```

**Listing 1.** HOT Template excerpt

- an element of the enterprise model of the given type is mapped to a *MatrixNode* by the rule *EClassToXNode*.

When the rule is executed at run-time, all *MatrixNodes* with matching enterprise elements without parents are inserted into the *xRootNodes*-list.

The generated transformations correspond to viewpoints and can be stored for use by the intended stakeholders. When a stakeholder refreshes a visualisation, the transformation is fed with the up-to-date data from the EA repository and the desired view is produced.

We see a real benefit for the usability of EAM visualisations in providing a model (the ViM) for the configuration of viewpoints instead of having to write transformations, as the task is shifted from a programming to a modelling domain. The ViM captures all required information for producing views and the transformation generation is then fully automated (and transparent). Furthermore, by using modelling techniques instead of transformation programming, tool support for modelling becomes applicable to this task.

## 4 Conclusions and Future Work

We have shown how model transformations and higher order transformations can add a degree of flexibility to EA endeavours, while maintaining a high standard of rigorous modelling. The inherent benefit comes from catering for the separation of concerns by employing different techniques for different stages involved. We have constructed a first prototype of our approach and shown how visualisations can be developed so that they integrate well with customised and unique enterprise models.

We still see a knowledge-intensive and error prone task in the configuration of ViMs for viewpoints. Although well defined model constraints can reduce

the impact of configuration errors, an enterprise architect still needs detailed information on the effects of all the elements of a ViM. We have so far developed a simple pattern-matching algorithm for the matrix-map visualisation, which finds suitable candidates in the enterprise model for the x- and y-axes when an entity is chosen as an *Item* and integrated it into a simple GUI. We plan on taking this idea further by providing an intuitive interface to the visualisation configuration by matching entities to user selections according to the model characteristics required by the given visualisation.

## References

1. Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie: Systematische Darstellung von Anwendungslandschaften. In: Ferstl, O.K., Sinz, E.J., Eckert, S., Isselhorst, T. (eds.) *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*, 7. Internationale Tagung Wirtschaftsinformatik 2005, Bamberg, February 23-February 25, pp. 1443–1462. Physica-Verlag (2005)
2. The Open Group: TOGAF Version 9. Van Haren Publishing (February 2009)
3. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* 26(3) (1987)
4. Schekkerman, J.: How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework. Ebookslib (2003)
5. Henderson, J.C., Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal* 32(1), 4–16 (1993)
6. Hess, A., Humm, B., Voß, M., Engels, G.: Structuring software cities - A multi-dimensional approach. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 122–129. IEEE Computer Society, Los Alamitos (2007)
7. Postina, M., Sechyn, I., Steffens, U.: Gap analysis of application landscapes. In: The First Workshop on Service oriented Enterprise Architecture for Enterprise Engineering, Auckland, New Zealand, September 2. IEEE Computer Society, Los Alamitos (2009)
8. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: Reichert, M., Strecker, S., Turowski, K. (eds.) *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures - Concepts and Applications (EMISA 2007)*, St. Goar, Germany, October 8-9. LNI, vol. P-119, pp. 143–161. GI (2007)
9. Kurpjuweit, S., Aier, S.: Ein allgemeiner Ansatz zur Ableitung von Abhängigkeitsanalysen auf Unternehmensarchitekturmodellen. In: 9. Internationale Tagung Wirtschaftsinformatik, Wien, Österreichische Computer Gesellschaft, pp. 129–138 (February 2009)
10. TUM: Enterprise Architecture Management Pattern Catalog. Technische Universität München, Munich. Release 1.0 edn. (February 2008)
11. Gustafsson, P., Höök, D., Franke, U., Johnson, P.: Modeling the IT impact on organizational structure. In: 13th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2009), pp. 14–23. IEEE Computer Society, Los Alamitos (2009)

12. Närman, P., Johnson, P., Ekstedt, M., Chenine, M., König, J.: Enterprise architecture analysis for data accuracy assessments. In: 13th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2009), pp. 14–23. IEEE Computer Society, Los Alamitos (2009)
13. Frank, U., Heise, D., Kattenstroth, H., Schauer, H.: Designing and utilising business indicator systems within enterprise models - outline of a method. In: Loos, P., Nüttgens, M., Turowski, K., Werth, D. (eds.) *Modellierung betrieblicher Informationssysteme (MobIS 2008) - Modellierung zwischen SOA und Compliance Management*, Saarbrücken, Germany, November 27-28. LNI, vol. 141, pp. 89–105. GI (2008)
14. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A tool for enterprise architecture analysis. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 142–156. IEEE Computer Society, Los Alamitos (2007)
15. Addicks, J.S., Steffens, U.: Supporting landscape dependent evaluation of enterprise applications. In: Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P. (eds.) *Proceedings of Multikonferenz Wirtschaftsinformatik, MKWI 2008*, München, February 25- February 28, GITO-Verlag, Berlin (2008)
16. Maier, M.W., Emery, D., Hilliard, R.: Systems and software engineering - recommended practice for architectural description of software-intensive systems. Technical report, IEEE Standards Association (2007)
17. Postina, M., González, J., Sechyn, I.: On the architecture development of utility enterprises with special respect to the gap analysis of application landscapes. In: *Proceedings of the 3rd Workshop MDD, SOA, and IT Management (2009)* (to appear)
18. TM Forum: Business process framework (eTOM), release 8.0. Technical Report GB921, TM Forum (February 2009)
19. Buckl, S., Ernst, A.M., Lankes, J., Schweda, C.M., Wittenburg, A.: Generating visualizations of enterprise architectures using model transformations. In: Reichert, M., Strecker, S., Turowski, K. (eds.) *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures - Concepts and Applications (EMISA 2007)*, St. Goar, Germany, October 8-9. LNI, vol. P-119, pp. 33–46. GI (2007)
20. Czarnecki, K., Helsen, S.: Feature-based survey of model transformation approaches. *IBM Syst. J.* 45(3), 621–645 (2006)
21. OMG: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. Object Management Group, Needham, MA. Version 1.0 formal/08-04-03 edn. (April 2008)

# Cross-Organizational Security – The Service-Oriented Difference

André Miede\*, Nedislav Nedyalkov, Dieter Schuller,  
Nicolas Repp, and Ralf Steinmetz

Multimedia Communications Lab (KOM) – Technische Universität Darmstadt  
Department of Electrical Engineering & Information Technology  
Rundeturmstraße 10, D-64283 Darmstadt, Germany

**Abstract.** Service-oriented Architectures (SOA) are a powerful paradigm to address integration challenges for information technology systems in enterprises. The service-based integration of legacy systems and business partner systems makes it necessary to introduce and adapt suitable SOA security measures in order to secure the enterprise both within and for cross-organizational collaboration. While there is an active research community for SOA security, standard literature on the topic has not yet identified the influence of the SOA paradigm on security aspects in a structured manner, especially in an enterprise context. In our paper, we work towards this goal by identifying the main elements of cross-organizational SOA in the form of a conceptual model and by discussing these elements regarding their impact on security issues. Based on this, research challenges for SOA security are defined and structured.

## 1 Introduction

Globalization and technological advancements are major drivers for enterprises in the modern world. Enterprises face constant challenges to react to rapidly changing market requirements in order to stay competitive. An important factor to achieve this goal is the underlying enterprise information technology (IT). It has to provide solutions to model, operate, and adapt business processes efficiently and effectively. In the last years, the integration of both company-wide and inter-company IT systems has emerged as a big challenge in this context [1, 2]. Often, enterprise IT systems have not been designed and developed as a holistic concept but they have rather grown in a heterogeneous and organic fashion over time. This causes serious problems when changing things at the process-level due to the high amount of redundant data and code.

Paradigms such as *Service-oriented Architectures* (SOAs) [3] offer technological and organizational possibilities to evolve towards a more horizontally organized IT landscape and, thus, to improve the alignment between the business and the IT side – which is difficult in silo-like, vertically organized landscapes. SOAs are based on the “service” concept, where services can be seen as black boxes

---

\* Corresponding author, [Andre.Miede@KOM.tu-darmstadt.de](mailto:Andre.Miede@KOM.tu-darmstadt.de)

representing business functionalities. In a typical SOA setup, a service consumer learns about a suitable service provider via a service registry and then requests functionality from this service provider via an arbitrary communication channel. These services are used to assemble business processes as service compositions and may even cross enterprise boundaries, thus, enabling *service-based, cross-organizational workflows* [4,5]. In the last years, the SOA concept has become a successful way of addressing the challenges mentioned above, e.g., using *Web service technology* as an implementation.

In order to enable service-based, cross-organizational collaboration, the security of the participating systems, exchanged messages, and used communication channels has to be ensured. Achieving and guaranteeing basic IT security goals such as confidentiality, authentication, authorization, non-repudiation, integrity, and availability [6,7,8] is an absolute must in this context and still an active topic both in research and industry. Although security introduces additional costs and has an impact on the Quality of Service, unsecured business transactions are not an option in most scenarios.

The concrete application scenario for our research is the domain of service-based collaboration between organizations, e.g., enterprises. In order to execute its business processes, an enterprise often needs to integrate third-party services offered by different external service providers. An example for this is the retrieval of credit ratings for customers from an external rating agency in a credit application workflow or the request for different types of market data from a stock exchange for a financial derivatives workflow.

Based on this scenario, the goal of this paper is to analyze the major elements of cross-organizational Service-oriented Architectures regarding their impact on security. This is a step towards a better understanding of the security implications SOA has in general and in particular. It aims at identifying important research areas in this field in order to make SOAs more secure.

The rest of the paper is structured as follows: Section 2 gives an overview of the current understanding of security in SOA, based on the standard literature in this area. Section 3 introduces a conceptual model for cross-organizational SOA which identifies and orders its core elements. Based on this model, the security impact of the identified elements is discussed. Section 4 sums the findings up by including them into an integrated approach towards SOA security. The paper closes with a brief outlook on future work.

## 2 Related Work

Due to severe space constraints, this section gives only a brief summary of how SOA security is seen by major researchers in the field [9,10,11,4]. A more detailed discussion can be found in [12].

The main tenor of current SOA security research is that conventional security measures are not sufficient in the SOA context. In summary, the major aspects of SOA security in standard literature are the following: A switch from point-to-point-security towards end-to-end-security is necessary. For example, secure



communication channels using Transport Layer Security (TLS) or authentication by digital signatures are seen as a prerequisite but they need to be extended to fulfill the goal of building a secure architecture based on the SOA paradigm.

A dedicated security infrastructure is required to integrate different security mechanisms in a flexible and automated manner. The concept of security-as-a-service is seen as an important component of this infrastructure.

If any structure is chosen, the preferred structure for presenting the topic is according to the classic IT security goals (as described above).

Another trend is to equalize SOA security with Web service security, reducing SOA security requirements to Web service security standards and their configuration.

Having such an incomplete and rather unstructured understanding of security – a mission-critical element of any IT architecture – is a major obstacle towards a dedicated security analysis. Without a clear understanding of the impact the SOA paradigm and its elements have on security, an assessment of risks or the prevention of attacks is difficult.

### 3 Cross-Organizational Security – The Service-Oriented Difference

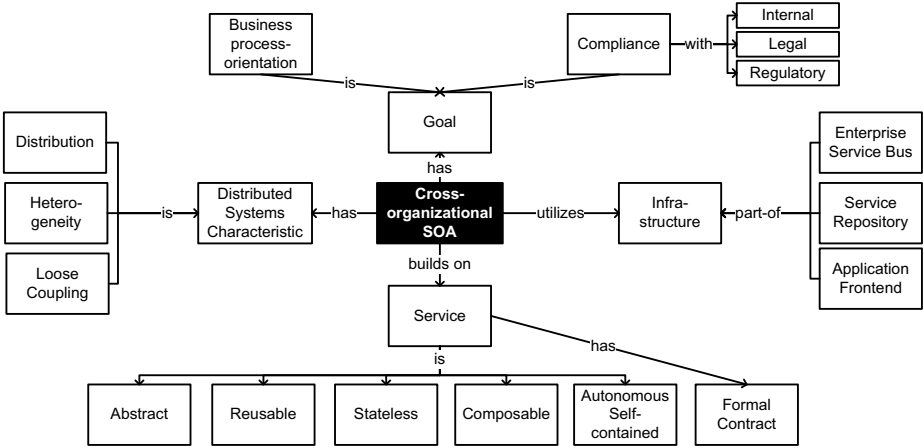
In this section, we propose two steps towards a more detailed and structured approach for SOA security. First, we introduce a means to capture the elements of cross-organizational SOA. Second, building on these elements, their impact on security is discussed. Both aim at getting a better understanding of the particular security requirements the SOA paradigm poses. Furthermore, it lays the foundation for an integrated SOA security approach and the proposal of a research agenda.

#### 3.1 A Conceptual Model for Service-Oriented Architectures

In order to understand the differences which SOA security requires for cross-organizational collaboration, we have assembled an inventory of the major elements used for such a scenario. These elements are represented and structured in the form of a conceptual model, which is based on SOA definitions and descriptions found in standard literature on the topic, i.e., books from practitioners and researchers [4, 5, 11, 2, 13]. Fig. 1 shows the conceptual model. A description of this model and its elements is omitted here due to space constraints. However, more details can be found in [12].

#### 3.2 Security Impact of SOA Concepts

The security analysis of the previously outlined SOA elements is based on typical security concepts such as *asset*, *threat*, and *vulnerability*. We briefly describe these concepts in the following, a simplified overview of how they relate to each other and to other important security concepts is shown in Fig. 2.



**Fig. 1.** Conceptual model for cross-organizational SOA (based on SOA concepts, characteristics, and ingredients described in [4,5,11,2,13])

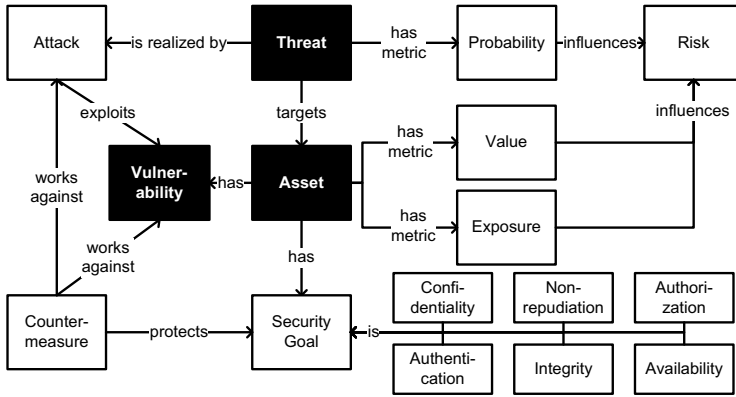
Assets are the objects and concepts to be protected from attacks [14]. The assets relevant for SOA security are shown above in Section 3.1.

A Threat is a possible way a system can be attacked [14]. Threats can be categorized in four classes according to their consequences [15]. While the general classes are the same for SOA, possible scenarios for them are briefly outlined: *Disclosure* is the unauthorized access to data. In an SOA context this could be for example the interception of messages or the forced access to a restricted service. *Deception* is the provision of false data (which is believed to be true). From a service consumer point of view this could be an attacker masquerading as a (known) service provider. *Disruption* aims at preventing an asset from correct operation, i. e., Denial of Service (DoS). From a service provider point of view this could be a “consumer” flooding services with requests. *Usurpation* leads to losing control of the asset to an unauthorized entity. A possible SOA scenario could be a hijacking of the service registry, resulting in requests being redirected to malicious services (see *Deception*).

*Vulnerability* is a point or characteristic which enables an attacker to bypass the security mechanisms of a system [6,16].

This is the focus of the following analysis. Each of the identified cross-organizational SOA assets is analyzed regarding its vulnerabilities and their exposure towards the threat classes.

1. *Goals* of an SOA offer new objectives for attackers and introduce new integration challenges.
  - (a) *Business Processes-orientation* is another driver for security requirements, e.g., the “four-eyes principle”, where two persons have to be successfully authenticated in order to authorize an action. It is an important aspect of many business processes and has to be supported by



**Fig. 2.** Important IT security aspects and their relationship (this is the core of a larger, more detailed metamodel for IT security [12])

the underlying IT. Thus, if these requirements cannot be communicated and modeled in a way which both business and technical experts can understand and implement, an SOA can lose many of its advantages. Especially, introducing security as an afterthought can put flexibility and the integration capabilities at stake.

- (b) *Compliance* attacks can seriously compromise an organization, e. g., by making an enterprise fail to comply with privacy laws or by manipulating or deleting audit records. This can result in fines, the revocation of licenses, or the loss of reputation and trust. Compliance vulnerabilities are not necessarily of a technical nature, but can also target governance processes within the organization, i. e., reference processes or key performance indicators.
2. *Characteristics of Distributed Systems* bring along with their advantages also their security challenges to the area of SOA.
- (a) *Distribution* of services extends the security perimeter. Where systems were rather closed with limited access, e. g., only from within a specific department, services make these systems now available (and thus attackable) all over an organization and across its boundaries. Along with distribution come also different owners of resources and services, which makes the alignment of security interests and their enforcement challenging – especially if a central, common control instance is missing. These uncertain conditions are a fruitful environment for attackers to exploit potential disputes of authority between organizations.
  - (b) *Heterogeneity* of communication and underlying systems is embraced by the SOA paradigm. However, different security concepts and implementations of the underlying technology have to be integrated seamlessly on the service layer, for example, to align countermeasures. Otherwise attackers can exploit the transitions between systems and security components, i. e.,

different user identities and roles. An important element towards a general security approach are open standards (cf. Section 2).

- (c) *Loose Coupling* is an important prerequisite for improved flexibility, but it can make it harder to establish trust between service parties due to spontaneous and even semi-anonymous communication among machines. Thus, reputation systems have to be established, for example. Side-effects must also be minimized, such as switching from a well-protected service to one which offers no protection at all. This makes it necessary to make services comparable not only regarding their functionality (e. g., semantics) and Quality of Service (e. g., SLAs) but also regarding the level of security they offer.
3. *Infrastructure* is necessary to operate an SOA but has both many new attack points and possible means to protect the organization.
- (a) *Service Repository* is an important, but very exposed part of the infrastructure and offers new attack scenarios. Availability is crucial here, as service consumers need the registry to find suitable service providers. Disruption threats, i. e., in the form of an DoS attack by massive fake requests, seriously affect the operation of a service-based organization. In addition, attacks on the communication between registry and service consumers and providers, i. e., to hijack requests, are likely and feasible, too. Last but not least, poisoning the repository with entries about non-existent or even malicious services, e. g., in order to undermine the trust between all participants, is yet another possible attack scenario.
  - (b) (*Enterprise*) *Service Bus* serves as a mediator for the communication between service consumers and service providers within and between organizations, thus, requires end-to-end-security for the whole way. Furthermore, an ESB creates an open system based on standardized protocols and can be used by attackers as well, e. g., to make service calls or to access service results of other consumers. Thus, the ESB has to integrate and automatize security mechanisms. On the other hand, the ESB must not only protect access to services but itself as well. Both disruption and usurpation of ESB operations are very attractive attack scenarios in order to take over control or to damage the organization, e. g., via Distributed Denial of Service attacks.
  - (c) *Application Frontends* can be – as described above – both an interface for a human user or another machine. A serious threat in this context is deception, e. g., in the form of phishing, where a human user is lured into providing his credentials to a malicious interface, which then misuses this information for its own purposes. Another threat arises in the form of usurpation, e. g., by manipulating or exchanging a batch program in order to execute a completely different process than intended. In general it is unclear, how information provided to an frontend, e. g., credentials, is used in the following services and what reaches the backend systems.
4. *Services* are the main components of an SOA and, therefore, a likely target for attacks.

- (a) *Formal Contract* increases the exposure of a service as its interface is publicly described and available. This can be exploited manually and automatically, i. e., by automatic scans for weaknesses of the interface or by guessing additional, non-described interface functionality. This threat must not be confused with a call for “security by obscurity”, it rather requires a critical revision of what information is made publicly available, what other information can be derived from this, and how both can be misused for attacks.
- (b) *Abstraction* has a multiplier effect on security. Where before a single application was secured, in an SOA this application can be abstracted by multiple services (or even service compositions). Now each of these abstractions has to be secured and, therefore, an attacker has many potential targets to choose from and often needs to succeed only with one. On the other hand, a security flaw in one application can affect the security of multiple services. Additionally, many abstraction layers increase the overall complexity of the system, at least in terms of security, i. e., propagating security information through all these layers safely. Attackers can specifically target different layers or their seams for weaknesses.
- (c) *Reusable* services can introduce very serious security side-effects, because not all contexts in which they might be used can be foreseen. Thus, different contexts have different vulnerabilities, which cannot be secured against in advance. A service might also be reused in way it was not intended for, creating a vulnerability. For example, a service to access employee information was originally developed to show a *subset* of this information in a web frontend. The service is then reused – or better misused – by an insider to access and to show the full information. In this context a dedicated SOA governance is required to coordinate the reusability of services, especially from a security standpoint.
- (d) *Stateless* services are a challenge for security considerations, because a security “context” – storing and communicating information about the current session – is very important, e. g., when calling the same service multiple times for a business process or when a service consists of multiple other services. Furthermore, maintaining no state allows attackers to record messages and to perform replay attacks if no suitable counter-measures are taken.
- (e) *Composable* services must integrate security mechanisms throughout the whole service chain. Seams for exploitation can arise by unclear organizational responsibilities and incomplete integration of different security technology. It is also possible for attackers to introduce malicious services into the composition, i. e., to take over control, to intercept sensitive data, or to falsify information.
- (f) *Autonomous/ Self-contained* services face the challenging question of “who is in charge of security?”. If the service is fully self-contained, it has to include security functionality as well, otherwise it would depend on dedicated security services or the infrastructure, i. e., for access decisions. Autonomous services can introduce additional vulnerabilities as the tight

coupling of security and service functionality introduces greater complexity and is more error-prone, e. g., if security updates are performed and one service is forgotten. Decoupling service functionality and security, i. e., via a security proxy, can leave the core service unprotected if the proxy can be bypassed and if the service can also be contacted directly.

In this section we have shown how common IT security concepts such as assets, threats, and vulnerabilities relate to the elements of a cross-organizational SOA. The general IT security requirements and threats basically do not differ from those an SOA faces. Single aspects of the above analysis may already be known from research and experience in the domain of distributed systems, but the SOA security challenge is to master the mix, interactions, and dependencies of all of them.

## 4 Conclusions and Future Work

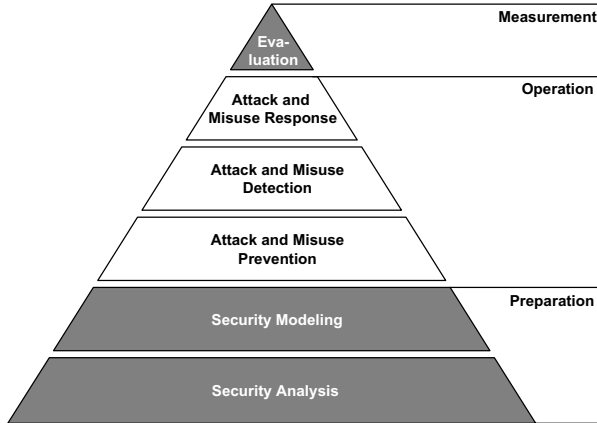
In this paper, we presented a model of cross-organizational SOA concepts in order to analyze these elements regarding their impact on security. *Cross-organizational SOA security* deals with the application of core IT security concepts such as threats, vulnerabilities etc. (cf. Fig. 2) on the elements of cross-organizational SOA such as loose coupling, composability etc. (cf. Fig. 1). We evaluated the security impact of single cross-organizational SOA elements and their relationships. While single security aspects of these elements are already well-known, i. e., for distributed systems characteristics, the combination of and relationships between the cross-organizational SOA elements as well as their impact makes cross-organizational SOA a special security challenge. Attackers thrive on uncontrolled complexity such as the manifold security challenges a cross-organizational SOA features. Thus, a structured approach to investigating and solving these challenges helps to decrease attack risks.

Furthermore, a common and clear understanding of research and industry topics in this area is necessary to work towards an integrated SOA security approach as depicted in Fig. 3. While the steps are well-known from general IT security, their contents have to be adapted to specific SOA-requirements, i. e., as we did in this paper for the analysis layer.

*Preparation:* security is a diverse topic spanning across departments, organizations and affecting both functional and non-functional requirements. This makes a thorough preparation process necessary, i. e., to identify and capture the SOA-specific impact on cross-organizational security. A *Security Analysis* includes a fundamental investigation of all involved assets, vulnerabilities they might have, threats they face, potential attacks, a calculation of associated risks, possible countermeasures, and so on. In this context, an analysis of the applicability of “Jericho-style security”<sup>1</sup> on the SOA paradigm might prove beneficial. In addition, *Security Modeling* is the basis for operating an organization and its IT systems. It consists of security-aware (business) process modeling, i. e.,

---

<sup>1</sup> <http://www.opengroup.org/jericho/>



**Fig. 3.** Overview of integrated (SOA) security approach

by annotations to or extensions of already existing languages such as the Business Process Modeling Notation (BPMN) to achieve compliance with laws and regulatory requirements. Furthermore, it includes the configuration of policies, e. g., to automatize standard security choreographies such as certain sequences of encryptions and signatures. Finally, on an organizational level this should include following risk management best practices, i. e., for security certification and accreditation of the used systems [17].

*Operation:* one of the main goals of security is to deal with intentional misuse and attacks. Thus, dedicated security controls have to be implemented in order to realize service security solutions for different purposes. *Prevention* works against the violation of general security goals. The Web service stack already includes various standards to address this issue. *Detection* is important to notice when prevention is failing, i. e., when attacks occur. Classic approaches include intrusion detection systems or firewalls, a cross-organizational SOA requires further message- and service-based approaches due to its abstraction from network operations *Response* has to be triggered when detection was successful. It can include, e. g., the activation of additional countermeasures, attack mitigation by cutting off any access to an asset, forensics, or counter-attacks.

*Measurement:* security solutions have different qualitative and quantitative impacts, which can and have to be measured. *Security Evaluation* uses various metrics to determine how effective prevention, detection, and response are. Important in this context is the concept “Quality of Protection” [18], which aims at bundling different measures for security and which is used in analogy to “Quality of Service”. In addition, the side-effects of security, i. e., on performance and usability have to be taken into account and evaluated.

Our next steps will be to extend the presented analysis and to build a model for SOA-specific attack scenarios and suitable countermeasures, i. e., in the *Internet of Services* scenario. Gaining a better and structured understanding of how attacks on SOAs work, what elements they consist of, and how to respond

to them allows for developing effective defense mechanisms, thus, brings us closer to safer service-based cross-organizational collaboration. Attack knowledge must not be left exclusively to attackers.

## Acknowledgments

This work is supported in part by E-Finance Lab e. V., Frankfurt am Main, Germany and BearingPoint Management and Technology Consultants. Furthermore, the authors would like to thank the SOA security working group of TeleTrusT Deutschland e. V. for fruitful discussions.

## References

1. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR, Englewood Cliffs (2004)
2. Melzer, I., et al.: Service-orientierte Architekturen mit Web Services. Konzepte – Standards – Praxis, 2nd edn. Spektrum Akademischer Verlag (2007)
3. Papazoglou, M.P.: Service-oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of WISE 2003, pp. 3–12 (2003)
4. Josuttis, N.M.: SOA in Practice: The Art of Distributed System Design. O'Reilly Media, Inc., Sebastopol (2007)
5. Newcomer, E., Lomow, G.: Understanding SOA with Web Services (Independent Technology Guides). Addison-Wesley, Reading (2004)
6. Eckert, C.: IT-Sicherheit: Konzepte – Verfahren – Protokolle. Oldenbourg (2007)
7. Schneier, B.: Secrets and Lies: Digital Security in a Networked World. Wiley, Chichester (2004)
8. Bishop, M.: Computer Security: Art and Science. Addison-Wesley, Reading (2002)
9. Kanneganti, R., Chodavarapu, P.: SOA Security. Manning Publications (2008)
10. Hafner, M., Breu, R.: Security Engineering for Service-Oriented Architectures. Springer, Heidelberg (2008)
11. Bundesamt für Sicherheit in der Informationstechnik: SOA-Security-Kompodium: Sicherheit in Service-orientierten Architekturen (2008)
12. Miede, A., Gottron, C., König, A., Nedyalkov, N., Repp, N., Steinmetz, R.: Cross-organizational Security in Distributed Systems. Technical Report KOM-TR-2009-01, Technische Universität Darmstadt (2009)
13. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR, Englewood Cliffs (2005)
14. Schneier, B.: Beyond Fear: Thinking Sensibly About Security in an Uncertain World. Springer, Heidelberg (May 2003)
15. Shirey, R.W.: Security Architecture for Internet Protocols: A Guide for Protocol Designs and Standards. Internet Draft (1994), <https://datatracker.ietf.org/drafts/draft-irtf-psrg-secarch-sect1/>
16. Anderson, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, Chichester (2008)
17. Ross, R., Swanson, M., Stoneburner, G., Katzke, S., Johnson, A.: Guide for the Security Certification and Accreditation of Federal Information Systems. National Institute of Standards and Technology (NIST) Special Publication 800-37 (2004)
18. Gollmann, D., Massacci, F., Yautsiukhin, A. (eds.): Quality Of Protection: Security Measurements and Metrics. Springer, Heidelberg (2006)



# Enterprise Oriented Services (Position Paper)

Daniel Oppenheim, Krishna Ratakonda, and Yi-Min Chee

IBM T.J. Watson Research Center  
19 Skyline Drive, Hawthorne, NY 10532, USA  
{music, ratakond, ymchee}@us.ibm.com

**Abstract.** We describe a service-oriented framework that supports how distributed enterprises can collaborate on doing work. Our model separates the concerns of *doing* from *managing* work. Work is modeled as a *capability* and can be provided as a *service* by some organization. A desired business outcome can then be described by its required capabilities. The framework enables dynamic composition of capabilities into *just-in-time service plans* that can be executed collaboratively by distributed organizations. A Hub is used to *manage* and coordinate the overall work. It comprises stakeholders from the collaborating organizations. The Hub's infrastructure enables them to see the big picture, detect issues early, decide on the best response, and quickly enact their decision. Executing service plans can then be modified in real-time, providing the enterprise with agility and flexibility.

**Keywords:** enterprise, adaptive processes, agility, human-services, work.

## 1 Introduction

It is widely recognized that an enterprise's survival and marketplace leadership largely depend on both the effectiveness and flexibility of its operations. As the marketplace evolves, or when unpredictable things happen, business must respond quickly. Enacting a response can be complicated, costly, and timely, especially when several organizations are affected. This is the problem addressed by this position paper: agility of business operations across distrusted organizations. We present a model and framework that can optimize the operations of collaborating enterprises through real-time modularity of work. As our focus is the overall enterprise, and our approach is to model work as chunks that can be serviced by interchangeable organizations, we term this concept *Enterprise Oriented Services*. Our approach can be viewed as a significant generalization of the "Distributed Enterprise Services" pattern [1, 2] combining Malone's Theory of Coordination [3].

There is an inherent tension between the need for standardization and predictability, that is essential for planning and marketing, and the need for flexibility and adaptation during execution, which is essential to meet the stated goals. Successful completion of complex projects depends on being able to identify problems early and then quickly enact an appropriate response. This inherent tension between *formal* and *flexible* is

clearly understood in the software development community as manifested by the *planned* and *agile* approaches [4].

Services Oriented Architecture (SOA) recognizes the need to blend formal structures with flexibility. Dynamic and adaptive processes that can “continually morph themselves to respond to environmental demands and changes without compromising operational and financial efficiencies” are not yet a reality, but have been identified as a research challenge. [5]. In our context of distributed work, the lack of flexibility is a severe limitation. An additional limitation is the paucity of support for people, and as will be explained later in this paper, especially for the role they must play when work is distributed. The addition of Human Tasks to BPEL is an attempt to address “how people can interact with business processes” but does not provide additional flexibility [6]. BPEL4PEOPLE interaction patterns, which include *4-eyes*, *escalation*, *nominations*, and *chained execution*, clearly do not address the needs of globally distributed teams that must be coordinated around a common business goal.

Recent developments provide a significant step toward bringing people and work closer to process. The *Amazon Mechanical Turk*, for example, exposes the work a person can perform as a service. This idea is expanded by the notion of *human services* in the context of Cloud Computing with the addition of composability through mashups [7]. Kern [8] initiated work on managing the quality of human-based eService. Vasko [9, 10] integrated orchestration of information derived from WS-BPEL with role-based collaboration of process participants in a web-service intended to intensify human interaction in social networks. Shcall [11, 12] expanded SOA even further by intermixing *Human Provided Service* framework with Web-Services.

Our approach separates between the *doing* and *managing* of work. We model the *doing* of any work by an organization as a *Capability*. This enables ‘chunks’ of work to be performed as a service by any qualified organization, and to a large degree, also encapsulate domain specific work. The capabilities required to meet a certain business goal can then be composed together in what we term a *Just-In-Time Service Plan*, providing modularity and composition. The framework dynamically assigns each capability within a plan to the most appropriate organization, and then provides continuous feedback on progress. This allows us to focus on the most critical issue: the management and coordination of the overall work. For this our model introduces the concept of a *Hub*. We discuss the Hub from a people and infrastructure perspective. The goal of the Hub is to bring together key stakeholders from the collaborating organizations so that they can see the “big-picture”, identify issues early, make informed decisions that balance conflicting concerns, and then enact these decisions. The infrastructure components of the Hub provide the necessary support, emphasizing dynamic composition of work as needed and runtime agility.

Next we provide three use-cases that demonstrate the problem and outline core requirements. We follow with a system overview of an enterprise and its operations that clarify important aspects that require support. We then present our model of cross-enterprise collaborative work, and end with a discussion.

## 2 Representative Use-Cases

The following three use-cases are from different industries and are typical of large enterprises that are globally integrated. They all boil down to this requirement: *agility*;

e.g. an ability to quickly and easily change operational procedures across an ecosystem of collaborating organization. The cases are real, but some details have been changed to protect the identity of the organizations.

## 2.1 Wireless Telecommunications

A leading US communications provider **A** had a strategic vision comprising two goals: increasing its speed to market, and increasing the yearly launches of new offerings from dozens to hundreds. Their business faced three fundamental challenges: intense competition, retention of existing customers, and an insufficient revenue stream. The key issue was losing competitiveness due to a lack of agility in responding to its competition. A careful analysis determined that in order to remain competitive this company must quickly introduce differentiating offerings; or simply put, reduce its time-to-market. The following goals were identified: (1) reducing the time required to enhance existing products or produce new products, (2) decreasing both development and maintenance costs, and (3) including customers and partners as co-creators of new value.

Processes governing existing products were optimized using LEAN [13, 14]. But when new products required a new process things remained problematic. Moreover, if a new product required a new capability that could only be obtained through partnership with another company, integrating **A**'s business processes with the new organization remained a stumbling block. Our model addresses this problem by (1) modeling *Capabilities* as services that can be executed interchangeably by any qualified organization, and (2) enabling new offerings to be modeled as *Just-In-Time Service Plans* that can operate across organizational boundaries. The need to include customers as co-creators is addressed by the Hub's infrastructure that provides a "plug-and-play" mechanism for both partners and providers.

## 2.2 Financial Sector

A leading financial corporation **B** has transformed over the past two decades from a Multi-National to a Globally Integrated Enterprise (GIE). It has presence in more than 50 countries. Over the years it has carefully placed clones of its business units in countries it identified as strategic. Each unit was given the autonomy to leverage local opportunities, and therefore adapted to local practices, policies, and regulations. Whereas each unit is doing similar work, in reality their processes vary considerably. As a result **B** is now experiencing pain in monitoring and coordinating its global operations. Moreover, finding out the true financial state of the corporation has become so complex that it was only done once a quarter and required intensive manual labor by highly skilled people. After a careful analysis **B** concluded that it wanted a drastically reduced set of simplified business processes that will be used uniformly by every business unit, yet still allow individual units to adapt locally. It also wanted the ability to provide its executives with a "real-time" view of its finances.

The two fundamental issues challenging **B** are (1) a standard way to manage similar work, yet allowing local variations for handling geo-specific policies and regulations; and (2) overall operational visibility and transparency. Our model of work as a *Capability* can provide the overall standardization required by **B**. Encapsulation of the

*doing* of work as a *service* permits individual organizations to make local changes within their internal procedures. As the overall operations would be managed utilizing a *Hub*, operational visibility and transparency across the global units would be supported on a primitive level.

### 2.3 Automotive Industry

A large company **C** in the automotive industry has its own business unit for developing the IT required by all its product lines. Over the last decade this unit transformed from a centralized organization to five centers that are distributed world-wide. The initial motivation for this was to leverage labor arbitrage. However, following the Wipro Factory Model [15], each location also specialized in a different aspect of the Software Development Lifecycle. Accordingly, each phase in the *Design, Build, Test, Deploy* and *Manage* development lifecycle is now executed by a different global team.

The main challenge facing **C** is the ability to effectively manage its IT development process. As it was originally designed for a centralized and co-located organization, it lacks the ability to coordinate, monitor, and govern new projects across its new distributed organizations. Our model of work as *Capabilities*, and *Hub* as a cross-enterprise operations center address these new needs.

## 3 The New Enterprise Eco-system

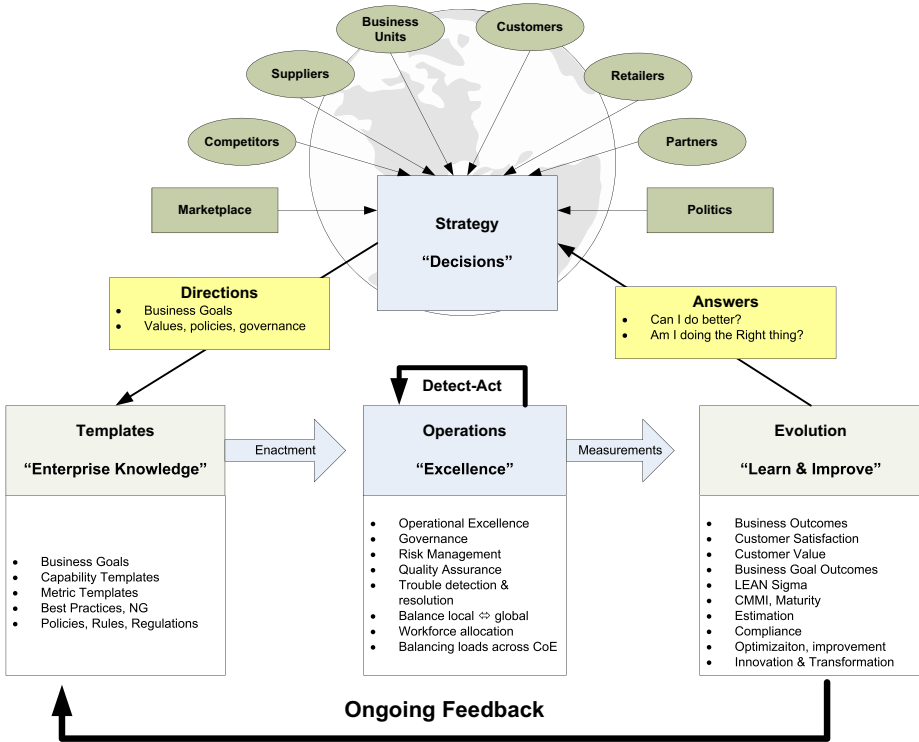
Globalization, internet, and technology are having a profound impact on all aspects of the enterprise, its organization, and the way in which people perform work. Freidman [16] describes the emergence of “Globalization 3.0” in 2000; ten forces and a triple convergence are shaping every aspect of business, politics, and society. Verna describes how businesses are “evolving into the networked patterns of living systems” and defines the organization as “a complex adaptive social system where people systematically cooperate to achieve a common purpose” [17, p. 14]. Malone describes how the low costs of communications are transforming society, businesses, and work and notes a move towards decentralization that empowers people and allows them to participate in decisions that matter to them [3p. 5]. People have always been at the center of business operations, but globalization is empowering people world-wide in radically new ways.

The enterprise is a complex system with many facets, including strategy, operations, organization, people, partners, co-producers, or service providers. The enterprise eco-system is represented by many different stakeholders that can each have different concerns that may even conflict with those of others: risk, quality, governance, intellectual property, tooling, methods, policies, or regulations. It is only by taking a broad systemic view that both recognizes and supports this seeming complexity of facets and concerns that a simple model that will support the broader operations of an enterprise can be devised.

### 3.1 A System View of the Enterprise

Figure 1 depicts a simplified system view of an Enterprise. The top of the diagram outlines its global eco-system of the marketplace, suppliers and retailers, customers,

partners, etc. This dynamic value-networks is constantly changing, evermore so with the accelerators of “Globalization 3.0” [16]. Each member brings with him capabilities that would otherwise not be available to the enterprise and all members collaborate to achieve a greater value.



**Fig. 1.** An Enterprise system view depicting its dynamic eco-system, relationship between Strategy, Operations, Knowledge, and Evolution, ongoing feedback loops, and the role of measurements and governance

Two important aspects are *Strategy* and *Operations* and the way in which they interact within the enterprise system. Strategy is primarily concerned with winning the market and is responsible for deciding what would be the “best” goals for the business. Given a goal, a careful analysis can determine the capabilities required to accomplish it [18]. E2E business processes are then designed to optimally control the flow of work between the different business units and organizations in the value-network. In the current state-of-the-art, specialized business applications, such as SAP or ORACLE, are used to automate work-flow. Supporting flow in such applications is costly, timely, and lacks flexibility once complete. Organizations can be therefore tempted to resist innovation and transformation but that can have severe consequences [20, 21]. Our framework to be described encapsulates such applications from process and flow, simplifies process design, and significantly shortens enactment time.

Operations are concerned primarily with getting things done well. Excellence is usually associated with the capability of doing work in a way that is consistent, repeatable, and therefore predictable. Operations are usually measured in terms of time, effort, and quality. Quality is often measured by a combination of customer satisfaction and post-delivery defects. As already noted, there is a fundamental tension between the desire for standard-repeatable-predictable operations and the fluid, often unpredictable nature of work. Our model addresses these concerns.

Two additional important dimensions are *knowledge* and *evolution*. The enterprise knowledge is its differentiating “secret sauce.” It comprises its best-practices, business processes, capability ‘how-to’ guides, measurement and monitoring strategies, and so on. Knowledge is important not only to effective operations but also to informed strategic decision-making. Harvesting new knowledge to update the knowledge base is highly desirable. The ability to act on new knowledge is a starting point for ongoing evolution.

### 3.2 Two Critical Feedback Loops

Two critical feedback loops can be seen in Fig. 1 that supports operations, strategy, knowledge, and evolution. As Strategy makes decisions regarding business goals and directions, it must also monitor their outcomes so it can answer questions from three fundamental perspectives:

1. *Operations*: am I functioning in the most effective way to meet my goals?
2. *Strategy*: Am I achieving my stated goals? Are my goals still the right goals or should I replace them?
3. *Innovation*: what can I do to improve my operations or strategy?

The data required to answer these questions is a combination of metrics collected during operations, its own knowledge base, and information from the marketplace. The ability to answer these questions is done through specialized analytics that is carried out by the Evolution team. Answers to these questions trigger innovation and transformation.

Operations have their own special feedback loop to enable what we call on-demand *detect-act* activities. It denotes the ability to detect an issue at the earliest possible time and take immediate corrective action (see discussion in the context of supply-chain [22]). In the context of people doing work this is critical for operational excellence, quality, coordination of work, and global governance. Our model inherently supports all of the loops. Runtime flexibility enables deploying new sensors to address new and unforeseen situations, alongside the triggering of their immediate, well governed resolution. Runtime flexibility then enables modifying both plans and services as required. Analytics deal with post-work analysis, the second feedback loop, which provides answers to strategy that can lead to ongoing evolution.

## 4 A Model of Work

This section defines our model of work leading to its encapsulation as a service. We factor work using an object-oriented methodology and take an artifact-centric approach to define the core model elements [1, 2, 14, 23]. In this way our model is

meaningful to stakeholders in the business domain yet also forms the underpinning to our IT framework.

#### 4.1 Definitions

For the sake of clarity and brevity our definitions are applicable only within the context of this paper.

- *Business Goal* defines some value that the enterprise wants to create. One or more capabilities may be required to obtain this value.
- *Capability*: the ability to perform actions that generate an outcome. As applied to human capital, capability is the product of expertise and capacity. (Adapted from Wikipedia). Our model associates a given capability with only one entity that will service it: a work-hub.
- *Compound Capability*: A capability can be composed of child capabilities and their relationships which can be represented in a directed acyclic graph (DAG).
- *Work*: human effort applied to realize a capability by performing the actions that generate its outcome (adapted from Wikipedia).
- *Capability Template*. A business artifact that specifies the attributes required for (1) providing the people who do the domain work with all they need to execute, and (2) providing the work-hub with all that it needs to manage, monitor, coordinate, and govern the work. Templates are static documents that represent a “best practice” of how to do work.
- *Enterprise Service, or Service*. A service is an enacted Capability Template that is appropriately configured, in the context of its execution, with the correct attribute values (data), people (team), and lifecycle (process). A service provides a team with everything required to *do* its work, and a work-hub with everything it requires to *manage* ongoing work, including monitoring and governance.
- *Service plan, or just-in-time Service Plan*. A service plan is a structure that contains one or more enacted services and their relationships that together service some higher level capability. Service plans are lightweight objects that can be easily assembled by reusing available Capability Templates. The structure of a plan can be modified at runtime, as can any aspect of a service.
- *Team*: a virtual entity comprising one or more people that *do* the work required to realize a capability. A team belongs to one organization. Each role definition specifies responsibilities, accountability, and skill level.
- *Work Hub*: A virtual organizational unit that is responsible for *managing*, coordinating, and governing the work around a service. The hub manifests our separation of concerns between *doing* and *managing*. Simple service requests could be implemented by one person. At the other extreme, complex requests may represent enterprise-level compound capabilities. A hub could then internally breakup this request into a service plan in which different capabilities could be managed by internal teams, other hubs, or combinations thereof.
- *Service Flow Engine*: the IT infrastructure that supports and automates all the above.

## 4.2 Artifact Centric Objects

We make a core distinction between templates and instances. Templates embody best-practices and are an important component of the enterprise's knowledge. They are created by experts, intended for broad reuse, and provide a standardized start-point for work. Templates can be continuously modified and improved with experience. Instances are managed by the Process Flow Engine. They are designed to be flexible and easily modifiable at runtime. When a Template is instantiated, its structure, attributes, and behaviors are copied into the instance. Then, specific values can be configured in the instance, such as people, dates, artifacts, policies, teams, governance, or metrics. Table 1 lists several of the more important objects.

**Table 1.** Templates and Instances

Template	Instance
Capability Template	Service
Role	Assigned Role
Team Template	Team
Work Hub Template	Work Hub
Metric Template	Metric
Compound Capability Template	Service Plan

The *Capability Template* and its *Service* counterpart are intended to provide everything that is required to both *do* and *manage* work. They have many attributes that are grouped to address different facets of the problem:

- *Service Composability.* The most rudimentary requirement for composability is a clear definition of *inputs* and *outputs*. In the context of people and work these are usually documents in the form of work-products, but they could be anything. Inputs specify everything that is needed by people to do their work, such as requirements, designs, or specifications. Policies, rules, and regulations are another category of inputs. Normative Guidance, examples, manuals, and anything that can help the person be more effective are a third category of inputs; there can be others. Output specifies what must be the outcome of the work; this is the value derived from this capability. The output might be a new work-product, document, or object, or it may be a coherent set of updates to existing work-products, documents, or objects; or any combination thereof.
- *Quality of Service.* Several complementing strategies are used overall to ensure operational excellence. In the context of a capability template quality of service is addressed by the specification of everything that needs to be monitored to ensure optimal operations, alongside the specification of governance when things are not. This would primarily include a list of metrics and related baselines against which they will be compared at runtime. There is infrastructure to enable the displaying of dashboards.
- *Governance* is critical when people do work, and is therefore built in as a primitive mechanism that is both systematic and dynamic. This section specifies one or more governance teams that will be assembled dynamically to



resolve any issue detected at runtime; different types of anticipated issues may require a different team composition. In addition to specifying roles, responsibilities, and accountability, a governance specification will outline the scope of the stakeholders required from the global value-network of collaborators.

- *Late service binding* covers interoperability of services. There may be more than one work-hub that can service a given capability. The attributes in this category provide information on required skills, capacity, required CMMI certification level, specific regulations, and so on. This enables the *Just-In-Time Service Plan* to dynamically allocate the most appropriate Work-Hub at runtime.

The Work-Hub template and instance represent everything that is required to *manage* work. In this respect, all work-hubs represent similar capabilities. As *managing* is different from *doing*, and as it is a capability that is required whenever any work takes place, it has earned its own representation. Different templates can represent needs of different business domains as well as needs of varying complexities of similar work. Managing the work of an entire enterprise is very different from managing work done by a single person. The following groups of attributes are worth noting:

- *Discovery*. Every hub must make available the list of capability templates it can service. In addition, the Hub publishes information regarding available roles, skills, and capacities. This is not unlike the idea of a Universal Description Discovery and Integration (UDDI).
- *Private Capabilities and Teams*. There are many activities that routinely take place that relate to *managing* work but do not contribute to the *doing* of work. Governance around issue resolution is probably common to all work. In the domain of software development, for example, one can name project-management, change-requests and defect-management. Different domains may have similar or additional activities. The concept is that by factoring out these activities from the *doing* of work, such activities can be handled by all hubs uniformly. This separation of concerns between *do* and *manage* makes it easier to evolve one without affecting the other. It also provides consistency and uniformity where it is needed without compromising the ability to be flexible and adaptive.
- *Live service plans*. Hub instances may manage more than one live Service Plans. After all, this is what the Hub is intended for.

### 4.3 An Operational Model of Work

The literature around Globally Distributed Software Development generally considers inter-team collaboration to be a key pain point [24, 25]. Our observation of co-located teams that must collaborate with globally distributed teams led us to different conclusions. We observed that teams enjoy working together in co-location and view frequent interactions with other teams as an annoying distraction. These realizations lead to our separation between *doing* and *managing* work. This approach minimizes the need to interact with other teams and thus maximizes the benefits of co-location. In honor of the memory of Richard Feynman we name this principle the *conservation of co-location*.

Another distinct advantage of this factorization is that we are conceptually moving from a procedural to a declarative model: we describe *what* should be accomplished (the capability) rather than *how* to accomplish it (process steps). This simplifies the composition of a *Service Plan* to the degree that it can be done *just-in-time*. It also provides runtime flexibility, especially the ability to replace providers dynamically.

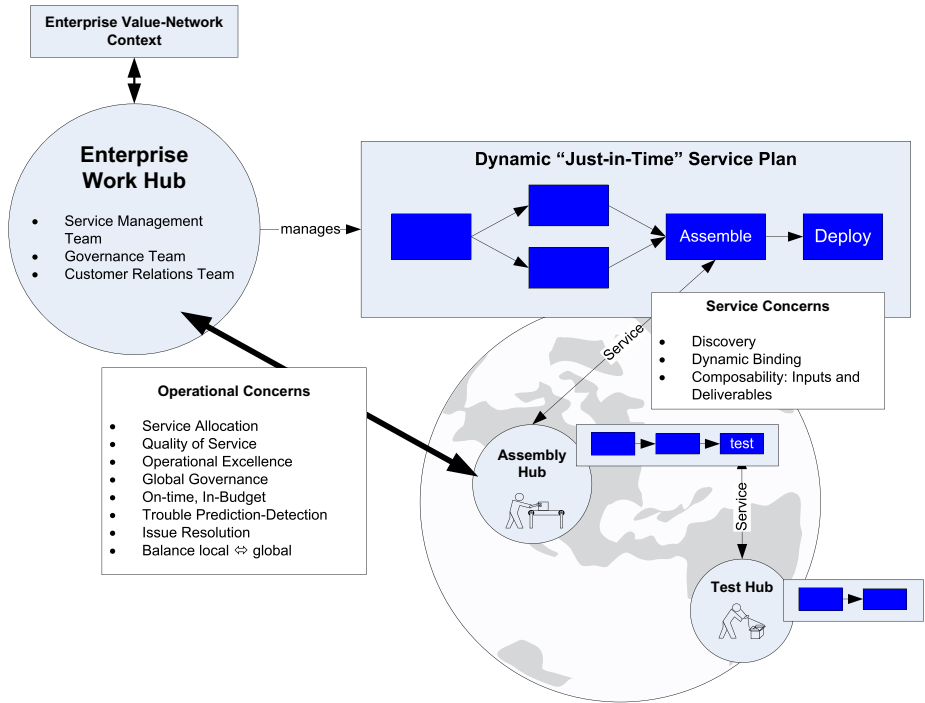


Fig. 2. The *manage-do* model of work

Figure 2 depicts our *manage-do* model which is both dynamic and recursive. The enterprise work-hub represents the highest level of the enterprise, managing the end-to-end work around some key business goal. For simplicity we have depicted a small set of representative hub-teams that deal with managing operations, governances, and the customer. The required capabilities are specified in a dynamic just-in-time service plan that is executed by the framework described in the following section. The capability *Assemble* is dynamically assigned to some Assembly Hub. In order to service the request this assembly hub uses the same model to manage its internal work. It dynamically creates its own service plan which comprises three capabilities. The first two capabilities may be worked upon by the hub's internal work-teams. However, for whatever reason, the third capability, *Test*, is dynamically assigned to a different hub. The Test Hub may be anywhere in the world, and belong to any organization.

The work-teams in each Hub need only be concerned about the *doing* of work specified by the *Service*, primarily with the *inputs* and *outputs*. Other service-oriented concerns include *discovery* and *dynamic binding*, but these are managed by the parent

hub, not the work-team. Hub-teams are primarily concerned with operations and Quality of Service. They require frequent coordination between their child work-teams and/or child or parent hubs. For example, any problem relating to the execution of the *Test* capability will be detected and handled by the *Assembly-Hub* without involving any of the work-teams unless some coordination is required. However, if the problem is serious enough, the *Enterprise-Hub* may be brought in, so that stakeholders from the enterprise value-network context, such as a customer CxO, can help make the right decisions. In this way the hub hierarchy also defines a management and governance context.

### 4.4 IT Framework Architecture

Figure 3 depicts a schematic architecture of an enterprise IT framework which supports globally distributed work in a service-oriented approach. At its core is a *Service Flow Engine* that can manage the instantiation of services, dynamic composition within Service Plans, flow of artifacts between globally distributed teams, and ongoing sensing for issues. And should an issue be detected the Service Flow Engine will, immediately initiate their resolution. The flow engine provides the flexibility and agility to modify any aspect of process and work.

There are four important system components. The *Templates* component supports the definition, modification, and persistence, of process templates. It is an important aspect of the Enterprise Knowledge. *Runtime* deals with template instantiation and configuration, service flow, and any related automation. *Health* assists with quality and operational excellence, primarily through a dynamic and flexible framework for sensing and alerting. *Analytics* includes a rich suite of data mining, pattern matching, prediction, and other algorithms that provide a key input to strategy, innovation and transformation. Persistence for these components is provided by several virtual data-bases. Runtime persistence provides extensive traceability such as ‘*who did what, when, and why*’.

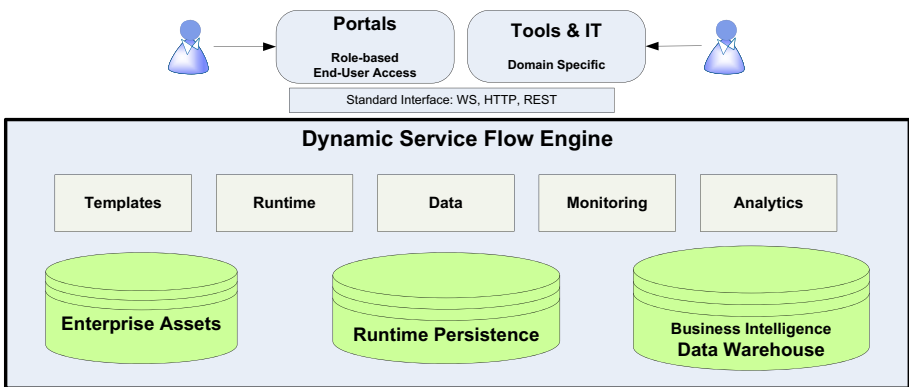


Fig. 3. Schematic architecture of enabling IT framework

Access to the framework is enabled only through carefully defined interfaces; and information is guarded by a data-centric security layer. Teams and practitioners access the framework through a portal. Access is role based and information presented to the user is further filtered by the individual person and his current activity. Tools used by users to do their work can also interact, and in some cases integrate, with the framework through external interfaces.

## 5 Discussion

We have presented a model of work within a service oriented framework that supports the operation of a globally distributed enterprise. We have demonstrated how SOA concepts can improve the way in which an organization can manage people doing work across geographical and organizational boundaries. Compared to the current state-of-the art, our *Just-In-Time Service Plan* provides considerable run-time flexibility alongside the agility to react through ongoing cycles of strategy & transformation. Referring back to the use cases presented in section 2 should make this more clear.

Enterprise **A**'s first pain-point was reducing "time to market" of enhancements or new products. There are two inhibitors: process modification, and then process enactment. Process modification can be achieved by either recombining existing Capability Templates into new Service-Plans, or by creating new Capability Templates that provide new capabilities. In either case, enactment is almost instantaneous when utilizing the Service Flow Engine. The second pain point of decreasing development cost is also addressed by the Capability Templates and Service Plans, and especially by their inherent ability to monitor and measure any aspect of operations in such a manner that process improvement and optimization can take place from one enactment to the other. **A**'s third requirement to include customers and/or partners as co-creators is the hardest to implement under current practice. However, as any third party who can service a Capability-Template can be included in a Service Plan, this becomes a non-issue.

**B**'s first concern is to have fewer processes that are standardized yet remain flexible. All **B** needs to do is to decide upon one Capability Template and have that used by all its business-units. Enactment will not be a problem. **B**'s second concern is increasing visibility and transparency. This is managed by our framework on a primitive level, so it is both systemic and dynamic. All **B** needs to do is to specify which "sensors" should be deployed in each Capability Template's Quality of Service (QoS) section. Another advantage of our approach should be noted. The standard approach for monitoring and dashboards is to create a Business Intelligence (BI) Data Warehouse (DW) into which business units are required to replicate data. The limitation of this approach is a lack of flexibility in changing the DW schema when new concerns arise. Our framework provides the runtime flexibility of being able to add sensors that can address multiple stakeholder concerns, even if these concerns conflict. Thus, our runtime persistence model complements the BI-DW and can provide increased visibility that is flexible and can be fine-tuned during each enactment.

Enterprise **C**'s primary concern is coordinating its end-to-end processes that were once co-located but now span its globally distributed business-units. The Service Plan replaces the end-to-end process, and each business unit can be managed and coordinated effectively through the Capability Template's Service instantiations. This separation of concerns between the *doing* and the *managing* of work effectively supports issues relating to globalization and collaboration.

However, this is but a first step in what we believe is a promising direction. There are many issues that need to be flushed out. There are at least two critical areas we feel should be explored in depth. The first is the relationship of our model and framework to Business Process Management (BPM). The Service-Plan is similar in concept to a business processes. However, it is simpler to define, and is dynamic and flexible when enacted. It unifies process with data; supports monitoring, measurements, and governance; focuses on business capabilities and their outcomes; and is deeply recursive in structure. When instantiated, plans are both configured and customized to the specific context of the enactment. In principle, one could easily create a new Service Plan on the fly to address similar needs. This might enable a new approach to business process management and evolution.

An additional aspect which we think merits closer investigation is relationship of the framework to the new paradigms of Cloud Computing. Cloud could be an ideal delivery platform for Enterprise Oriented Services, where “everything is a service.” Cloud virtualization of applications and IT could further assist in providing application and IT capabilities that are specific to a business domain, yet can be factored away from the Service Plans and Capability Templates. This could provide a separation of concerns between the general “how-to”—Templates—and the specific “enactment”—Instances and IT.

## References

1. Bhattacharya, K., Caswell, N., Jumaram, S., Nigam, A., Wu., F.: Artifact-centered operational modeling: Lesons from customer engagements. *IBM Systems Journal* 46(4) (2007)
2. Bhattacharya, K., Hull, R., Su, J.: A Data-Centric Design Methodology for Business Processes. In: Cardoso, J. (SAP), van der Aalst, W.M.P. (Eindhoven Univ.) (eds.) *The Handbook of Research on Business Process Modeling*. Information Science Publishing, and imprint of IGI Global, Hershey (2009)
3. Malone, T.W.: *The Future of Work*. Harvard Business School Press, Boston (2004)
4. Boehm, B., Turener, R.: *Balancing Agility and Discipline, A guide for the Perplexed*. Addison-Wesley, Reading (2004)
5. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: *Service-Oriented Computing: State of the Art and Research Challenges*. *IEEE Computer*, 64–71 (November 2007)
6. WS\_BPEL Extension for People – BPEL4People. A joint white paper by IBM and SAP (2005),  
<http://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/cfab6fdd-0501-0010-bc82-f5c2414080ed>
7. Tai, S.: *Cloud Service Engineering*. Public presentation made at IBM Research, Hawthorne NY, May 27 (2009)
8. Kern, R., Sirpins, C., Agarwal, S.: *Managing Quality of Human-Based eServices*. In: *IC-SOC 2008 International Workshops*, Sydney, Australia (2008)
9. Vasko, M., Oberortner, E., Dustdar, S.: *Collaborative Modeling of Web Applications for Various Stakeholders*. In: *Proceedings of the 9th International Conference on Web Engineering (ICWE)*, San Sebastian, Spain (2009)
10. Vasko, M., Dustdar, S.: *Introducing Collaborative Service Mashup Design*. In: *Proceedings of the 9th International Conference on Web Engineering (ICWE)*, San Sebastian, Spain (2009)

11. Schall, D., Troung, H., Dustdar, S.: Unifying Human and Software Services in Web-Scale Collaborations. *IEEE Internet Computing* 12(3), 62–68 (2008)
12. Schall, D.: Human Interaction in Mixed Systems – Architecture, Protocols, and Algorithms. Ph.D. Dissertation. Technical University of Vienna (2009)
13. George, M.: *Lean Six Sigma for Service*. McGraw-Hill, New York (2003)
14. Fleming, J.H., Asplund, J.: *Human Sigma: Managing the Employee-Customer Encounter*. Gallup Press (2007)
15. Upton, D.M., Fuller, V.A.: *Wipro Technologies: The Factory Model*. Harvard Business School Case No. 9-606-021
16. Friedman, T.: *The World is Flat, a Brief History of the Twenty-First Century*. Farrar, Straus and Giroux, New York (2005)
17. Verna, A.: *The Future of Knowledge: Increasing Prosperity through Value Networks*. Butterworth-Heinemann, Butterworths (2003)
18. MacDavid, D. W.: A standard for business architecture description. *IBM Systems Journal* 38(1) (1999)
19. Mann, C.C.: Beyond Detroit: On the Road to Recovery, Let the Little Guys Drive. *Wired Magazine* (June 2009)
20. Christensen, C.M.: *The Innovator’s Dilemma: The Revolutionary Book that Will Change the Way You Do Business*. Collins Business Essentials (2003)
21. Bliss, G.: *Principles of Modern Manufacturing and Supply Chain Management*
22. Nigam, A., Caswell, N.S.: Business Artifacts: An Approach to Operational Specification. *IBM Systems Journal* 42, 3 (2003)
23. Herbsleb, J.D., Moitra, D.: Global Software Development. *IEEE Software* (March-April 2001)
24. Cataldo, M., Herbsleb, J.D.: Communication patterns in geographically distributed software development and engineers contributions to the development effort. In: *CHASE 2008: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pp. 25–28. ACM, New York (2008)

# Automated Realization of Business Workflow Specification\*

Guohua Liu<sup>4,\*\*</sup>, Xi Liu<sup>3,\*\*</sup>, Haihuan Qin<sup>1</sup>, Jianwen Su<sup>2,\*\*</sup>, Zhimin Yan<sup>1,5</sup>,  
and Liang Zhang<sup>1,5</sup>

<sup>1</sup> School of Computer Science, Fudan University, China

<sup>2</sup> Department of Computer Science, Univ. of California at Santa Barbara, USA

<sup>3</sup> Department of Computer Science & Technology, Nanjing University, China

<sup>4</sup> Department of Computer Science & Engineering, Yanshan University, China

<sup>5</sup> Fudan-HangZhou Joint Lab. of Digital Real Estates

ghliu@ysu.edu.cn, liux@seg.nju.edu.cn, qinhaihuan@gmail.com,  
su@cs.ucsb.edu, gis001@zj.com, lzhang@fudan.edu.cn

**Abstract.** Business workflow assembles together a collection of tasks or activities in order to accomplish a business objective. Management of business workflows is facing many significant challenges, including in particular design, making changes, interoperations, etc. A key step in addressing these challenges is to develop techniques for mapping logical workflow specifications into executable workflow systems. In this paper we introduce a new artifact-centric workflow model called Artifact Conceptual Flow (ArtiFlow) and show that automated translation from ArtiFlow to BPEL is achievable. We also discuss technical issues in the translation.

## 1 Introduction

A business workflow assembles together a collection of tasks or activities in order to accomplish a business objective. It is a natural extension of the production pipeline concept. The goal of business process management (BPM) is to support design, execution, evolution (making changes) of a large number of inter-related business workflows [12] with the same or different schemas. While the BPM problems are not new, they have received significantly increasing interest from research communities over the last decade, due to the rapidly widespread use of computing devices and Internet technology. Indeed, “digitizing” documents and processes has made business workflow to operate effectively and more efficiently. The demand on underlying software systems for managing the digitalized versions has increased significantly. As a result, software systems become much more complex. However, software research has not yet developed mature technology for software system design and management in general and business workflow software in particular. Significant challenges are facing the business workflow application communities in many aspects of business

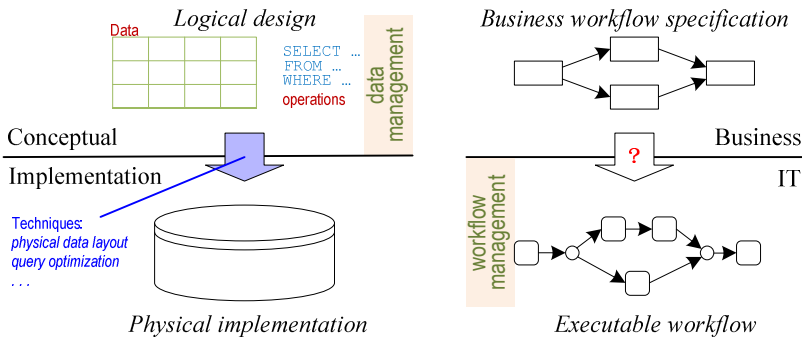
---

\* Work supported in part by US NSF grants IIS0145195, CNS0613998, and IIS0812578; and also by NSF of China grant 60873115, the National Basic Research Program (973) grant 2005CB321905.

\*\* Part of work done while visiting School of Computer Science, Fudan University.

workflow, including in particular, design, evolution, interoperation [8]. Technical difficulties include: (1) business process models used by the business managers are drastically different from executable workflows IT engineers develop and maintain, (2) key business performance indicators (KPIs) are hard to, (3) it is extremely difficult to change existing workflows according to the changes made to the corresponding business models, and (4) there is a lack of tools and support for business workflow interoperation. In addition, there are also significant barriers caused by differences between disciplines (business vs. IT) and between cultures (clients vs. managers).

In this paper, we make three position statements that outline a general approach for developing necessary technology for BPM challenges.



**Fig. 1.** Automated realization: *databases vs. business workflow*

**Statement 1:** *Automated or semi-automated realization of business workflow specifications is the key issue in tackling BPM challenges.*

Business workflow design starts with modeling to specify how the workflow should function. This step is important and usually done by business managers. The subsequent step is to turn the specification into an executable workflow. In this paper, we refer to this process of turning a business workflow specification into executable workflow as “realization”. Currently, realization is a labor-intensive task and often done in an ad hoc manner. Figure 1 (right) illustrates the two steps. In particular, workflow management happens at the software systems level. But it does not have to be the case. We consider the evolution of database systems as a comparison.

Prior to the mid 70’s, mapping logical database design to physical implementation was mostly done by hand. The resulting system naturally embedded many human decisions. The arrival of relational DBMSs in the 70’s brought automation to the design process with a suite of techniques including query optimization, physical data design, etc. (Figure 1, left). Automation of realization means systematic physical design that allows one to address data management problems at the conceptual level rather than implementation level.

Figure 1 illustrates a similarity between database and business workflows. Both have the steps of logical designs and physical realizations. The ad hoc nature of workflow realization makes it hard to reason about and maintain workflow systems, and make needed changes. We argue that automated translation of logical workflow specification into executable systems will provide a significant help to business workflow management.



**Statement 2:** *Artifact-centric models are most suitable for automated realization. In particular, the correspondence between logical models and executable workflows makes it easier to do BPM, in particular, handle KPI and modifications.*

Automating workflow realization is not easy. Traditional workflow modeling languages emphasize tasks (or activities) and the control flow. The lack of modeling data early makes it impossible to automate realization, since the logical specification concerns only “structural constraints” (tasks and control flow); without data, any workflow specification will not have the complete semantics. Clearly, the semantics has to be a necessary ingredient for automated realization.

A recent shift from process-centric to data-centric workflow specification is happening in the BPM arena and has shown promising signs [5, 8]. Conceptual models for data-centric workflow are emerging in business workflows [2], healthcare delivery [4], and digital government [8]. These models elevate the data being manipulated by the workflows to the same level of prominence as given to control flow in conventional models. A leading approach is the artifact-centric workflow models [10, 3, 7]. Business artifacts are the key entities referenced and manipulated in workflows. In particular, the lifecycle, i.e., how an artifact navigates through the workflow is an integral part of artifact. An artifact-centric modeling approach represents business workflows as artifacts (with lifecycle). This modeling approach appears natural to business managers [6] and software engineers [2, 4].

The artifact-centric workflow models made a significant step towards complete specification of workflow semantics at the logical level. The main part of this paper is to demonstrate that it is possible to automate realization of artifact-centric workflows and to establish clear relationships between elements in the specification and executable workflows.

There were prior efforts in mapping logical designs to executable workflows [11, 9]. However, these either focus on product design workflows [11], or workflows that composed of functionally independent tasks [9]. And thus they have limited applications.

We introduce a new variation of the artifact-centric workflows models, called “Artifact Conceptual Flow” or “ArtiFlow”. The model is used as a specification tool in an actual system being developed for a city government managing real estate transactions and licenses in China. With an example from this application, we illustrate the ArtiFlow model and a mapping approach for ArtiFlow specifications into executable (BPEL) processes. From this exercise, we conclude with the following.

**Statement 3:** *Automated realization of artifact-centric workflows needs events for execution control and data services for artifacts. BPEL plus data service wrappers is adequate but not ideal. Also, many technical problems including transactions remain to be solved.*

Although we are optimistic that automated realization of business workflow is an achievable goal, there are many technical problems to be solved.

This remainder of the paper is organized as follows. Section 2 illustrates the ArtiFlow model with an example. Section 3 outlines the ArtiFlow to BPEL translation strategy. Section 4 discusses several technical issues associated with the translation. A brief conclusion is included in Section 5.

## 2 An Artifact Conceptual Flow (ArtiFlow) Example

In this section, we introduce the Artifact Conceptual Flow (ArtiFlow) model through an actual workflow in an e-government application conducted in the Real Estate Administration Bureau (REAB) in the city of Hangzhou, China (with a near 7 million population). Being part of the city government, REAB manages all records and permits concerning real estate properties (mostly in type of apartments) within the city territory. In this e-government application, over one hundred workflows have been identified. The example presented here, called Commercial Apartments Preselling Approval (CAPA), is a simplified version of a typical workflow.

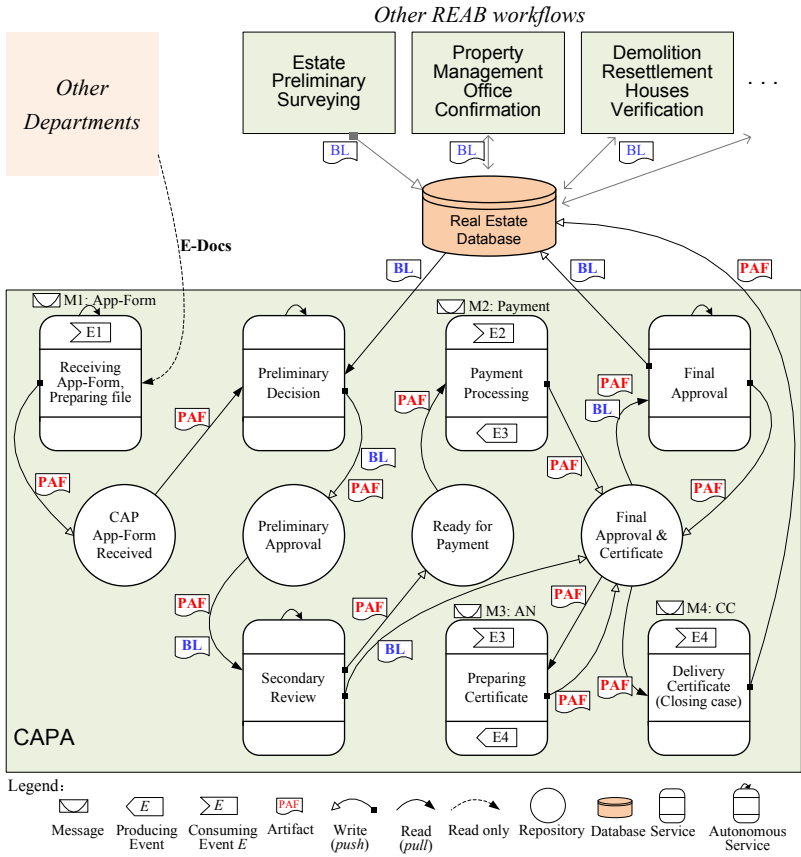
The need for CAPA is driven by the rapid economic development and a tremendous demand on the housing needs. “Preselling” refers to the practice of selling an apartment (by a developer) before the completion of building construction. Preselling is strictly controlled by the city: a developer must obtain a permit with a formal certificate for preselling a group of apartments, prior to putting the apartments on the market. REAB is the only authority that issues such certificates and CAPA is the workflow used in REAB for the presell approval process.

CAPA workflow requires results from several other workflows, including: *Estate Preliminary Surveying* (conducted by a company) that collects relevant files and sets up a *Building List (BL)*; *Property Management Office Confirmation* and *Demolition Resettlement Houses Verification*, both workflows are conducted by other departments in REAB, which may change the category of some apartments to ensure that the buyers’ interests are protected and that the habitants in the demolished structures on the site are properly placed in the buildings being constructed. CAPA workflow also needs data in E-Docs (see below) that exist and are managed outside of REAB.

The key information involved in preselling approval process includes the following:

- *Commercial apartments preselling application form (App-Form)* that includes the information and qualification certificates about the developer (applicant) and the details of the apartments: an apartment listing, the total area of preselling, etc.
- *External documents (E-Docs)* from other organizations needed for the approval process that include *Planning Permit of Construction Engineering (PPCE)*, construction progress certification of estate project, grant contract of land use right, permit of national land use, etc.
- *Building list (BL)*, a list of apartments with associated information. In particular, each apartment belongs to one of four main categories: *Commercial Apartments*, *Property Management Office (use)*, *Demolition Resettlement Apartment*, or *Other Public Auxiliaries*, and its status will be changed into one of several categories: *Pre-sellable*, *Sealed up by Court*, or *Frozen by REAB*, etc as the application is reviewed.

The CAPA workflow is triggered by the submission of an application for preselling by a developer. The REAB receives the application, justifies it according to relevant laws and regulations via a process consisting of automated services and human tasks, changes the status of listed apartments, and finally decided if the preselling application can be approved. An approved application will result in a corresponding *Pre-selling Approval Certificate (PAC)* from REAB. The developer can then proceed to sell those apartments listed in PAC.



**Fig. 2.** An ArtiFlow workflow schema for the CAPA Workflow

We now briefly outline the key elements in ArtiFlow and then show how the CAPA workflow can be modeled using ArtiFlow. ArtiFlow has four types of basic elements: “(business) artifacts”, “services”, “repositories”, and “events”. An *artifact* stores the essential information needed for completing a workflow execution including initial input data, temporary data needed during the process, the final results, and the information about the enactment at the current point (e.g., what has been done, context, etc.). An artifact-centric modeling approach starts from identifying the artifacts [5]. In ArtiFlow, an artifact type is represented by a name with an associated XML Schema type.

Unlike business data or objects, artifacts (types) in artifact-centric models should have their *lifecycles* specified. In fact, the lifecycles effectively provide a declarative functional specification of the workflow. The ArtiFlow model defines lifecycle specification as a graph, where nodes are either “services” or “repositories” and edges indicate how the artifacts move between services and repositories.

A *repository* may store one or more types of artifacts (instances). In an ArtiFlow graph, a repository is shown as a circle with a unique name indicating the “state” of the processing for artifacts stored in it.

In ArtiFlow, a *service* acts on one or more artifacts, e.g., it could create artifacts, or read the information in the input artifacts, could also access external information sources, and finally modify parts of the input artifacts. Services are shown as rectangles with rounded corners in ArtiFlow. *Directed edges* in an ArtiFlow graph are between services and repositories where: an edge label indicates the artifact(s) traveling along the direction, the solid end of an edge (either ► or ■) is attached to the actor of sending/fetching the artifact(s). Typically, a solid end is attached a service; when it is attached to a repository, it means that the repository will send it out (e.g., using a trigger). A dashed edge means read-only access of data (or artifacts). A service may be performed by a software system, a hardware device, or human. However, this distinction is not made in ArtiFlow. Instead, our model separates *invocable* (e.g., typical WSDL, REST) services from *non-invocable* or *autonomous* ones. This is because workflow management needs to control the execution: when and what invocable services have to run, and needed artifacts are ready in place for non-invocable services. ArtiFlow manages execution control through the use of “events”; in particular, each invocable service should have at least one associated event.

An *event* in ArtiFlow represents a change either *external* to or *internal* in the workflow execution that needs the attention from the workflow manager. Examples of external events include submission of an application form. Internal events are primarily generated by the end of a service execution. An event may have an associated *message* which records the information of the event. Each event has a handler (shown as a concave pentagon); each internal event also has a producer (shown as a convex pentagon). A handler inside a service indicates that the service should be invoked if the event happens.

The bottom of Figure 2 shows the elements and their graphical representation in ArtiFlow.

We now present the ArtiFlow model of the CAPA workflow described above. Fig. 2 shows the detailed CAPA workflow that interacts with other workflows in REAB. Two key artifacts identified are

- **Preselling Approval File (PAF)** that maintains necessary information for reviewing and approval. Specifically, it contains the original App-Form received from the developer and (space for) E-Docs. However, only the App-Form will be modified through the workflow, others are just for reference. A PAF artifact is created upon receiving a Presell Approval application from a developer.
- **Building List (BL)** that contains all data about apartments proposed to be sold in the preselling request, such as *building number*, *floor number*, *purpose*, and *status*. The physical properties (e.g., floor number) in a BL remain fixed whereas the purpose and status fields are clarified, checked, and approved as the CAPA reviewing process progresses.

Fig. 2 illustrates the CAPA workflow as an ArtiFlow graph and the (names of) three preceding workflows mentioned above. In the ArtiFlow model, the submission of a Commercial Apartments Preselling Application is modeled as an event E1 whose message contents contain the application form filled out by the developer. The CAPA

workflow starts upon this event. Specifically, the handler of the event E1 invokes the service *Receiving App-Form, Preparing File*. The service creates a new artifact instance of PAF, fetches the necessary documents E-Docs, packages the E-Docs information into the artifact, and finally stores the new artifact in the repository *CAP App-Form Received*. In general, each artifact during the execution of an ArtiFlow workflow is temporarily stored in a repository. A *Preliminary Decision* (through the service with the name) is then made and followed by a *Secondary Review*, both of which need the corresponding artifact *BL*. Note that these two tasks are not invocable, i.e., they are always running and act on an artifact when it becomes ready. When Secondary Review completes successfully, a payment notification is sent to the applicant. Upon receiving the payment (event E2), the payment is recorded into the artifact, the services *Preparing Certificate* and *Final Approval* complete the CAPA workflow, and as a side effect, a certificate is made and the *PAF* artifact is modified to note this fact. Finally, the certificate is sent to the applicant, and the final PAF artifact is deposited into a shared *Real Estate Database* so that other workflows can access. Note that the corresponding artifact *BL* has a different path whose contents are written by the Final Approval process before it is stored back into the share database.

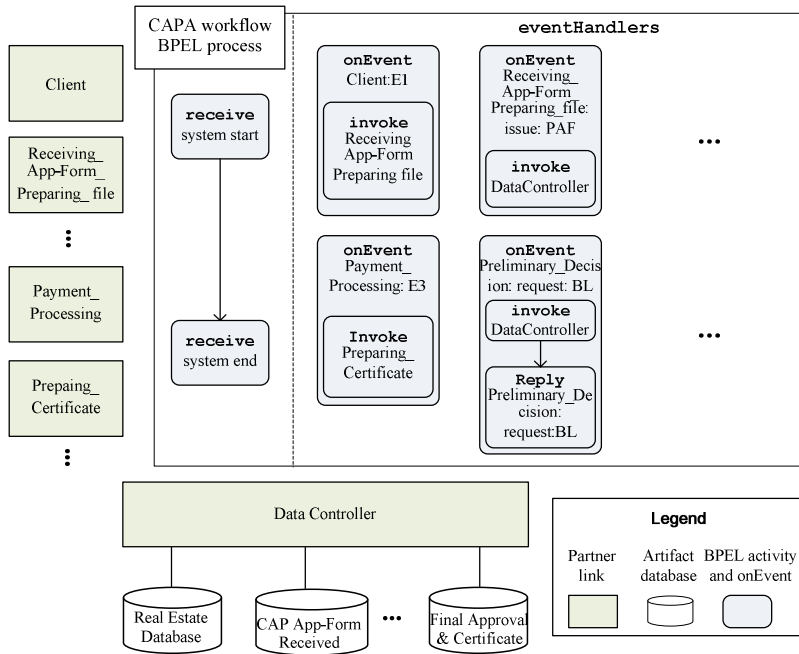
### 3 Translation of ArtiFlow to WS-BPEL: An Initial Approach

Our goal is to translate automatically an ArtiFlow workflow into an executable workflow. There are many choices for the target executable workflow language. As a starting point, we consider BPEL as the target language. In this section, we outline the key elements for the translation and illustrate it with the CAPA workflow example. We shall discuss several technical issues in Section 4.

In the translation, we focus on four primitive constructs of ArtiFlow: services, events, artifacts and repositories. In the translation, we only consider invocable services, since the autonomous ones need no control of their executions. For them, the workflow engine only needs to deposit the artifacts in their input repositories. To simplify the following discussion, we assume a fixed ArtiFlow workflow.

Clearly, BPEL does not provide artifacts/repositories. We construct a *data (controller)* service that handles all artifact-related operations in ArtiFlow, i.e., retrieving, storing an artifact. The service manages a database that stores the contents of all artifact repositories. The data service performs simple tasks: it receives a request and translates it into and executes an SQL command that perform data-related operation in the specified repository.

The ArtiFlow workflow graph under question is realized by a BPEL process that we call a *flow controller*. The flow controller handles all ArtiFlow events and invocation of services including the data service. Services in ArtiFlow are simply mapped to partner links of the flow controller. Events in ArtiFlow control the execution flow and they simply become BPEL events for the flow controller. The event handlers in the flow controller listen to artifact requests and event messages, when an event with a message arrives, the flow controller simply forward it to the corresponding service (or artifact repository) by invoking the service. If the service requests an artifact, it invokes the flow controller, which then invokes the data service. After the data service is completed, the result is converted to a message and sent to the service. The mapping from ArtiFlow primitives to BPEL sample code fragments is shown in Table 1.



**Fig. 3.** Translated BPEL process (flow controller) of the CAPA workflow

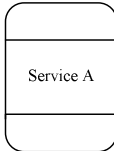

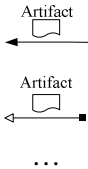
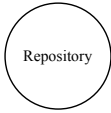
The translation from ArtiFlow to BPEL is done with the following three steps.

1. *Build partner links and interface according to the invocable services in the ArtiFlow workflow.* (We assume WSDL services here.) Specifically, for each invocable service, a partner link and a partner link type are constructed. Port types of the services are copied to the associated WSDL of the translated BPEL process, and each port type is mapped to one role in the partner link type definition.
2. *Construct databases for repositories and the data controller service.* We use a database for each repository and build the data controller service to perform all necessary data operations in ArtiFlow. We then add the partner link and partner link type for the data controller service.
3. *Add the event handlers to the (BPEL) flow controller.* The event handlers route each event and message to the corresponding service or repository. If the event enacts a service in BPEL, the service is invoked by forwarding the event message. If the message is a request for an artifact, the event handler sends the request and conditions to the data controller, after the data controller finishes the operation, it replies the service by forwarding the data controller's reply. Some more actions may be added to the on event branches according to the policies of the ArtiFlow process (such as the actions when no satisfied artifacts can be found).

Using the above steps, we can translate the CAPA ArtiFlow graph into BPEL automatically. Fig. 3 shows a part of the translated BPEL process (flow controller) for the CAPA workflow. The *Client* partner link is the interface of the BPEL process itself, and each service maps to one partner link, each artifact repository maps to one database. All event

and message routing are processed in the `onEvent` branch of event handlers in the BPEL process. “Receive system start” and “Receive system end” in a sequence are the only activities in the main process of BPEL which is used for system initiation and system termination. The main process is used to keep the process listening to the coming event and messages until the process terminates. The translated BPEL process can then be deployed on a BPEL engine.

**Table 1.** Mapping of ArtiFlow primitives to BPEL

ArtiFlow primitive	BPEL sample
<p>Service</p> 	<p>WSDL:  <code>&lt;partnerLinkType name="Service_A" ...&gt;</code>                      BPEL:  <code>&lt;partnerLink name="Service_A"</code>  <code>  partnerLinkType="Service_A" ...&gt;</code></p>
<p>Event for service enactment</p> 	<pre> &lt;eventHandlers&gt; &lt;onEvent partnerLink=Producer of E1 ...&gt; &lt;sequence&gt; ...   &lt;invoke partnerLink="Service_A" ...&gt; ... &lt;/sequence&gt; &lt;/onEvent&gt; &lt;/eventHandlers&gt;                     </pre>
<p>Artifact request and issue</p> 	<pre> &lt;eventHandlers&gt;   &lt;onEvent partnerLink=Service ...&gt;     &lt;sequence&gt;       &lt;invoke partnerLink=DataController ...&gt;       &lt;reply partnerLinke=Service ...&gt;     &lt;/sequence&gt;   &lt;/onEvent&gt; &lt;/eventHandlers&gt;                     </pre>
<p>Artifact repository</p> 	<p>Database operated by Data Controller Service</p>

## 4 Technical Issues and Challenges

The translation described in Section 3 is mostly straightforward as it addresses the core issue of realization without considering many factors in the operating environment, e.g., optimizations, and constraints. In this section, we briefly discuss a range of such issues that a practical operational workflow system must address.

### 4.1 Management of Logical and Physical Enactments

In this subsection we first describe the enactment of the BPEL process translated using the method described above, analyze its advantages and disadvantages, then

give an alternative translation strategy and have a comparison between the two strategies.

In Section 3, we outlined a translation of ArtiFlow workflows into BPEL: using a global event handler to deal every arrival event, regardless its target process, so-called global event handler strategy. Using this strategy, the ArtiFlow is realized as a controller over a collection of independent services. The order of the events arrival is unnecessary specified. Thus we can keep the most flexibility of the flow. A glance over the physical enactment of the translated CAPA workflow is given as following.

Once the global event handler receives the message *Client:E1* upon event *E1*, the service, *Receiving\_App-Form\_&\_Preparing\_File (RAFPPF)* is invoked. The arrival of message *Store:PAF* from *RAFPPF* will cause invoking service *DataController* to store the newly created artifact. At the arrival of message *Payment\_Processing:E3*, services *Preparing\_Certificate* is invoked...till an execution of the BPEL flow controller completes. It is easy to find out that there is no explicit relationship between the invoked services, i.e., the same flow controller may invoke two services for two different workflow enactments. Each service is invoked in response of the corresponding message, and the state-change is kept in the related artifacts. The event handler is stateless. The approach is simple and fairly easy to implement with a centralized messaging handling that may be easy to tune for performance. On the other hand, it is hard to trace or monitor a particular enactment of an ArtiFlow workflow because there is a single BPEL flow controller instance for all workflow enactments, and it needs additional mechanism for correlations.

An alternative translation strategy may be an *enactment aware* flow controller—each ArtiFlow enactment corresponds to its own BPEL flow controller instance. Under this approach each new enactment starts a new BPEL flow controller and there is a one-to-one correspondence between flow controller instances and enactments. Such an approach allows easy monitoring and auditing. A central issue is whether physical workflow enactments and logical enactments should correspond and the management of their relationships.

The notions of pool and lane in BPMN reflect the physical properties (location, agents, etc.) of the services and workflow. The ArtiFlow model does not currently have the corresponding concepts. Clearly, a practical realization should allow the specification of physical properties of services that can be used by translation algorithms.

## 4.2 Feasible Workflow Realization

A workflow execution may be distributed geographically over many locations, each of which may manage its own resources including data. Also, the services in a workflow may also have their own access privileges on, especially data in an artifact. We enumerate some of the questions below.

**Data modeling and management.** There are several issues concerning the design of the management mechanism for artifacts (and other documents) in a workflow. One issue is the data modeling for all the data involved, including artifacts. In the following discussion, we assume the use of relational DBMS in storing and access artifacts. On an extreme, one may view an artifact as a tuple and design a relation schema for each artifact class. This might be adequate if the artifact class does not have complex



data and the normalization. However, the artifact PAF has a rather complex structure. In particular, it contains information from external sources (this part of information will be read only and not changed throughout the CAPA workflow). In this case, it may be more appropriate to view each PAF artifact as a single database and employ the usual design techniques. To manage multiple artifacts, we could simply attach an artifact id to all tuples in the databases for involved artifacts. There is, however, the need for views. One use of views is to model the part of a PAF artifact that store the external information. The technical question is the design of the view mechanism that involves external data (e.g., from relation databases). Another use of views is that different services should see different parts of the PAF artifact. Such views should be updateable since the changes on the views by a service should be reflected in the artifact.

The second class of decisions concern how many database should we have. Clearly, one database per repository is sufficient but may not be optimal since we can always merge the databases into a single one and use tags to identify the repositories they are currently in. This, however, needs to take into consideration of the organizational boundaries and geographical locations (network speed). It would be desirable to develop a technical model for studying this problem.

**Distributed control flow management.** In the ArtiFlow realization outlined in Section 3, the BPEL process is the global event handler: it receives an event/request and dispatches it to an appropriate partner. One would naturally ask if services are distributed over many geographical locations, is it necessary to have multiple event handlers? Or even if all services are geographically co-located, is there a need for more than one event handler from performance considerations? Although the event handler behaves like a receptionist, a global event handler simplifies the execution machinery, but on the other hand, it adds dependencies and thus the cost for maintenance. By reducing the number of services an event handler would interact, the impact of changes to the services would be constrained to only the local handler.

**Operations on workflow execution.** In the CAPA workflow, it is very desirable to time service executions (e.g., in order to fulfill SLA for the approval workflow). In this case, if an exceptional situation happens, a service should be permitted to “suspend” the execution. This allows the clock to be stopped and the current context and state of the artifact to be saved. The dual operation would “resume” the enactment. In the current realization, the context only refers to the executing service. It is an interesting question to make the context notion definable for specific workflows. For example, one would allow the context to also include the information about one or more of the following: the number of active Preselling Applications or PAF artifacts at the moment, the current time, the environment for the workflow, e.g., the number of unsold units in previously approved Preselling filings. These issues are worth further investigation.

In addition to suspend/resume, there may be other operations on workflow executions. For example, one PAF application may be split into two after the preliminary approval is completed. Also, one may allow execution to proceed until some conditions are met, e.g., a specified time has been reached, or to terminate the workflow.

### 4.3 Workflow Transactional Properties

In all business processes, the notion of a transaction is vital. In ArtiFlow, a transaction mechanism has to guarantee to remain in “consistent” states for each workflow enactment. There are at least two categories of transactions, namely database transactions and workflow transactions, and their issues are different.

For database transactions, one can use the transaction mechanism provided by an underlying DBMS. Database transactions ensure that what’s written into the database being managed makes (logical) sense, i.e., satisfies the ACID properties. This is a good first step.

However, transactional properties in ArtiFlow workflows are different. For instance, artifacts in ArtiFlow are logical business objects; they could be composed of data stored in multiple databases, e.g., a PAF artifact in our running example. As a result, updating an artifact usually requires accessing multiple underlying databases that might be autonomously maintained by many stockholders. Relying only on the individual database transactions may not be sufficient. For example, if two database transactions commit and a third failed on one enactment (one or more artifacts), in what sense redo the failed database transaction is consistent with the workflow execution? Also, from the other angle, a correct workflow enactment may require some committed database transactions to be “compensated” or logically “rolled back”. It is unclear what the right notion of a “workflow transaction” should be, although it is apparent that it is not identical to a (local) database transaction.

The workflow transaction issues in ArtiFlow may be related to several WS-\* specifications, such as WS-Coordination, WS-Atomic Transaction, and WS-Business Activity developed by OASIS WS-TX TC. However, these proposals concern merely about how to “program” rather than what the logical notion is. The framework provided by the Phoenix project [1] may be a starting point to explore.

### 4.4 Safety of Workflow Executions

The final group of issues concerning realization is on the safety of execution. An ArtiFlow workflow is *safe* if every execution completes within finite steps. The safety notion is independent of whether the business logic is correctly formulated in the workflow. There are many possibilities that an execution may not complete. For example, if two services executing on different enactments are competing for resources (e.g., artifacts) they may get into a deadlock. Also, if the fetch conditions on artifacts are not properly formulated, some artifacts may forever stay in a repository. Other situations may include live lock, non-terminating executions, etc. It is clear that many of these properties have been well studied. Static analysis or dynamic checking techniques should be included in practical ArtiFlow realization algorithms.

## 5 Conclusions

In this paper we outline an approach to automatically translate business workflow specification to executable workflows. The possibility of translation is largely due to the use of a data-centric workflow model which describes detailed workflow semantics through data. This translation establishes a nice correspondence between the executable workflow

and components and the specification. The use of an artifact-centric model in the translation and the fact artifact-centric models are gaining acceptance in the BPM arena brings a huge potential: workflow management at business level, easy performance (KPIs) management and monitoring, support for interoperation, etc. However, there are many technical obstacles to overcome.

**Acknowledgments.** The authors are grateful to participants in group discussions that lead to this paper, including Leilei Chen, Weihua Chen, Qi He, Ying Wang, and Yong Yang.

## References

1. Barga, R., Lomet, D.B.: Phoenix: making applications robust. In: Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 562–564 (1999)
2. Bhattacharya, K., Caswell, N.S., Kumaran, S., Nigam, A., Wu, F.Y.: Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal* 46(4), 703–721 (2007)
3. Bhattacharya, K., Gerede, C., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 288–304. Springer, Heidelberg (2007)
4. Bhattacharya, K., Guttman, R., Lyman, K., Heath III, F.F., Kumaran, S., Nandi, P., Wu, F., Athma, P., Freiberg, C., Johannsen, L., Staudt, A.: A model-driven approach to industrializing discovery processes in pharmaceutical research. *IBM Systems Journal* 44(1), 145–162 (2005)
5. Bhattacharya, K., Hull, R., Su, J.: A data-centric design methodology for business processes. *Handbook of Research on Business Process Modeling* (2009)
6. Chao, T., Cohn, D., Flatgard, A., Hahn, S., Linehan, M., Nandi, P., Nigam, A., Pinel, F., Vergo, J., Wu, F.: Artifact-based transformation of IBM global financing. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) *BPM 2009*. LNCS, vol. 5701, pp. 261–277. Springer, Heidelberg (2009)
7. Cohn, D., Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Engineering Bulletin* 32(3), 3–9 (2009)
8. Hull, R., Su, J.: Report on 2009 NSF Workshop on Data Centric Workflows (2009)
9. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-driven design and configuration management of business processes. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 246–261. Springer, Heidelberg (2007)
10. Nigam, A., Caswell, N.S.: Business artifacts: An approach to operational specification. *IBM Systems Journal* 42(3), 428–445 (2003)
11. Reijers, H.A., Limam, S., van der Aalst, W.M.P.: Product-based workflow design. *Journal of Management Information Systems* 20(1), 229–262 (2003)
12. van der Aalst, W.M.P.: Business process management demystified: A tutorial on models. *Systems and Standards for Workflow Management, Lectures on Concurrency and Petri Nets* (2004)

# PeopleCloud for the Globally Integrated Enterprise

Maja Vukovic<sup>1</sup>, Mariana Lopez<sup>2</sup>, and Jim Laredo<sup>1</sup>

<sup>1</sup> IBM T.J.Watson Research, Hawthorne, NY 10532, USA

<sup>2</sup> Carnegie Mellon University, 5000 Forbes Ave, Pgh, PA 15213

{maja,laredoj}@us.ibm.com, nanalq@gmail.com

**Abstract.** Crowdsourcing has emerged as the new on-line distributed production model in which people collaborate and may be awarded to complete a task. While many existing services enable enterprises to employ the wisdom of crowd, there is no existing practice defined for integration of crowdsourcing with the business processes. We propose PeopleCloud, as the (1) mechanism to enable access to scalable workforce on-line, connecting it to the enterprise and (2) an interface to services required for crowdsourcing tasks. We define requirements for PeopleCloud, based on our experiences in employing wisdom of crowd to source business and IT information within the enterprise.

**Keywords:** Collaborative intelligence, Crowdsourcing, Globalization.

## 1 Introduction

As the physical and digital worlds are becoming universally connected, and computational resources and data are available beyond their immediate owner, it is now possible to effortlessly reach out to the masses, and externalize the “*function once performed by employees and outsourcing it to an undefined ... network of people in the form of an open call*”, the process which Howe [1] defines as crowdsourcing.

Leveraging the wisdom of crowds in the form of open calls and through marketplaces is not an entirely new paradigm. In the past companies have run competitions to engage their customers and end-users to contribute towards certain enterprise functions, such as advertising campaign design and specific problem resolution challenges [2].

Numerous purpose-built crowdsourcing solutions are becoming available, allowing enterprises to reap the benefits of scalable, global workforce, while lowering the cost of execution. For example, uTest is the marketplace for software testing services, offering real-world QA services through their community of more than 14,000 professional testers from 151 countries around the globe. Mob4Hire Inc reaches out to 86 countries, 2000 handsets and more than 130 mobile operators, helping mobile application developers access variety of testing platforms and testers in real field conditions. TopCoder.com is a community of over 140,000 skilled software engineers.

Despite some of the success stories, such as The Goldcorp Challenge [3] and Threadless.com, the actual realization of the promising advantages for enterprises from crowdsourcing are far from being well-achieved and pose an extensive range of interesting challenges along social, legal, economical and technical dimensions [4]. One of the key barriers for enterprises to effectively employ crowdsourcing is how to

integrate this process with the existing enterprise operations, as well as provide the incentives encouraging (honest) contributions. Tapscott and Williams [5] discuss how businesses can harness collective capability to facilitate innovation, growth, and success. In contrast, our research investigates applicability of crowdsourcing methodology within the enterprise.

Xerox's Eureka system [6] is one of the early examples of a crowdsourcing within the enterprise, where wisdom of crowd was employed to enrich enterprise support knowledge base. Enterprises address their business goals by formulating them as tasks, typically addressed by the subject matter experts. This approach is often inefficient because related know-how might be scattered among different teams and as a result hidden behind the teams' organizational structure. This problem becomes more significant as the enterprise's size and geographic coverage increase.

In this paper, we propose PeopleCloud, a model for encapsulating work tasks and people activities as a service. It provides a framework for business processes that span geographical and organizational borders. By defining a set of roles and their activities in the PeopleCloud, we take the first step in exploring the governance model for crowdsourcing-enabled processes. Finally, we describe the findings from deploying the PeopleCloud to source IT and business information within an enterprise.

## 2 Overview of PeopleCloud

Building on the requirements for a general-purpose crowdsourcing service, presented in [8], and our experience with employing wisdom of crowd within the enterprise, we define the PeopleCloud as an on-demand service system that spawns and manages scalable virtual teams of knowledge workers by building on the wisdom of crowds within an enterprise or across a value chain. PeopleCloud are design to increase effectiveness in building the transactive knowledge networks within an enterprise to execute complex and transformative knowledge-intensive tasks.

We envision enterprises being enabled to create highly customizable requests to access the PeopleCloud, while defining the business process standards and crowd requirements, as well as services required for their execution. PeopleCloud concept necessarily builds upon state of the art in CSCW, CHI, organizational theory and service design. In addition to the crowdsourcing features discussed in [8] PeopleCloud provides mechanisms for:

- a) Effective and sustainable expert discovery
- b) Supporting teamwork
- c) Task creation, precedence rules and integration to the enterprise systems
- d) Provisioning services required to execute tasks

Figure 1 depicts the main roles and their corresponding activities in the PeopleCloud.

**Requestors** initiate PeopleCloud process by defining the request, specifying task goals, task sequence and the crowd requirements. The requestor searches for crowd members, manages notifications, accepts bids and reviews submissions. A requestor may also cancel work demand or extend deadlines.

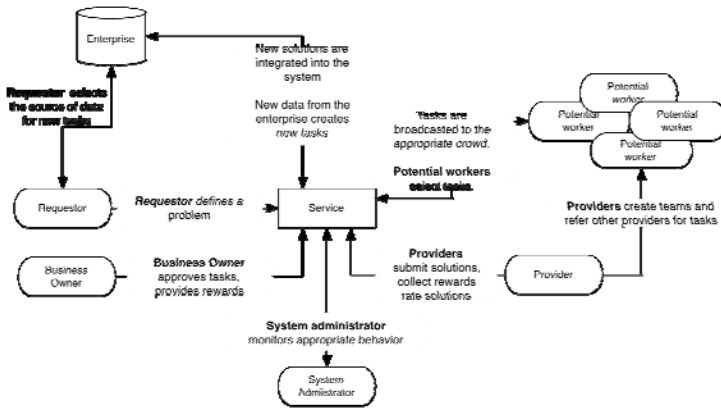


Fig. 1. Key Roles and activities in PeopleCloud

**Business owners** work jointly with the requestor, and are responsible for defining integration points, validating the successful completion and compensating the appropriate parties. They handle the selection of appropriate compensation models depending on the task complexity and the level of crowd expertise required.

The PeopleCloud is divided into two categories: the potential worker and the service providers. Potential providers may search amongst tasks or be suggested tasks. Once potential worker accepts a task, the requestor or the system must approve this. The **provider** executes the tasks they have selected. Each task has a specific space for the worker to provide their results and comments. Providers may return tasks that cannot be executed, or suggest another provider. On completion the provider submits the task in exchange for rewards, and may also rate other solutions.

The **collaborator** is responsible for forming teams, dividing tasks between team members, commenting on open tasks, suggesting providers and rating submissions. A collaborator could be either a provider or a requestor performing their roles in teams; and is crucial to expanding the knowledge network and to aiding the process of finding the correct providers for the task [9]. Thus, the PeopleCloud becomes a self-sustainable and self-discovery mechanism. In the enterprise context often we find existing repositories of task owners, based on their business responsibilities. This can in addition be used to jump-start the crowdsourcing process.

The **facilitator** is responsible for providing mediation if conflicts arise in crowdsourcing process, whether between requestors and providers as well as issues between team members. The facilitator monitors the behavior of the crowd, ensuring that requests are valid, ratings and rewards are fair and legitimate.

### 3 PeopleCloud for IT Optimization

In this section, we present the results from deploying a PeopleCloud to manage and control enterprise knowledge, specifically to identify the business use of the underlying physical IT infrastructure. Complexity of crowdsourced tasks, specifically in the

knowledge transfer domain [8] may vary, from the microtasks – requesting a single piece of information or activity to be performed, to crowdsourcing large-scale projects. In this work, we focus on discovery of "extreme information", that is, information that only a few possess, out of a pool of thousands of candidate information holders. IT inventory management is a domain where such information is hidden in the knowledge of individual team members, yet there is little transparency on who knows what.

**Background.** At present, information about the enterprise applications is typically stored in a centralized repository. This data is often outdated and incomplete, with limited bindings to the server infrastructure. At the same time, there is a lack of global knowledge within the enterprise (e.g. the different stakeholders, such as production control and application support experts possess partial information that can be leveraged to enrich and co-create the enterprise-wide view of the business capabilities of the infrastructure). Furthermore, interaction among employees exposes the inherently embedded social networks, which can be further exposed and employed to control the enterprise knowledge.

The goal of this crowdsourcing exercise was to discover, integrate and manage the knowledge about the business applications' infrastructure. Our PeopleCloud achieves this by harvesting Business-IT knowledge through wisdom of crowd and rewarding participants for the knowledge contributions (server to application mappings) and knowledge seeking behavior (e.g. finding other knowledge experts), which captures the global enterprise knowledge. Our PeopleCloud embeds capabilities to build, discover and maintain linkages among user communities to enable collaborative intelligence.

**PeopleCloud deployment.** We have deployed People Cloud to gather Business and IT information of more than 4,500 business applications within an enterprise, owned by more than 2,300 unique business application owners within the global enterprise. The initial seeding e-mail was sent to existing application owners inviting them to contribute their knowledge. Once they accessed the service, contributors had the option to refer their tasks to other experts in the enterprise. There was no access control, thereby allowing anyone from the group of application owners to complete the crowdsourced tasks. The system, however, had the capability to provide full audit trails, thereby enabling transparency of the community contributions. Following the seeding e-mail, a reminder for pending tasks was sent to the currently known application owner each week for the first 6 weeks of the run. After the sixth week, once we reached 90% task completion, the reminder frequency was increased to twice weekly.

**Effectiveness.** Within the first four days of deployment we have collected 50% of the targeted mappings between applications and servers. Table 1 shows the distributions of responses over the time.

The overall experiment increased the efficiency of the process by 30x, in comparison to the traditional approach (that involved two full time experts). They shared a collaborative virtual team-room, in which they documented the servers, applications and their owners, as they discovered them. They typically contacted the application owners through instant messaging, email or phone, and have then updated the master spreadsheet. Using this approach, two full time experts would take 2-3 months to collect the information about 140 applications and roughly 700 servers.

**Table 1.** Distribution of crowd responses over the time

% Completed	Duration in days
96	70
87	56
50	4
10	1

During this experiment, we have discovered that 5% of the original population of business application owners has changed. 220 referrals resulted and were completed from the access to the crowdsourcing service, providing an insight about the social network and dynamics within the enterprise that govern the flow of the knowledge that we seek. The feedback from the application owners has demonstrated the value of the knowledge that was collected. They were interested in the inferred information and the various communities that were formed, such as:

1. 'I would like to see the list of servers for my application.'
2. 'Which other applications are hosted on the same servers?'
3. 'On which servers is my application running?'
4. 'Who are the Application Owners of these applications?'
5. 'Which applications will get affected if I consolidate this data center?'

**Incentives.** Incentives play the key role in crowdsourcing efforts [10], both within and outside of the enterprise. Numerous incentive mechanisms exist, which may introduce inconsistency in crowdsourcing process and make collaborative production challenging [11]. Traditional award schemas are presently employed by enterprises, e.g. salary, performance bonuses. Incentives for crowdsourcing raise new legal challenges (e.g. compliance and taxation). Furthermore, existing HR policies may further impose limitations on participation of permanent and contract employees. The deployed PeopleCloud did include the capabilities allowing participants to collect virtual points for crowdsourcing task, as well as for any successful referrals. The points, however, at this stage were not exchangeable for a tangible, material award. The users had the ability to see their rating, compared to other participants. Finally, many users found the access to the consolidated knowledge that was a result of the crowdsourcing task, and incentive on its own.

## 4 Summary

In this paper we outlined the challenges for enterprises to reap the benefits promised by employing wisdom of crowd approach in their business processes. We proposed PeopleCloud, as the (1) mechanism to enable access to scalable workforce on-line, connecting it to the enterprise and (2) an interface to services required for crowdsourcing tasks. Finally, we described our experience in deploying PeopleCloud within the enterprise to execute IT inventory management exercise.



**Acknowledgments.** We would like to thank Kumar Bhaskaran for insightful discussions on PeopleCloud.

## References

1. Howe, J.: The Rise of Crowdsourcing. *Wired* 14(6), <http://www.wired.com/wired/archive/14.06/crowds> (accessed October 23, 2008)
2. Surowiecki, J.: *The Wisdom of Crowds*. Anchor (2005)
3. GoldCorp Challenge, <http://www.goldcorpchallenge.com/> (accessed November 14, 2008)
4. Brabham, D.C.: Crowdsourcing As A Model For Problem Solving: An Introduction And Cases. *Convergence. The International Journal of Research into New Media Technologies* 14(1), 75–90 (2008)
5. Don, T., Williams Anthony, D.: *Wikinomics: How Mass Collaboration Changes Everything*. Portfolio Hardcover (December 2006)
6. Galegher, J., Kraut, R.E., Egidio, C. (eds.): *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*. Lawrence Erlbaum Assoc., Inc., Mahwah (1990)
7. Vukovic, M.: Crowdsourcing for Enterprises. In: *International Workshop on Cloud Services at 7th IEEE International Conference on Web Services (2009)*
8. Argote, L.: *Organizational Learning: Creating, Retaining, and Transferring Knowledge*, 1st edn. Kluwer Academic Publishers, Dordrecht (1999)
9. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with Mechanical Turk. In: *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, CHI 2008, Florence, Italy, April 05 - 10 (2008)*
10. DiPalantino, D., Vojnovic, M.: Crowdsourcing and all-pay auctions. In: *Proceedings of the Tenth ACM Conference on Electronic Commerce, EC 2009, Stanford, California, USA, July 06 - 10 (2009)*
11. Bartlett Christopher, A., Ghoshal, S.: *Transnational management: text, cases, and readings in cross-border management*, 3rd edn. Irwin/McGraw Hill, Boston (2000)

# Public Disclosure versus Private Practice: Challenges in Business Process Management (Position Paper)

Stacy Hobson<sup>1</sup>, Sameer Patil<sup>2</sup>, and Xuan Liu<sup>1</sup>

<sup>1</sup> IBM T.J. Watson Research Center  
Hawthorne, NY

stacypre@us.ibm.com, xuanliu@us.ibm.com

<sup>2</sup> Department of Informatics, University of California, Irvine  
Irvine, CA 92697-3440  
patil@uci.edu

**Abstract.** This paper explores the gap between *actual* work practices and their articulation. Our goal is to bring this gap to the forefront as an important consideration for operational process modeling. Business process models presuppose accurate disclosure of employee work practices. However, the presence of a gap between personal practices and their public disclosure is a challenge for accurately representing the true nature of business operations. We describe a field study of the working practices of a municipal organization where we identified this gap. We then offer several underlying motivations that contribute to the existence of this disparity. These findings hold important implications for global enterprises, and for process modeling efforts in general.

**Keywords:** Information Sharing, Business Process Management, Privacy, Process Model.

## 1 Introduction

Businesses and organizations are increasingly becoming interested in ways to manage their operational processes efficiently and, whenever possible, utilize information systems to support the work practices of their employees. Business Process Management (BPM) is the field that encompasses these efforts and offers concrete approaches to enable specific business or organizational goals. In order to streamline operations, design appropriate workflow systems, and to identify and eliminate work redundancies, the *correct* elicitation of processes is sought.

This work-process identification relies heavily on participation of, and disclosure by, the persons who carry out the work, and assumes reliability of the information obtained. However, there has been little focus in BPM on the gap between the information an employee may disclose publicly and the *actual* details of their work practices. This gap, which emerges through concerns of privacy and information sharing, leads to practices that are often undisclosed and may greatly impact the accuracy of the resulting information representation (e.g. a business process model).

This paper describes a field study of a municipal organization in which we encountered this gap when trying to understand intra- and inter-departmental work processes and communication. Based on observations and interviews conducted during the field study, we initiate an examination of motivating factors for withholding public disclosure of the details of one's work in Section 3.2. These findings hold important implications for business process management efforts (Section 4.2). We plan to extend this work with investigations that include additional governmental and corporate entities.

## 2 Background

A key component of business process management (BPM) is process identification and documentation as the foundation of workflow systems. van der Aalst [1] states that earlier approaches to BPM encouraged the adaptation of business processes to available information technology, but more recent approaches emphasize explicit modeling of operational processes. Numerous methods are employed to understand business processes including consulting sessions with the employees who enable the operations. These methods, in general, rely heavily on the belief that the information shared by the employees correctly reflects their actual work practices. However, we maintain that people invariably distinguish between the actual details of their practices and how they describe them publicly.

Literature in the Human to Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW) fields shows that people distinguish between information that should be publicly available and information that they want to remain private. Some examples include investigations of privacy in relation to e-commerce [2], ubiquitous computing [3], and social networks [4]. People are often concerned about whom information should be shared with and in what form, and they have complex understandings of others' views of them, and the possible effects of disclosure [5]. Similar work in regard to privacy in work environments includes studies of workplace monitoring [6] and groupware [7]. We focus here on concerns related to privacy when disclosing and describing one's individual work practices. We highlight the underlying issues and discuss their impact on BPM and process elicitation efforts.

## 3 Investigation of Work Practices of a Town Government Office

Information sharing in the workplace has been noted to serve as a foundation for collaboration [8] and to reduce duplication of effort [9]. As noted above, detailed information sharing regarding work practices is also critical for building models of the work and designing information systems that can support these practices.

### 3.1 Setting and Methodology

We conducted a field study that investigated the work practices of a municipal organization. Over a period of five months, the authors organized twelve site visits employing informal and semi-structured individual and group interviews, work process observations, and formal discussions with employees. Nine employees from three departments and two members of the municipality's management participated in these

sessions. Each participant used multiple information systems as part of their work routines. The work of the departments was heavily interconnected; each department produced information for and/or received information from another department.

Our goals in this endeavor were to understand how the employees accomplished their work and how information was transferred inter-departmentally. Towards this end, we held group interview sessions with the employees from two departments and also conducted individual interviews to refine the knowledge gained from the group as a whole. In total, three group interviews were conducted, and fourteen individual interviews (including follow-up interviews to elicit additional details or clarifications). Notes from each session were documented by at least two members of our research team and were discussed and analyzed in detail afterwards. Additionally, significant amounts of electronic and paper records were collected to enable realistic simulations of work efforts and data flow amongst the employees and aid in process analysis.

During our investigations we noted that employees provided more detail regarding their personal work practices in individual interviews than in group interviews. In fact, one of the employees commented that she specifically did not wish to provide finer details of certain practices and to reveal a few other practices during the group discussion. Observations that people communicate different levels of information in individual versus group settings have been reported in other literature [10,11]. We chose to investigate this point further to determine the underlying motivations for this type of phenomena. When reconciling the data collected from the observations against the information gained from the individual and group interview sessions, we discovered that the gap between personal work practices and their public descriptions<sup>1</sup> often contributed to misunderstandings and lack of communication regarding each other's information needs, duplication of effort, following of unnecessary procedures, and so on. Moreover, it also hampered our own efforts of modeling the work activities of the two departments. We suspect that our experience is typical of most organizations – government or corporate. Thus, we believe that by exploring this gap further, we can make important contributions to how process models are generated and utilized.

### 3.2 Motivating Factors for Non-disclosure of Work Practices

Through the experiences of our field study, we formulated possible motivating factors for the hesitance of or incompleteness in public disclosure of one's private work practices and we describe the factors in detail below<sup>2</sup>. We do not presume this list to be exhaustive; extending our work to more organizations will likely reveal additional factors. Moreover, it should be noted that although we present the factors separately for the sake of facilitating their description, these are often intricately interrelated.

*Specialized skills* – A person may possess skills that other employees in the workplace do not share, e.g. accounting, programming, or computer-aided design knowledge.

---

<sup>1</sup> Although the individual interviews are more private compared to a group session, these are still “public disclosures”. As a result, it is quite likely that there was a further gap between the descriptions of work practices in the individual interviews and how the work is *actually* carried out. The discussion in Section 3.2 relates to both of these gaps combined.

<sup>2</sup> Part of this discussion was inspired by observations reflected from the authors' experiences in other process modeling efforts.

These skills may be used to develop reusable techniques that help make work more efficient. However, descriptions of the methods may be generalized to avoid communicating details of these methods or personally developed techniques. The perceived drawback to disclosing the methods may be that the employee is rendered less competitive; other employees may then learn the same techniques thereby reducing the ability to differentiate one's working methods from others. An example of this was seen in our study; an employee created an excel spreadsheet to perform computations and track historical data. The employee ensured that the others in the department and organization group were not aware of the existence of this customized spreadsheet, but the others were aware that this employee was the person to ask for information related to a specific area.

*Job security* – The above example also points to the motivation for protecting one's employment. People may not want to share their work methods to help ensure their continued importance to the organization. Disclosing personal work practices gives others the opportunity to learn and use these methods and, as stated above, may limit a person's ability to distinguish themselves and their work from that of their colleagues. Work practices that help ensure job security can be tied to one's specialized skills (as discussed above) or could be a process or efficiency that others have the knowledge to implement, but have not yet figured out. Conversely, disclosure concerns may also arise from the fear that the discovery of inefficiencies or unnecessary processes may lead to job loss through consolidation of that work or replacement by an automated system that can accomplish the same work goals.

*Anxiety regarding being judged* – People are often hesitant to disclose their work practices because of anxiety regarding how their work would be judged by those to whom it is accessible, and in turn, regarding how their value as a worker or their value to their team would be evaluated. Co-workers may also criticize the methods, or compare them for effectiveness or efficiency against those of others. Ackerman [5] offers support for this factor, stating "*However, it has been found that people are aware that making their work visible may also open them to criticism or management; thus, visibility may also make work more formal and reduce sharing.*" In other words, public descriptions of an individual's work practices are tailored to avoid criticism and conceivable side effects, such as embarrassment, micro-management or surveillance of one's work.

*Convenience* – It is understandable that people may not want to spend significant amounts of time explicitly detailing their work methods. This could not just be bothersome for the discloser but also a burden on the recipient(s). As Schmidt [12] states "*An actor will thus routinely expect not to be exposed to the myriad detailed activities by means of which his or her colleagues deal with the contingencies they are facing in their effort to ensure that their individual contributions are seamlessly articulated with the other contributions*". It is indeed plausible that the recipient(s) may find some of the detail useful in their own work. However, depending upon the amount of information, level of detail, and the time consumed, this could be perceived as a fruitless effort by the discloser. Also, the practices being described may be complex, and therefore, not easy to articulate and/or communicate.

*Social aspects* – The interplay of social dynamics between coworkers introduce additional considerations that influence disclosure decisions. *Trust* is the primary factor that comes to mind for many people; this may apply to the perception of trust in coworkers, or in the entity or individual requesting the information, *if they reside outside of the circle of colleagues*<sup>3</sup>. Other factors include attitude (whether they like or dislike the other party), and the level of cooperation (denoting views of interaction formed from joint efforts) or competition (e.g., for promotion).

*Self-promotion* – When describing their work to others, people often make their work methods seem more complicated than necessary [12]. These actions may be motivated by one's need to ensure that others believe that the person works hard, and that their work is critical to the organization. In effect, this becomes a way of conveying the importance of oneself and one's work. This can also be linked to the job security factor described above. Such deliberate misrepresentation also contributes to the gap between actual work practices and their provided description.

*Concealing misconduct* – Deliberate misrepresentation may also arise from negative motivating factors e.g. people describing work practices to match the work expected of them. This includes methods that conflict with what the employee has previously communicated they do, are contrary to business policies, or that try to save time and/or effort in non-permissible ways. Therefore, such descriptions represent what *should* be done rather than what *is* done.

## 4 Discussion

These findings do not suggest that employees are against information sharing entirely, rather they point out that public disclosures of private work practices may be neither complete nor fully accurate, especially when such disclosure is impacted by one or more of the aspects described above in Section 3.2.

For process identification and documentation efforts, information about how people carry out the operations must be understood and employees are often encouraged to share a level of detail of their working practices that they typically would not divulge in normal day-to-day work. This raises interesting challenges and questions regarding the granularity at which process information should be and *could be* collected for the purposes of operations modeling.

Also, incomplete, or less-than-accurate descriptions of work practices is an issue that can be at an individual, group, or organizational level. Although the motivations described above were prompted through study of a number of individuals, many of the factors could also apply in disclosure from one work group or organization to another. For example, a human resources department may have sensitive information such as employee personal data, or strategies for compensation, employment, or termination that they are unwilling to share with people outside the department. This “data security” issue is in addition to the motivating factors described in the Section 3.2 and may arise from concern of sharing proprietary or highly sensitive information.

---

<sup>3</sup> Organizations interested in utilizing BPM methods may engage personnel who have specialized expertise to lead the effort, and the personnel may be external to the company as a whole or to the originating organization.

#### 4.1 Applicability to Global Enterprises

Although the gap between private work practices and public disclosure was noted through the study of a co-located team, we believe that similar issues can be expected to exist in globally-dispersed teams. In fact, in such distributed teams, these issues are likely to be felt even more acutely because of the lack of face-to-face social interaction. The social cues available through face-to-face interaction, such as looking people in the eye and shaking their hands, have been stated as a pre-requisite to establishing working-level trust in the business world [13]. And trust, in turn, has been reported as a key factor for disclosure and knowledge sharing in the workplace [14]. It has also been reported that attitudes toward privacy and willingness to share information vary significantly across cultures [8] which could further impact these considerations in global teams since they are invariably comprised of members from disparate cultures. Additionally, organizations often impose country-specific export regulations that limit the sharing of certain types of information outside of national borders. Global enterprises must be aware of these regulations and provide applicable mechanisms to ensure compliance. Thus, we believe that extension of our findings to the global collaborative enterprise could hold great importance.

#### 4.2 Implication for Business Process Management

Section 3.2 speaks to the complex interplay of individual, social, and organizational factors that contribute to the reluctance in disclosing one's work practices. One of the main implications for BPM is the need to embrace the inherent inaccuracies, incompleteness and ambiguities in the disclosure of work practices rather than attempting the impossible venture of eliminating them. How to achieve this in practice without losing the benefits of process modeling is a critical challenge for future research. We offer some preliminary thoughts on ways to address these issues:

- At the launch of a BPM effort, aim to identify people's willingness to share details related to their work methods. This may help the organizers gauge which of the methods specified below may be useful in the work-practice elicitation effort. This may also help identify additional methods (not stated here), or a combination of methods, in order to address any hesitancy in disclosure.
- Allow mechanisms for specifying and distinguishing between public and private information. For example, utilizing encryption to protect private data or enabling specialized views through access controls. An example of this is seen in [15], which describes a system designed for use by multiple collaborating enterprises that partitions global data (information needed and used by all participants) and local data (information needed only by the employees within a specific company).
- Encourage *organizations* to develop policies identifying the types of information that are publicly owned, and the information and methods that can be retained privately.
- Encourage *employees* to share useful and efficiency-improving private work practices in exchange for an incentive or positive reinforcement. For instance, public verbal accolades, awards, compensation, and promotions can

be offered as a way to encourage employees, who have developed or identified methods to accomplish work more efficiently/effectively, to share these methods with other employees.

- Offer information systems that facilitate the sharing of individual work practices. Information sharing in the workplace has been noted as a foundation for collaboration [8] and to reduce duplication of effort [9]. In our observations, we also noted that work efforts were sometimes duplicated unbeknownst to the parties involved due to lack of communication and sharing regarding specifics of individual work practices.

## 5 Conclusions and Future Work

Based on field work conducted to study the work practices of a small town government organization, we identified the disconnect between how individual work practices are carried out as opposed to how they are described to others. Interviews and observations from the field work as well as prior literature points to a host of interrelated factors that contribute to this gap. Since BPM relies heavily on the accuracy of description of the processes being modeled, these insights point to important implications for how such models are constructed and utilized. In particular, we suggest that BPM techniques ought to treat such disconnect as inherent and unavoidable. This presents fruitful avenues for future BPM research on how models can be reconciled with the practices *on the ground*. We are currently extending this work to provide statistical data drawn from a larger population (employees from additional government organizations and corporate enterprises) to further support the factors mentioned above.

## References

1. van der Aalst, W.: Business Process Management: A Personal View. *J. Bus. Process Management* 10(2), 248–253 (2004)
2. Ackerman, M.S., Cranor, L.F., Reagle, J.: Privacy in e-commerce: Examining User Scenarios and Privacy Preferences. In: *Proceedings of the ACM Conference on Electronic Commerce*, pp. 1–8. ACM Press, New York (1999)
3. Bellotti, V., Sellen, A.: Design for Privacy in Ubiquitous Computing Environments. In: *Proceedings of ECSCW*, pp. 77–92. Kluwer, Dordrecht (1993)
4. Gross, R., Acquisti, A.: Information Revelation and Privacy in Online Social Networks. In: *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pp. 71–80. ACM Press, New York (2005)
5. Ackerman, M.S.: The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *J. HCI* 15, 179–204 (2000)
6. Miller, S., Weckert, J.: Privacy, the Workplace and the Internet. *J. Bus. Ethics* 28(3), 255–265 (2004)
7. Bellotti, V.: What You Don't Know Can Hurt You: Privacy in Collaborative Computing. In: *Proceedings of HCI and People and Computers XI*, pp. 241–261. Springer, London (1996)
8. Olson, J.S., Grudin, J., Horvitz, E.: Technical Report Number: MSR-TR-2004-138, Microsoft Research (2004)



9. Olson, J.S., Grudin, J., Horvitz, E.: Towards Understanding Preferences for Sharing and Privacy. In: CHI 2005 Extended Abstracts on Human Factors in Computing Systems, pp. 1985–1988. ACM Press, New York (2005)
10. Borkan, J.M., Miller, W.L., Neher, J.O., Cushman, R., Crabtree, B.F.: Evaluating Family Practice Residencies: A New Method for Qualitative Assessment. *J. Fam. Med.* 29(9), 640–647 (1997)
11. Wight, D.: Boys Thoughts and Talk about Sex in a Working Class Locality of Glasgow. *Soc. Review* 42(4), 703–737 (1994)
12. Schmidt, K.: The Critical Role of Workplace Studies in CSCW. In: Heath, Hindmarsh, Luff (eds.) *Workplace Studies: Rediscovering Work Practice and Informing Design*. Cambridge University Press, Cambridge (2000)
13. Cassell, J., Bickmore, T.: External Manifestations of Trustworthiness in the Interface. *Communications of the ACM* 43(12), 50–56 (2000)
14. Panteli, N., Sockalingam, S.: Trust and Conflict within Virtual Inter-organizational Alliances: A Framework for Facilitating Knowledge Sharing. *Decision Support Systems* 39(4), 599–617 (2005)
15. Adam, O., Hofer, A., Zang, S., Hammer, C., Jerrentrup, M., Leinenbach, S.: A Collaboration Framework for Cross-Enterprise Business Process Management. In: *Preproceedings of the 1st INTEROP-ESA 2005*, pp. 499–510 (2005)

# Analysing Dependencies in Service Compositions

Matthias Winkler<sup>1</sup>, Thomas Springer<sup>2</sup>, Edmundo David Trigos<sup>1</sup>,  
and Alexander Schill<sup>2</sup>

<sup>1</sup> SAP Research Center Dresden, SAP AG, Chemnitzer Str. 48,  
01187 Dresden, Germany

{matthias.winkler, edmundo.david.trigos}@sap.com

<sup>2</sup> TU Dresden, Faculty of Computer Science,

Institute for Systems Architecture, Computer Networks Group,

Nöthnitzer Str. 46, 01187 Dresden, Germany

{thomas.springer, alexander.schill}@tu-dresden.de

**Abstract.** In the vision of the Internet of Services (IoS) services are offered and sold as tradable goods on an open marketplace. Services are usually consumed as part of service compositions defining complex business processes. In a service composition the execution of one service depends on other services. Thus, changes or problems during the provisioning of services may affect other services. While information about dependencies is necessary to handle problems and changes in service compositions, this information is usually only implicitly available in the process description and SLAs. In this paper, we propose an approach where the dependencies between services in a composition are analysed at design time and captured in a dependency model. This information is used to validate the negotiated SLAs to ensure that proper collaboration between the services is possible. At runtime this model can then be applied for determining the effects of events such as service failure (SLA is violated) or SLA renegotiation on other services. Our major contributions are a classification of service dependencies in business processes and an algorithm for semi-automatic dependency model creation based on a process description and the related SLAs. We have evaluated our approach based on a logistics scenario.

## 1 Introduction

The IoS vision aims at creating an open marketplace where services can be offered and sold as tradable goods. Services may be provided fully automatic or involve human and machine tasks. They may be composed to complex business processes and resold as composite services. The involved services cooperate to achieve a common goal. The cooperation is regulated by service level agreements (SLA) negotiated between the service provider and the service consumer. A SLA consists of a set of *service level objectives* (SLO). A SLO refers to a key performance indicator (KPI) and specifies a value or value range and unit together with an operator to express the expected value of a KPI. Example SLOs are *max.temperature*  $\leq 15^\circ$  and *availability*  $\geq 95\%$ .

Due to the collaboration the execution of one service in a composition has dependencies on other services. SLAs need to be negotiated in such a way that the different SLOs of one SLA do not conflict with the SLOs of another SLA. The violation or renegotiation of one SLO by a service can also influence the execution of other services in

the composition or the whole composite service. Thus, information about dependencies is necessary to support the handling of problems and changes in business processes at design time and runtime. However, dependency information is only available implicitly in the process description and SLAs.

In this paper, we propose an approach for analysing dependencies between services in a composition in a semi-automatic manner at design time and capturing them in a dependency model. This model can then be applied for validating negotiated SLAs and determining the effects of events such as service failure or SLA renegotiation on other services.

The paper is organised as follows: In section 2 we introduce the background of our work and a logistics scenario. From the scenario we identify a set of dependency types. Related work is then discussed in section 3, focussing on the detection and modelling of dependencies. As the first major contribution of our paper, we analyse the identified dependency types and extract their characteristics. These characteristics are then exploited in our approach for detecting and modelling dependencies which is presented in section 5 as the second major contribution. Our evaluation is based on a case study using a logistics scenario, the results are described in section 6. We conclude the paper with a summary and an outlook to future work.

## 2 Logistics Use Case

In the IoS vision services from various domains (e.g. logistics) can be traded via service marketplaces. In the logistics domain not only software services are performed. For example, the service of transporting a good is provided by a truck. The cooperation of several logistic companies to deliver goods can be modelled as business process and the contractual details are negotiated as SLAs. In the following we present a scenario from the logistics domain and derive relevant dependencies.

In our scenario Trans Germany Logistics (TGL) acts as organizer of a logistics process offered as composite service Transport Germany. The service is bought by Dresden Spare Parts (DSP), which needs to ship spare parts for cars from Dresden to Hamburg. During the pre-carriage phase goods are picked up by Truck DD (pallets P1 and P2) and AutoTrans DD (pallet P3 and P4) and are transported to a warehouse where they are loaded to a large truck for further shipment to Hamburg. Three different logistics providers are responsible for picking up goods from Warehouse HH and delivering goods to the customer. The process is illustrated in Fig. 1. During the provisioning of this composite service TGL manages the underlying process.

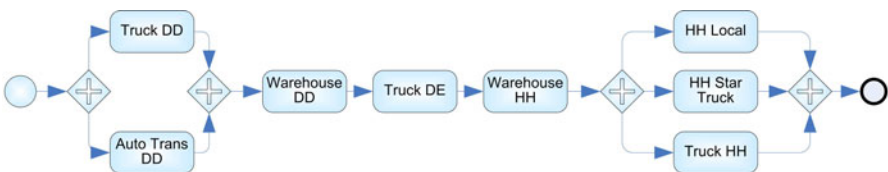


Fig. 1. Collective Goods Transport Scenario

In this use case a number of different dependencies occur. Resource dependencies occur when two services handle the same goods at different points in time. This is for example the case for Truck DD and Warehouse DD with regard to pallets P1 and P2. Time dependencies occur between two interacting services when their interaction is bound to a certain time. The time for making goods available for pickup (Warehouse DD) and the pickup time (Truck DE) depend on each other. Location dependencies occur, similar to time dependencies, when two services interact at a certain location. Price dependencies occur between the composite service and its contained atomic services. The price of the composition is calculated from the price of the atomic services.

These aspects are regulated by SLAs. It is the responsibility of composite service providers to ensure that the SLAs negotiated with the different providers enable the smooth execution of the composition. If the location for delivering goods negotiated with the composite service consumer does not match the delivery location of the atomic services, problems occur. Requests for changes (e.g. renegotiating the number of goods to be transported by the composite service) or problems (e.g. delivery time cannot be met) during provisioning of one service need to be managed due to their effects on other services. If e.g. DSP changes its order and wants to ship fewer goods, TGL needs to adapt the SLAs with its subservices. To automate this process, information regarding the different dependencies needs to be captured. The different types of dependencies need to be considered.

### 3 Related Work

Numerous research projects are focusing on the creation of dependency models including the modelling as well as the discovery of dependencies. Dependency information is used to describe service interactions (e.g. [1]), to optimize sequencing constraints in service compositions (e.g. [2,3]), to do root cause analysis in the case of service failure (e.g. [4,5]), and to manage composite service level agreements [6]. Depending on the application domain and the purpose of the dependency information, the approaches to creating dependency models vary greatly.

Wu et al. [2] present an approach for modelling and optimizing the synchronization dependencies of activities in business processes. A synchronization model, which contains dependency information, is used to support activity scheduling in business processes. Dependency information is modelled manually. The automatic discovery of dependencies is not supported.

The MoDe4SLA [5] approach supports the handling of response time and price dependencies between composite and atomic services. The goal of the system is to support root-cause analysis. The dependency models for each type of dependency are created by a modelling approach. The discovery of dependencies is not supported.

In [4] the authors present an approach to determine the effects of service failure on other services in a composition as well as for the analysis of the root-cause of problems. The discovery of service dependencies is based on message logs. The approach assumes the execution of services as base for the dependency analysis. This is not feasible for our approach, since all dependencies need to be captured prior to service execution.

In [3] the authors use dependency information to support the scheduling of activities in business processes. They discuss the discovery of control and data dependencies

from semantically annotated business activities by evaluating their pre-conditions and effects as well as input and output parameters. This approach differs from our approach in several ways. While our resource dependencies are similar to the data dependencies of their work, we also support dependencies regarding time, location, QoS, and pricing information. Furthermore, their approach is limited to dependencies between atomic services (horizontal dependencies) while our work also supports dependencies between atomic and composite services. Another difference is that Zhou et al. use semantically annotated business activities as a base for their discovery while our work focuses on BPMN process descriptions and SLAs negotiated between services.

The COSMA approach [6] supports service providers to manage composite service SLAs. All information about the different SLAs of the composite service as well as constraints (dependencies) regarding different SLA elements are expressed within a COSMADoc document. For example, composite QoS values are calculated based on aggregation formulas from atomic service QoS values. While the aggregation formulas for the different QoS values are automatically derived from the process description, further constraints need to be added manually or from configuration files. In contrast to our work, COSMA focuses on the relationship between composite services and atomic services. Dependencies between different atomic services are not handled. Dependency types such as resource, location, and time are not covered. However, our approach to QoS and price dependency discovery is based on COSMA.

## 4 The Nature of Service Dependencies

Service dependencies are the underlying cause for the fact that effects of events regarding single services are not limited to these services but influence other services in service compositions. In this chapter a definition of service dependencies is provided. Different types of service dependencies are classified with regard to their occurrence in service compositions and the underlying cause of these dependencies.

### 4.1 Defining Service Dependencies

A general definition of the word *dependence* can be found in [7]. It is described as “the state of relying on or being controlled by someone or something else”. Dependencies are an important aspect in many different scientific disciplines such as economics, organization theory, and computer science [8]. The following definition illustrates dependencies in the light of services and forms the base for our work. *A service dependency is a directed relation between services. It is expressed as a 1:n relationship where one service (dependant) depends on one or multiple services (antecedent). A service S1 is dependent on a service S2 if the provisioning of service S1 is conditional to the provisioning of service S2, i.e. if a property of service S1 is affected by a property of S2.*

### 4.2 Classifying Service Dependencies

In composite services dependencies occur regarding different service properties. Examples are the resources handled by services, different quality of service attributes (e.g.

max. temperature during transport), start and end times of service execution, as well as the location of service provisioning. Dependencies that occur regarding these different service properties can be classified with respect to their occurrence either between the individual services within the composition or with the service composition as a whole. We call dependencies that occur between the individual services of a service composition *horizontal dependencies* because they affect services on the same hierarchical level of composition. Dependencies can also have direct effects on the overall composition. We call these dependencies, which occur across a hierarchical level of composition, *vertical dependencies*.

Furthermore, dependencies can be classified according to the underlying cause of the dependency. In their study on coordination Malone and Crowston distinguish different classes of dependencies which occur in different disciplines (e.g. economics, computer science) [8]. *Task-subtask* dependencies occur when a process of several subservices is created as a refinement of another (composite) service. The goal of a composite service is broken down into sub-goals which are achieved by other services. Task-subtask dependencies cause vertical dependencies between the composite service and its atomic services. Two services have a *producer-consumer* relationship (dependency) when the outcome of one service is required by another service. Requirements of a service include its input resources as well as its pre-conditions. Producer-consumer relationships occur as horizontal dependencies. The execution of services may underlie certain temporal constraints between services. These constraints are called *simultaneity constraints*. Two services may need to occur at the same or at a different time. Simultaneity constraints occur as horizontal dependencies. They are often a result of a producer-consumer relationship but may also exist e.g. due to preferences of the process developer. In Table 1 an overview of the different dependency types and their classification with regard to horizontal and vertical dependencies is provided.

**Quality of service:** The QoS attributes of services (e.g. maximum temperature during a transportation process) describe quality aspects of service provisioning. QoS dependencies occur due to task-subtask dependencies. Thus, within a service composition QoS dependencies occur usually as vertical dependencies. Violations or changes to the QoS values of atomic services affect the composite service. Changing the composite QoS values requires a modification of all the respective atomic QoS values.

**Table 1.** Classification of service dependencies

Dependency class	Attribute	Horizontal	Vertical
Task-subtask	QoS	-	X
	Price	-	X
	Resource	-	X
	Start/end time	-	X
	Start/end location	-	X
Producer-consumer	Resource	X	-
	Start/end location	X	-
Simultaneity constraints	Start/end time	X	-

**Price information:** A price dependency exists when the price of a service depends on the price of other services. For example, the price of a composite service is calculated based on the prices of its atomic services or the prices of the atomic services are negotiated based on breaking down the price of the composite service. Price dependencies occur due to task-subtask dependencies and, thus, only occur as vertical dependencies. Renegotiating the price of an atomic service only affects the composite service.

**Time information:** Time dependencies express temporal relationships regarding the execution of services. They can occur as task-subtask relationships (vertical dependency) or simultaneity constraints (horizontal dependency). If the consumer of the composite service renegotiates the time when goods should be picked up, this affects the atomic services in the composition.

**Location:** Location dependencies occur between two services that need to be executed at the same or at a different location. Location dependencies occur as a result of task-subtask or producer-consumer relationships. Thus, there are horizontal and vertical location dependencies. If e.g. the consumer of the composition decides to renegotiate the location of delivery, this affects the atomic providers of the composition.

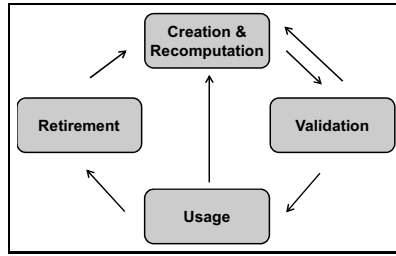
**Resource:** Two services have a resource dependency, when the availability of a resource, which is needed by one service, depends on another service. Resources include electronic data as well as material goods. Resource dependencies occur due to provider-consumer relationships as well as task-subtask dependencies. Thus, they occur as vertical or as horizontal dependencies. If the goods being transported are damaged, the SLAs for all services, which should have handled the same resource later in the process, should be adapted, because further handling is not possible. This is an example of a horizontal provider-consumer based dependency. An important aspect is that resource dependencies have a transitive nature. Thus, the failure of one service may not only affect the direct consumer but all other services which handle the same resource.

## 5 Dependency Model Creation

A dependency model is an explicit representation of dependency information between services. It was specified via a metamodel, which describes services and the different types of dependencies between these services. Services can play the role of a *dependant* or *antecedent*. Each dependency is described in more detail depending on its type. More details about the dependency model can be found in [9].

### 5.1 Lifecycle for Managing Dependencies

In order to manage the service dependencies in service compositions, a lifecycle consisting of four phases was developed (see Fig. 2). During the *Creation and Recomputation* phase the dependency model is created based on the process structure and SLA information. Information can be added or changed if conflicts are detected, SLAs change, or the process description is adapted. The *Validation* phase is necessary to ensure that the created dependency model is valid. It is also necessary to validate the negotiated SLAs to ensure that they enable proper collaboration between the different services. In the case



**Fig. 2.** Dependency Model Lifecycle

of problems either the dependency model or the SLAs need to be adapted. During the *Usage* phase the dependency model supports runtime dependency evaluation tasks such as the determination of SLO violation effects or handling SLA renegotiation requests. In the case of renegotiation the model may need to be adapted accordingly. During the *Retirement* phase the dependency model is discarded.

This paper is mainly concerned with the creation phase. The creation process of a dependency model was realized as a semi-automatic approach consisting of automatic dependency discovery and the explicit modelling of dependencies. This was necessary because it is not possible to automatically discover all dependencies based on the process and SLA descriptions. For example, it is not possible to discover time dependencies between services in parallel paths of the process or location dependencies, because there is not enough information available to discover these dependencies. Thus, it is necessary to enable the manual modelling of dependencies in addition to automatic dependency discovery. In this section the approaches to automatic dependency discovery as well as dependency modelling are presented.

## 5.2 Dependency Discovery

The automatic discovery of dependencies consists of two major steps, the determination of services relevant for analysis and the execution of the analysis. In order to determine the relevant services regarding which the dependency analysis is executed, all linear paths leading from the start node to the end node of a process are determined. All services, which are connected via a valid path in the process, may have horizontal dependencies. Creating all possible subpaths of a process facilitates the analysis of dependencies. For the discovery of vertical dependencies, the composite service is analysed with regard to different atomic services. Time dependencies exist between the composite service and each first or last atomic service within a path. Resource dependencies may exist between the composite service and any atomic service in a path.

As a second step the different services are analysed for dependencies. The approach for analysis of dependencies is specific for each type of dependency. Time dependencies are discovered based on the composite service process structure. Resource dependencies evaluate the SLAs of the different services. An excerpt from a simplified sample SLA is shown in listing 1 in pseudo-code notation. An example of the analysis is presented in the validation section.



```

service {
  serviceName Transport Germany
  description Transport of goods within Germany
  ..
  startLocation Teegasse 83, Dresden , Germany
  startTime 2009-02-10T10:00:00Z
  required resources P1, P2, P3, P4
  ..
  endLocation Im Tal 1, Hamburg, Germany
  endTime 2009-02-10T12:00:00Z
  provided resources P1, P2, P3, P4
  ..
  maxTemperature 35°C
}

```

Listing 1. Sample SLA

**Horizontal time dependencies:** Two atomic services ( $s1_{atom}, s2_{atom}$ ) which are directly connected via an edge in a process have a *finish-to-start time* dependency:  $s1_{atom}.endTime$  finish-to-start  $s2_{atom}.startTime$ .

**Horizontal resource dependencies:** Two atomic services ( $s1_{atom}, s2_{atom}$ ), which are directly or indirectly connected in a process, have a *resource* dependency, if two conditions are met: (1) a subset of the output resources of  $s1_{atom}$  matches a subset of the input resources of  $s2_{atom}$  and (2) the matching resources are not provided to  $s2_{atom}$  by another service  $s3_{atom}$ , where  $s3_{atom}$  occurs between  $s1_{atom}$  and  $s2_{atom}$  in this path.  $s2_{atom}.inputResource$  resourceDependent  $s1_{atom}.outputResource$ .

**Vertical time dependencies:** Each first atomic service in the process has a *start-to-start* time dependency on the composite service:  $s_{comp}.startTime$  start-to-start  $s_{atom}.startTime$ . Each last atomic service has a *finish-to-finish* time dependency on the composite service:  $s_{atom}.endTime$  finish-to-finish  $s_{comp}.endTime$ .

**Vertical resource dependencies:** An atomic service has a resource dependency on the composite service regarding its input resources if (1) a subset of the composite service input resources matches a subset of the atomic service input resources and (2) the atomic service does not have a horizontal resource dependency regarding the matching resources:  $s_{atom}.inputResource$  resourceDependent  $s_{comp}.inputResource$ .

The composite service has a resource dependency on an atomic service regarding its output resources if (1) a subset of the composite service output resources matches a subset of the atomic service output resources and (2) if the matching resources are not provided by another atomic service occurring after  $s_{atom}$  in the same path:  $s_{comp}.outputResource$  resourceDependent  $s_{atom}.outputResource$ .

The discovery of price and QoS dependencies is based on the work of Ludwig and Franczyk [6]. A formula for calculating the respective composite service attribute value from the atomic service values is created.

**Vertical price dependencies:** A composite service  $s_{comp}$  has a *price dependency* on all atomic services  $s1_{atom}..sn_{atom}$  which have a price.  $s_{comp}.price$  priceDependency  $s1_{atom}.price..sn_{atom}.price$ .

**Vertical QoS dependencies:** A composite service  $s_{comp}$  has a *QoS dependency* on all atomic services  $s_{1_{atom}}..s_{n_{atom}}$  which have the same QoS property:  $s_{comp}.maxTemp$  qosDependency  $s_{1_{atom}}.maxTemp..s_{n_{atom}}.maxTemp$ .

### 5.3 Modelling of Service Dependencies

The dependency discovery algorithm produces a valid dependency model. However, there are dependencies between services which cannot be discovered. The relevant information cannot be gained from the process description or the negotiated SLAs but is available as knowledge of domain experts only. To support the handling of these dependencies, a dependency model editor enables the manual adaptation of the generated dependency model. Optional modelling steps cover complex time dependencies e.g. between services in parallel paths, location dependencies (e.g. equals, not\_equals), and dependencies for QoS attributes where no automatic detection is supported, or the user wants to model a calculation formula.

## 6 Evaluation

As part of the validation the approach for creating dependency models was implemented. It was integrated into the Eclipse-based service engineering workbench called ISE, which was developed within the TEXO project. The automatic dependency discovery as well as the dependency editor were implemented as Eclipse plug-ins. In the current implementation time and resource dependencies are discovered. Location, price, and QoS attributes can be modelled. The implementation of QoS and price dependency discovery is not planned, since its functioning has been demonstrated [6].

Furthermore, we demonstrate the dependency model creation approach based on the logistics scenario. Due to space limitations only few samples of horizontal and vertical time dependencies are presented.

As a first step of the discovery, the composite service process is decomposed into six paths leading from the start node to the end node. Table 2 shows two of the six paths.

**Table 2.** Linear process paths

Truck DD - Warehouse DD - Truck DE - Warehouse HH - Truck HH
Truck DD - Warehouse DD - Truck DE - Warehouse HH - HH Star Truck

As a second step these paths are analysed regarding the different dependencies. Horizontal time dependencies of type finish-to-start are created between all pairs of services which are directly connected in the process. From the first path these pairs are e.g. Truck DD and Warehouse DD as well as Warehouse DD and Truck DE. Horizontal resource dependencies are discovered based on these paths as well. For example Warehouse DD depends on AutoTrans DD regarding pallets P3 and P4. Two examples of horizontal dependencies are depicted in table 3.

In order to analyse the vertical dependencies the composite service Transport Germany is compared with the different atomic services in the process paths. Time dependencies of type start-to-start are created between Transport Germany and the first services in the process, i.e. Truck DD and AutoTrans DD. Finish-to-finish time dependencies are created between Transport Germany and the last services in the process, i.e. Truck HH, HH Star Truck, and HH Local. Vertical resource dependencies are also discovered based on the different paths. However, their analysis is not limited to the services directly connected to the start and end node but continues until all input and output resources are matched according to our specification. An example of a vertical resource dependency is shown in table 3. AutoTrans DD depends on Transport Germany regarding its input resources (P3, P4).

In addition to the dependency discovery users can then model e.g. location dependencies or further time constraints, e.g. specifying that the two trucks in Dresden are required to pick up the goods at the warehouse at the same time. In table 3 we show a location dependency example which requires that the end location of Truck DD is the same as the start location of Warehouse DD.

**Table 3.** Dependencies of logistics process

Antecedent - Dependant	Dependency	Description
Truck DD - Warehouse DD	time	endTime <i>finish-to-start</i> startTime
AutoTrans DD - Warehouse DD	resource	P3, P4
Transport Germany - Truck DD	time	startTime <i>start-to-start</i> startTime
Transport Germany - AutoTrans DD	resource	P3, P4
Transport Germany - all atomic services	QoS	max( TruckDD.temperature, WarehouseDD.temperature,... )
Truck DD - Warehouse DD	location	endLocation <i>equals</i> startLocation

Based on this example it becomes clear that even in a relatively small use-case there are many dependencies. The semi-automatic dependency model creation process facilitates the process of capturing these dependencies.

## 7 Conclusion and Outlook

The explicit availability of information about dependencies between services in a service composition is important for the handling of changes requested by customers or violations of service level objectives. As illustrated in the paper, dependencies can occur vertically between a composite service and its atomic services as well as horizontally between different atomic services. Moreover, dependencies can be of different types, e.g. related to time, resources, location, price and quality of service attributes.

We have shown that these dependencies have different characteristics. Exploiting the knowledge about the characteristics of different types of dependencies we introduced a set of algorithms for discovering the dependency types relevant in logistics scenarios. The application of these algorithms is demonstrated in the evaluation section for our logistic use case.

In future work we will extend our evaluation to more complex use cases involving multiple levels of composition. Moreover, we plan to analyse the influence of process loops on the discovery of dependencies and to verify SLAs with respect to the negotiated times, locations, resources, QoS, and price based on the dependency model.

## Acknowledgements

The information in this document is proprietary to the following Theseus Texo consortium members: SAP AG and Technische Universität Dresden. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2009 by the Theseus Texo consortium.

## References

1. Zhou, J., Pakkala, D., Perala, J., Niemelä, E.: Dependency-aware service oriented architecture and service composition. In: Proceedings of IEEE International Conference on Web Services, ICWS 2007 (2007)
2. Wu, Q., Pu, C., Sahai, A., Barga, R.: Categorization and optimization of synchronization dependencies in business processes. In: Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE 2007), pp. 306–315 (2007)
3. Zhou, Z., Bhiri, S., Hauswirth, M.: Control and Data Dependencies in Business Processes Based on Semantic Business Activities. In: Proceedings of iiWAS 2008 (2008)
4. Basu, S., Casati, F., Daniel, F.: Toward Web Service Dependency Discovery for SOA Management. In: SCC 2008: Proceedings of the 2008 IEEE International Conference on Services Computing, Washington, DC, USA, pp. 422–429. IEEE Computer Society, Los Alamitos (2008)
5. Bodestaff, L., Wombacher, A., Reichert, M., Jaeger, M.C.: Monitoring Dependencies for SLAs: The MoDe4SLA Approach. In: IEEE SCC, vol. (1), pp. 21–29. IEEE Computer Society, Los Alamitos (2008)
6. Ludwig, A., Franczyk, B.: Cosma—an approach for managing slas in composite services. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 626–632. Springer, Heidelberg (2008)
7. Miller, G.A., Fellbaum, C., Teng, R., Wakefield, P., Langone, H.: Wordnet. WordNet Online Database (2009) (Online: accessed 21-July-2009)
8. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. *ACM Computing Surveys* 26, 87–119 (1994)
9. Sell, C., Winkler, M., Springer, T., Schill, A.: Two dependency modeling approaches for business process adaptation. In: Karagiannis, D., Jin, Z. (eds.) KSEM 2009. LNCS, vol. 5914, pp. 418–429. Springer, Heidelberg (2009)

# Open Service-Oriented Computing for Logistics: A Case in Courier, Express and Parcel Networks

Marcel Kunkel<sup>1</sup>, Christian Doppstadt<sup>2</sup>, and Michael Schwind<sup>3</sup>

<sup>1</sup> `mk@pickpoint.de`

<sup>2</sup> Business Information Systems and Operations Research, Technical University  
Kaiserslautern, Gottlieb-Daimler-Strasse 47, 67653 Kaiserslautern, Germany

`doppstadt@wiwi.uni-kl.de`

<sup>3</sup> IT-based Logistics, Goethe University, Grüneburgplatz 1, 60325 Frankfurt,  
Germany

`schwind@wiwi.frankfurt.de`

**Abstract.** Logistics service providers are forced to optimize their operations due to increasing market pressure. Outsourcing and network optimization are measures to improve their market position. Although the underlying operational research problems in logistics are well documented and the business tendency towards outsourcing already manifests itself in the growth of many local multi vendor logistics providers, the connections between both parts into an innovative and integrated logistics concept is not yet established. Our OPEN Service-Oriented Computing for LOGistics (OPEN SOC LOG) approach proposes a system which integrates state-of-the-art optimization tools with a service-oriented computing concept in order to provide a spectrum of standardized and combinable services for logistics companies ranging from tour and network optimization to contractual and accounting functionalities.

## 1 Introduction

There is high pressure to improve the efficiency of logistics services as a result of rising energy cost and fierce competition among carriers. Additionally, society asks for higher standards of environmentally good transportation practices [1]. This is also true for courier, express, and parcel (CEP) service providers that have to fulfill in-time delivery under multiple constraints. In Germany the CEP market has grown up to 2.3 billion shipments per year with an average revenue of 6.23€ per shipment in the last 20 years. Besides the three worldwide logistics service providers, the CEP market consists of several multi national companies (TNT, GLS, Trans-o-flex, etc) and a large number of local or regional service providers with special focuses (e.g. courier same day services, express next day, in-night transport etc.). In recent years, political and public pressure to lower CO<sub>2</sub> emissions, and to provide ‘green logistics’ has forced service providers to reduce the kilometers per delivery. This is a major factor for improving environmental and economic performance [2]. One way to improve the efficiency of logistics services is to introduce collaborative optimization. This can be done by

jointly finding optimal routes for the vehicles or sharing transportation capacity. Transportation exchanges are an example of such an instrument. However, the current transportation exchanges are neither capable of directly integrating tour planning and other optimization services for logistics nor of fulfilling the complex requirements of the CEP market [3]. The OPEN Service-Oriented Computing for LOGistics (OPEN SOC LOG) framework addresses those challenges and combines optimization and transport exchange mechanisms to provide a variety of standardized, combinable services for logistics enterprises ranging from tour and network optimization to contractual and accounting functionalities.

## 2 Market for Service-Oriented Computing in Logistics

As a result of the size of the CEP market, the interconnections of the participants and the complex business and transport processes which have to be supported, a variety of software applications have been developed. Until now the only approach to structure the logistics software market has been started by the IML and lead to the current Transport-IT initiative which is essentially a software catalog for transport logistics. Hence, the structure IML uses is more technically oriented rather than market oriented. This paper proposes the following supplier-oriented structure for logistics transport software:

1. Process documentation tools created by CEP networks
2. Analytical optimization approaches from universities and research labs
3. Mobile device development or depot automation from automation companies
4. Map, navigation, and route planning software from geo data providers
5. Logistics market places software used by transportation exchange providers

Important for the current software situation is the history some logistics networks in the CEP market experienced during their development over the last 20 years. A good example is the German DPD which was built as a network of German regional logistics service providers. They decided to work together for nationwide parcel deliveries rather than to hand over national requests to the big three worldwide service providers (DHL, UPS, FEDEX). Their requirement on IT systems which support trans-company shipments both commercially and process wise originated in the early days of their cooperation. They founded a central customer solutions company which controlled and monitored all regional activities including the IT solutions development. The big three historically had their own IT departments and employed their own drivers. Today, the companies outsource their drivers, the IT infrastructure and most of the software used in the depots. Companies like Pitneybowes and Siemens Automation develop software and hardware for mail and parcel sorting systems. In recent years, smaller IT process and logistics specialists have developed special solutions for mobile scanners and communication equipment, often interfacing with a central IT system. Some of them (e.g. Kratzer Automation<sup>1</sup>) also provide a central system

---

<sup>1</sup> [www.kratzer-automation.com](http://www.kratzer-automation.com)

for the communication with the decentralized scanners and act therefore as full service IT outsourcing partners. Companies like the PTV, with a background in map and navigation software development, focus on network and tour planning. University or research labs<sup>2</sup> concentrate more on special intra company optimizations. Additionally, logistics market places arise<sup>3</sup>, driven by the requirement for cooperation. Until now they have been freight marketplaces for bulk goods of full truck load (FTL) and less than truck load (LTL) but do not have a relevance for the CEP market. OPEN SOC LOG will fill the identified gaps in the logistics software market and unify three targets:

1. Inter company processing and commercial operations
2. Intra and inter company real time optimization
3. Interconnection between heterogeneous devices

For the current providers OPEN SOC LOG will be positioned as a supplement to the current solutions and will help to distribute further their own solutions through the network characteristics of OPEN SOC LOG.

### 3 OPEN SOC for Courier, Express and Parcel Networks

There is few literature on service-oriented computing applications for logistics. Talevski et al. [4] present a SOC-based framework for logistics service providers. They emphasize the importance of standardized SOC alliances especially for smaller and medium sized companies in the logistics sector. The introduction of combinable SOC structures is appropriate to foster cooperation and collaboration in the logistics sector. Unfortunately, the work of Talevski et al. [4] mainly focuses on the SOC aspect and does not mention in detail the requirements and types of services that are needed in logistics environments [5]. This is essential for effective and broad application of such a system. Therefore, we introduce the OPEN SOC LOG framework with requirements and services needed in logistics.

#### 3.1 A Pragmatic Definition of OPEN SOC LOG

Web services deliver a wide range of content in a machine-readable way [6, p.3]. In this context the terms service-oriented computing (SOC) and service-oriented architecture (SOA) are often used. SOC is a new paradigm in computer science, that aims at an easier integration of existing application components to construct and implement new applications [7]. This includes web services, but has also a broader scope. Generally, SOC can be seen as an approach to couple already implemented functionality on heterogeneous and distributed systems [8]. SOA is a building block within the concept of SOC. ‘SOA supports service-orientation in the realization of the strategic goals associated with service-oriented computing’ [9]. We base our project on the concepts of the ‘Reference Model for Service Oriented Architecture’<sup>4</sup> [10]. The basic elements of this reference model are:

<sup>2</sup> [www.iam.fraunhofer.de](http://www.iam.fraunhofer.de)

<sup>3</sup> [www.transporeon.com](http://www.transporeon.com)

<sup>4</sup> [www.oasis-open.org](http://www.oasis-open.org)

**Service:** The service is offered by a provider and grants access to a capability for a consumer. For OPEN SOC LOG the service consumer requests information from the service provider, who returns the requested information (e.g. daily route planning).

**Dynamics of services:** The dynamics represents three concepts that are required to interact with the service:

**Real world effect:** The real world effect describes the consequences of the online transactions to the real world. For OPEN SOC LOG, an example would be that after a tour optimization a driver actually drives the previously calculated tour.

**Visibility:** It is required that the service consumer is able to find ('see') the service. Moreover, the provider has to describe the service he offers thus enabling the consumer to find the requested service. As a prerequisite, the service has to be accessible for the operation process. This service registry is based on the idea of a triangular relationship between a *service requester*, a *service provider*, and a *service registry* in the SOA concept. This enables the supplier to register the service it offers and the consumer to find the solution that best fits its needs [11]. In our case, an OPEN SOC LOG service provider would register its service (e.g. route planing, tour planning, Geo coding etc.) at the *service registry* and a CEP service provider (*service requester*) would search for its preferred service (or combination of services) in the *service registry*.

**Interacting with services:** In many cases the interaction with the system is performed by sending messages, but it might also be done by modifying a shared data resource. For OPEN SOC LOG both ways are feasible and no decision regarding that concern has yet been made.

**About services:** In addition to the dynamics of the service within its environment, the 'about service' deals with aspects of the service description and execution. There are no specifics concerning OPEN SOC LOG.

**Execution context:** The basic idea behind execution context is to describe everything related to the execution of the service. This contains information on the infrastructure as well as agreements concerning the execution process. There are no specifics concerning OPEN SOC LOG.

**Policies and contracts:** The regulation of the agreements for the usage of a service and the relationship between provider and requester is summarized under the topic 'policies and contracts'. For OPEN SOC LOG this is an important topic, as the service consumer is charged for its request and in return has a precise expectation of the results and the QoS.

**Service description:** The service description contains all information required to use the service. For the OPEN SOC LOG scenario this description would contain details about the information required for the optimization process and an accurate definition of the results that service is going to deliver. This is especially important for the service broker to find the suitable combination of services via UDDI.

In addition to the above listed concepts, OPEN SOC LOG has to fulfill more requirements that are specific to our application domain:



1. Motivation, market development and standardization
2. Trustworthiness, privacy and security
3. Computational complexity
4. Billing and service level agreements

These specific requirements to service-oriented computing will be discussed in detail in section 4. Nevertheless we can already give a general definition of OPEN SOC LOG:

*OPEN SOC LOG is based on the idea of SOA but extends the design principles to features required to fit the needs of the logistic planning domain. The term ‘open’ reflects the aim of using a standardized way to allow as many customers as possible to make use of this framework.*

For reasons of simplification, we stick to this formal definition approach and leave out further technical details, such as SOAP and underlying XML.

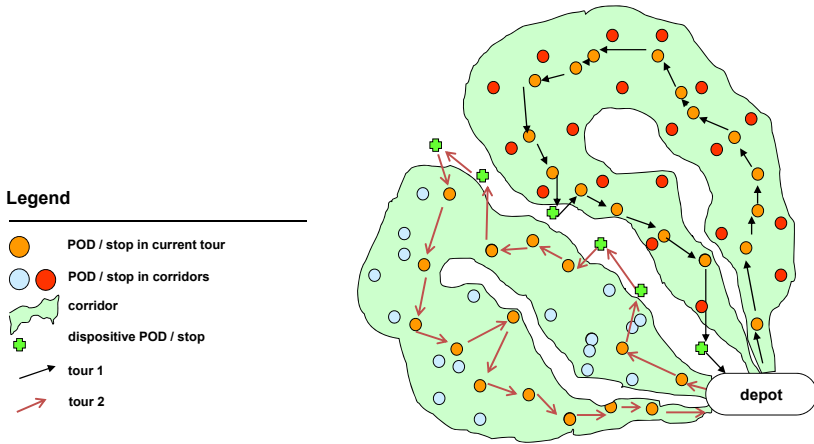
### 3.2 The Case of Courier, Express, and Parcel Networks: KolOptNet

The research project *KolOptNet*<sup>5</sup>, jointly realized by the chair of IT-based logistics of the Goethe University Frankfurt and Pick Point AG, focuses on the optimization of last mile delivery of CEP service providers [12].

The common industrial application is characterized by a periodical calculation of optimal tours (e.g. once a year). In order to improve the efficiency of a CEP delivery network the KolOptNet solution allows a daily optimized and simulation-based tour planning. The optimal tour is determined by solving a vehicle routing problem with time windows. Moreover, we simultaneously consider the effects of driver learning which reduce the delivery times by up to 40%. Therefore, we calculate regular tours for all drivers, which can be adapted every day. The set of potential recipients is aggregated within a delivery corridor.

The cost efficiency of the network is further improved by integrating collaborative aspects. Neighboring subcontractors may exchange dispositive customers who are situated between regular tours of two drivers. Fig. 11 illustrates the allocation of dispositive recipients to regular tours. This collaborative transportation planning is realized by introducing combinatorial auctions, which have proven their usefulness in this context in several previous studies [13]. In addition, we allow flexible delivery corridors in our optimization model instead of the commonly used fixed areas. Such drivers have to adapt stepwise to the new conditions in the flexible area by exploiting learning effects [12]. Finally, we aim at providing an integrated software prototype using service-oriented architecture (SOA) which combines the route planning and the combinatorial auction services. The modularly designed system uses historical as well as the current days data to determine the sorting plans for the packages and the optimal delivery tour. Furthermore, we transfer our output data to the wireless mobile package delivery scanner devices which in turn collect data from the generic delivery process.

<sup>5</sup> [www.koloptnet.de](http://www.koloptnet.de), the project is funded by the IKT2020 initiative of the Federal Ministry of Education and Research (Grant No. 01IS09016B).



**Fig. 1.** Integration of dispositive recipients into regular tours

## 4 Challenges of OPEN SOC LOG

Referring back to our definition in Section 3.1, OPEN SOC LOG is based on the SOA description while extending the design principles to features specific to the logistics planning domain. The term ‘open’ implies the use of standards in order to attract as many costumers as possible. With respect to the economic and technical properties we must also consider privacy and security issues as well as service levels concerns. The reasons are now explained according to section 3.

### 4.1 Motivation, Market Development and Standardization

Although logistics market places have become important for bulk goods logistics, they have not reached a relevant penetration in the CEP market yet. This comes from the fact that single trade values in the CEP market are too small to justify the setup costs unless there is a high degree of automation. The similarity of goods in the CEP industry is much higher than in the bulk goods logistics such that automated trading and order processing could work. But why should a supplier or customer actively participate in a new logistics market place which has not yet shown to have the critical mass to fulfill most of the requests or offers? The motivation could arise from individual optimization results the requester can get out of the OPEN SOC LOG framework. At the beginning, small local network optimizations for local requesters could ensure growth and usage of the platform. After that initial phase the broader usage of individual optimization advantages can be stimulated. This reflects the current situation in the bulk goods market places. Within a norming phase individual service providers encounter supply and demand structures which push market liquidity. Only by defining an open standard, the OPEN SOC LOG framework will be able to collect the critical mass of users that is required to generate considerable economies of network while fostering collaboration in the logistics network [14].

## 4.2 Trustworthiness, Privacy and Security

The OPEN SOC LOG has to overcome the issue of trustworthiness and privacy. With privacy, the participant fears that his personal data will be a common good or at least something which makes him dependent on the platform. This could keep him away from providing his network and parcel information. Trustworthiness addresses a special problem of game theory where the participants can improve their position in the negotiation process through deception. Whereas the privacy issues have to be overcome by technology and contractual design, the trustworthiness issue can be solved by special mechanisms [15]. On the technical side OPEN SOC LOG has to provide the common security mechanisms [7].

## 4.3 Computational Complexity

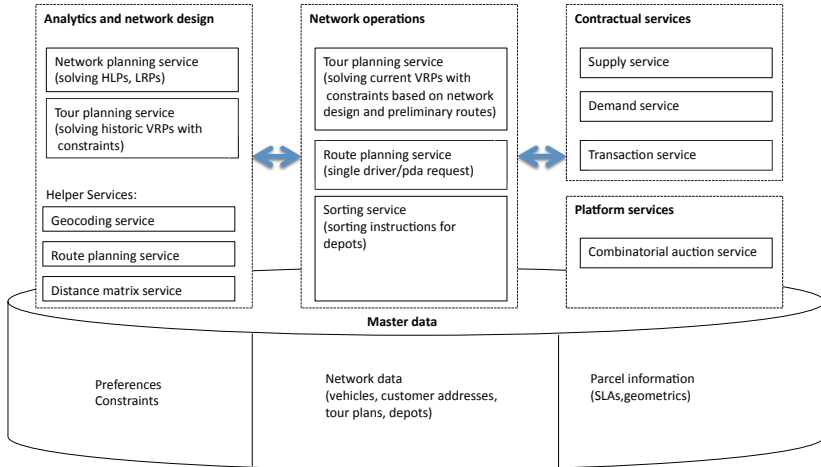
Optimization problems in CEP networks are often NP-hard. For this reason heuristics seem to be an appropriate solution method. Additionally, real-world constraints are often more complex than the ones used in the operations research literature. Complexity reduction is required at the expense of solution quality. OPEN SOC LOG enables the use of clustering methods and supports driver learning [12]. Due to this computational complexity, we have to define service levels for the services providers. Calculating a solution for a routing problem may take a long time if the optimal solution is required, but while the use of heuristics normally significantly reduces calculation time, it also reduces the quality of results. A trade-off between complexity reduction and service quality must be integrated into the billing services.

## 4.4 Billing and Service Level Agreements

In order to provide a powerful instrument for logistics service providers, the platform, with its services, has to ensure that the acceptance and realization of a request can be provided for a reasonable price. Otherwise the cost for backup strategies for the requester will be too high. Additionally, the platform has to ensure proper billing, payment and the creditworthiness of the platform participants. The platform will therefore use standard toolkits like credit check which are provided as web services to overcome that hurdle. In connection with the calculation time/quality trade-off, the OPEN SOC LOG framework should also provide reasonable service level agreements [7,16].

# 5 A Service-Oriented Architecture for Open Logistics

The OPEN LOG SOC services will be implemented in WSDL. Accordingly, the message exchange will be conducted using XML files. The message based communication model is an appropriate structure for exchanging large amounts of data, processing them through the service and returning the results. The results can be trivial or complex, the feedback can be synchronous or asynchronous. As



**Fig. 2.** Groups of services in the OPEN SOC LOG framework

described above a successful service definition must keep the services independent. Another challenge arises from the fact that logistics networks have tight time frames for the physical delivery as well as in the decision or deal making process.

The services in OPEN SOC LOG can be grouped as follows (see Fig. 2):

- Analytics and network design
- Network operations
- Platform and contractual services

The grouping reflects both the introduction of the services to the market and the attractiveness to the service requesters over time. The group ‘analytics and design’ can be used by local or national service providers to optimize their own business without interaction with other logistic service providers. The results of the optimization services can be used as input to ‘network operations’ where daily routes and tour plans can be optimized and used within the operations. The platform and contractual services fulfill two objectives:

- Commercial process support for logistics assignments
- Optimization across suppliers

and can be used in two different ways:

- Use of optimization outcomes to manually create supply and demand on the OPEN SOC LOG platform
- Use of inter organizational optimization capabilities of OPEN SOC LOG where tour plans of all connected service providers will be jointly optimized using tour optimization and combinatorial auction services.

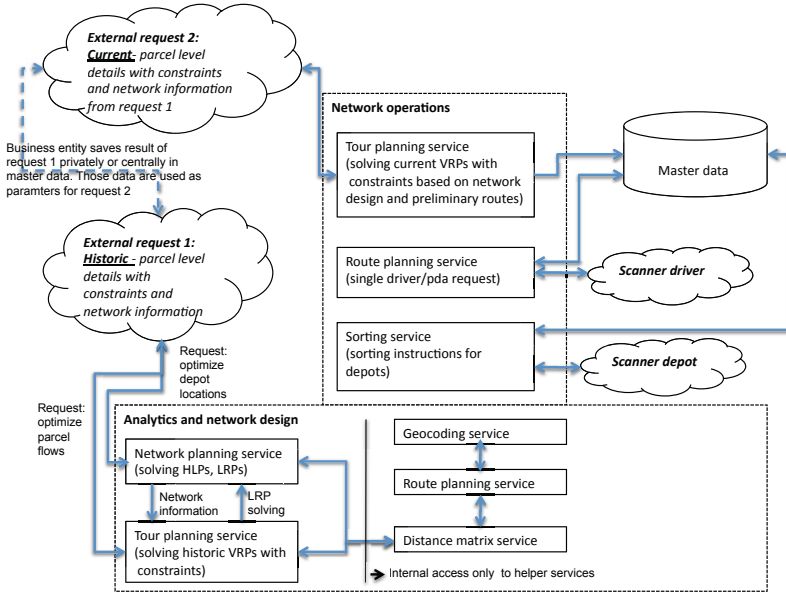


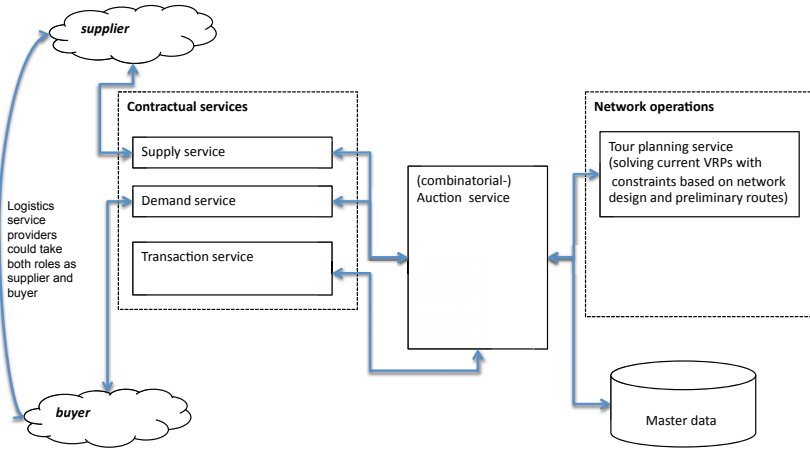
Fig. 3. Organization of technical services in the OPEN SOC LOG framework

### 5.1 Analytics and Network Design

Network planning and tour planning are the core services within analytics and network design. They are accessible externally by service requesters. Those services use the helper services ‘geo coding’, ‘route planning’ and ‘distance matrix’ to fulfill their optimization work properly. The network planning service basically solves depot and hub location problems [17]. Network planning requires parcel flows plus the planning parameters (vehicles, existing structures, etc.) as input and produces optimized locations plus tour plans as output. Route planning realizes a multi constraint VRP solution [18]. The meta heuristics used differentiate from existing solutions by integrating the driver learning aspect. The algorithm uses Ant and Tabu Search [18] optimization elements to produce good solutions within reasonable time.

### 5.2 Network Operations

The analytical and design services deliver optimized network data within the network operation services. The core is tour optimization applied on a day-to-day base. It uses historic tour data from prior operations [19] to optimize the daily business. This enables the OPEN SOC LOG framework to exploit driver learning extensively. The structure and relationship among network operations, analytics and design is shown in Fig. 3. Logistics service providers should not be bothered by interface and mobile scanner software development. Thus, there are



**Fig. 4.** Organization of business services in the OPEN SOC LOG framework

web services for sorting and driver scanners. After requesting sorting or route instructions based on a route or depot ID, the service delivers such data.

### 5.3 Platform and Contractual Services

The platform service basically realizes a logistics marketplace and uses combinatorial auctions for realizing complex buying scenarios where either the buyer or the seller of logistics services are not able to fulfill the logistics requirements alone [14]. As the input for the platform service has to be very accurate in terms of data structures, the supply and demand services help to translate more complex buying or selling offers into a ‘clean’ structure. Fig. 4 shows the input output relations of platform, contractual and network operations services. The combinatorial auction itself uses the route optimization service to evaluate the optimal exchange or trade of route elements.

## 6 Conclusion

The logistics software market has a lack of service-oriented logistics software which combines intra and inter-company processes. Our OPEN SOC LOG framework integrates several combinatorial optimization methods to solve large and rich VRPs while increasing provider’s efficiency and therefore stimulating participation in the inter-company components. The SOA-based framework allows inter company optimizations supporting the relevant business processes especially in CEP logistics. Following the analysis of the current supplier structure in the logistics software market, the open positioning of OPEN SOC LOG will facilitate cooperation with current software companies, allowing them to use that part of the service framework which fits into their own service offering.

## References

1. Leonardi, J., Baumgartner, M.: CO<sub>2</sub> efficiency in road freight transportation: Status quo, measures and potential. *Transportation Res. Part D* 9, 451–464 (2004)
2. Maden, W.: Vehicle routing and scheduling with time varying data: A case study. Technical Report 2009-21, Transport and Logistics Division, University of Huddersfield, Lancaster University Management School, Working Paper (2009)
3. Schwind, M., Gujo, O., Vykoukal, J.: A combinatorial intra-enterprise exchange for logistics services. *Information Systems and E-Business Management* 7, 447–471 (2009)
4. Talevski, A., Chang, E., Dillon, T.S.: Reconfigurable web service integration in the extended logistics enterprise. *IEEE Trans. on Ind. Informatics* 1, 74–84 (2005)
5. Dillon, T.S., Wu, C., Chang, E.: Reference architectural styles for service-oriented computing. In: Li, K., Jesshope, C., Jin, H., Gaudiot, J.-L. (eds.) *NPC 2007*. LNCS, vol. 4672, pp. 543–555. Springer, Heidelberg (2007)
6. Singh, M.P., Huhns, M.N.: *Service-Oriented computing: semantics, processes, agents*. John Wiley & Sons, Ltd., Chichester (2005)
7. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: *Service-oriented computing: State of the art and research challenges*. *Computer* 40(11), 38–45 (2007)
8. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.J.: 05462 service-oriented computing: A research roadmap. In: Cubera, F., Krämer, B.J., Papazoglou, M.P. (eds.) *Service Oriented Computing (SOC)*. Dagstuhl Seminar Proceedings, vol. 05462 (2006)
9. Erl, T.: *SOA Principles of Service Design*. Prentice-Hall, NJ (2007)
10. MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R.: Reference Model for Service Oriented Arch. 1.0 - Com. Spec. 1, OASIS Open (August 2, 2006)
11. Newcomer, E., Lomow, G.: *Understanding SOA with Web Services (Independent Technology Guides)*. Addison-Wesley Professional, Reading (2004)
12. Schwind, M., Kunkel, M.: Collaborative optimization of last mile networks for courier, express and parcel delivery services. In: *Multikonferenz Wirtschaftsinformatik (MKWI 2010)*, Göttingen, Germany (2010)
13. Caplice, C., Sheffi, Y.: Optimization-based procurement for transportation services. *Journal of Business Logistics* 24(2), 109–128 (2003)
14. Schwind, M., Hinz, O., Stockheim, T., Bernhardt, M.: Standardizing interactive pricing for electronic business. *Electronic Markets* 18(2), 165–174 (2008)
15. Schwind, M.: Design of combinatorial auctions for allocation and procurement processes. In: *7th International IEEE Conference on E-Commerce Technology 2005*, München, Germany, pp. 391–395 (2005)
16. Beck, R., Schwind, M., Hinz, O.: Grid economics in departmentalized enterprises. *Journal of Grid Computing* 6(3), 277–290 (2008)
17. Nagy, G., Salhi, S.: Location-routing: Issues, models and methods. *European Journal of Operational Research* 177(2), 649–672 (2007)
18. Gendreau, M., Potvin, J.Y., Bräysy, O., Hasle, G., Løkketangen, A.: Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In: Golden, B.L. (ed.) *The Vehicle Routing Problem: Latest Advances and New Challenges*. *Operations Research/Computer Science Interfaces Series*, vol. Part III, I, pp. 143–169. Springer, Heidelberg (2008)
19. Campbell, A.M., Thomas, B.W.: Challenges and advances in a priori routing. In: Golden, B.L. (ed.) *The Vehicle Routing Problem: Latest Advances and New Challenges*. *Operations Research/Computer Science Interfaces Series*, pp. 123–142. Springer, Heidelberg (2008)

# Gain in Transparency versus Investment in the EPC Network – Analysis and Results of a Discrete Event Simulation Based on a Case Study in the Fashion Industry

Jürgen Müller<sup>1</sup>, Ralph Tröger<sup>2</sup>, Alexander Zeier<sup>1</sup>, and Rainer Alt<sup>3</sup>

<sup>1</sup> Hasso Plattner Institute for IT Systems Engineering  
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany  
{juergen.mueller, zeier}@hpi.uni-potsdam.de

<sup>2</sup> Gerry Weber International AG  
Neulehenstraße 8, 33790 Halle/Westfalen, Germany  
r.troeger@gerryweber.de

<sup>3</sup> University of Leipzig  
Grimmaische Straße 12, 04109 Leipzig, Germany  
rainer.alt@uni-leipzig.de

**Abstract.** The diffusion rate of Radio Frequency Identification (RFID) technology in supply chains is lower than expected. The main reason is the doubtful Return On Investment (ROI) mainly due to high tag prices. In this contribution, we leverage a prototypical RFID implementation in the fashion industry, extend this prototype by a discrete event simulation, and discuss the impact of an Electronic Product Code (EPC)-enabled supply chain concerning higher visibility. Thereby, we illustrate the benefit of the EPC network for Supply Chain Event Management (SCEM). Future researchers are provided with the simulation data which is available online and can be investigated.

**Keywords:** EPC Network, Supply Chain Event Management, Simulation.

## 1 Introduction

The idea of RFID in the fashion industry is to attach a unique identifier to articles to be able to track them on their way in the supply chain. This results in higher visibility and enables the supply chain partners to react in time, e. g., taking measures to deliver a shipment in time even if there had been a delay in production [1].

Utilizing experiences of its prototypical RFID implementation with its distribution centers and selected retail stores, the fashion company Gerry Weber recently published a case study about how to obtain a positive ROI in RFID projects [2]. One recommendation was to deploy reusable RFID transponders to reduce expenditures. Since then, three developments have taken place: (a) major drop in RFID inlay prices; (b) applicability of RFID as Electronic Article Surveillance (EAS), and (c) advancements in the development of textile RFID labels.



Given this new situation, Gerry Weber has to reconsider its RFID strategy. In the former project, reusable RFID tags were used due to high tag prices. This is subject to change because of the achievements in RFID technology. Furthermore, we present a model of the potential structure of Gerry Weber's future supply chain including a global RFID deployment at all levels in the supply chain. To this end, we utilize the progress made in the EPC network which includes all components to collect RFID read events and share them with selected business partners [3, 4].

The remainder of this paper is organized as follows. In Section 2, we discuss related work and our research methodology. Section 3 describes Gerry Weber's supply chain how it is structured today. In Section 4, we discuss a projection of that supply chain if the EPC network was used. Section 5 describes how we modeled, implemented, and executed the supply chain presented in Section 4 in a simulation and shows the simulation results. Section 6 addresses the calculation of benefits for the utilization of the EPC network. Section 7 closes the paper with a conclusion.

## 2 Related Work and Methodology

In this section, we describe the related work about RFID, the EPC network, and SCEM. Finally, we present our research methodology and research questions.

### 2.1 RFID and the EPC Network

In recent years, application of RFID received increasing attention among researchers of information systems and operations management [5]. For instance, fields like RFID in manufacturing, cost-benefit sharing, drivers of adoption, system architecture, or integration in ERP systems have been addressed. Major aspects of RFID implementation especially for the apparel industry (requirements, business case, problem analysis and expected benefits) have also been investigated [6].

Concerning supply chain visibility and RFID, a comprehensive analysis was accomplished by Dittmann, who determined the appropriate level of visibility for supply chain networks from a normative, strategic and operational perspective [7]. Meydanoglu provided conceptual work regarding the use of RFID systems as data source for SCEM systems [8].

### 2.2 SCEM

SCEM can be characterized as an approach to achieve inter-organizational visibility over logistical processes enabling companies to detect critical exceptions in their supply chains in time. Thus, measures against those "events" can be taken proactively before they endanger given business plans [9].

Most authors agree on the perception that SCEM encompasses five core functionalities: *monitor*, *notify*, *simulate*, *control*, and *measure* [9, 10]. Especially for the two functions "monitor" and "measure", we see great usefulness by adopting RFID and the EPC network. The first one observes the supply chain for the occurrence of critical events (i. e., major delays or disturbances) by evaluating any incoming data. The latter one bridges the gap between SCEM and Supply Chain Management (SCM) by analyzing all data generated by the other four functionalities, i. e., number of events,

locality/ time of their occurrence, costs effects, and chosen action alternatives. This way, companies are able to identify certain patterns related to the causes of events in order to optimize their supply chain structures.

### 2.3 Methodology and Research Questions

Until now, no quantitative research about the consequences of RFID/ EPC utilization for a supply chain of the apparel/ fashion industry exists. As described by Müller et al., realistic test data is necessary to analyze information systems precisely [11].

For our research, we want to generate data about which events occur at which time in which place in a fashion supply chain. To this end, simulation is the appropriate methodology because it permits the evaluation of the complete supply chain prior to its implementation [12]. In the area of simulation, Discrete Event Simulation (DES) is widely used and well-established for supply chain simulations [13]. In DES, the activities in a system are represented as a chronological sequence of events [13]. Each event has a timestamp indicating the time when it occurs. Once the simulated time reaches this , the event is triggered, e. g. something is produced, shipped, etc.

As the simulation of an EPC network employment requires comprehensive data, this research method will be applied in combination with a case study [14].

Before conducting the investigation, the following research questions are set up:

- RQ<sub>1</sub>: What is the event frequency at different points of a fashion industry supply chain and how huge is the data volume if utilizing the EPC network?
- RQ<sub>2</sub>: Does the EPC network fit as data source for the monitoring and measuring task of SCEM systems?
- RQ<sub>3</sub>: How can the benefit of an edge on information through utilizing the EPC network in combination with SCEM systems be determined?

Our data sources are semi-structured interviews with seven fashion companies, desk research (press releases, annual reports, company web sites, etc.) and – as one of the authors is employed by Gerry Weber – data and information (such as quantity structures, packing instructions, average lead time tables, order data, etc.) directly from Gerry Weber.

## 3 Fashion Industry Case Study: Gerry Weber

With its three brands (GERRY WEBER, TAIFUN and SAMOON by GERRY WEBER along with several sub labels) the Gerry Weber group achieved a sales volume of about € 570 Mio. (circa \$ 800 Mio.) in the fiscal year 2007/2008 [15]. The German fashion company, which was founded in 1973, currently has 2,350 employees and follows a strategy of becoming increasingly vertically integrated. To this end, the company successively enlarges the number of its own retail stores.

Worldwide, the Gerry Weber group cooperates with around 200 manufacturers, several forwarders and logistics service providers, more than 1,470 shop-in-shops (ca. 1,160 in Germany) and over 300 retail stores. The suppliers are located in three major procurement markets: Far East, Turkey and Eastern Europe. After production, the garments are transported (either by sea, air, or road) to one of the European distribution centers (separated according to hanging/ lying garments). Most of the shipments in Far East and Turkey are consolidated in a cross docking center. After goods receipt

in a distribution center, all garments are checked for quality, conditioned, and transported to the retailers.

The company decided to evaluate the potentials of RFID to ease managing this complexity. Thus, RFID infrastructure was installed in all six distribution centers and four retail stores by March 2009. Initial experiences showed that RFID indeed is beneficial for Gerry Weber, especially regarding savings of time and costs. Currently, the company considers switching from reusable tags to one-way labels. One major advantage of using one-way tags is that those would be encoded with a Serialized Global Trade Item Number (SGTIN) [4]. In contrast to that, the reusable RFID tags had a persistent serial code that was associated with the product it was attached to.

## 4 EPC Network

The EPC network is one step towards the “Internet of things”. In this section, we will first describe the general EPC network architecture, then present the EPC network architecture for Gerry Weber, and finally discuss how a SCEM system based on the EPC network can be implemented.

### 4.1 General EPC Network Architecture

The EPC network is a set of standards and consists of a layered architecture [4]. A precondition is that an Electronic Product Code (EPC) uniquely identifies all items in the EPC network [4]. This includes articles, cartons, and containers. The data flow in the EPC network in a nutshell is depicted in Figure 1. Once an item is in the range of a RFID read point, a reader recognizes the unique identifier of the item and the current time. Together with the location of the read point, this data is sent to a middleware. The middleware enriches the data with a business step, i. e., “goods produced”, “goods issue”, or “goods receipt” [4]. After that, the data is sent to a repository implementing the EPC Information Services (EPCIS) standard [16]. Each company operates at least one EPCIS repository and stores all read events, which occurred in this particular company.

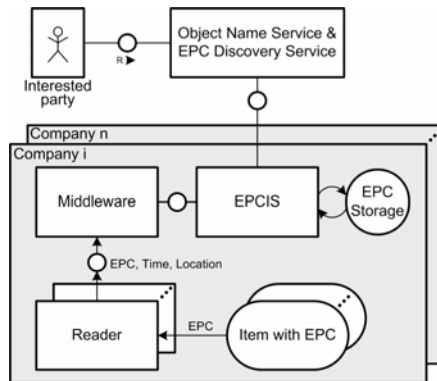


Fig. 1. EPC Network

Obviously, each company has to enforce access rights to secure its information. A situation might happen where the interested party does not have the complete information where an item has been in the supply chain and therefore it is unknown which EPCIS repository to query. For this situation, a Discovery Service acts as an intermediary service to retrieve all the involved parties [17]. The Discovery Service is still subject to research due to the expected data volume, privacy aspects, security aspects, and other challenges in the design of Discovery Services [17].

## **4.2 Gerry Weber EPC Network Architecture**

In this section, we describe how Gerry Weber's supply chain may be structured in the future. To this end, we assume that each participant in the supply chain is equipped with the necessary RFID infrastructure. Furthermore, Gerry Weber has access to the EPCIS repositories operated by its partners.

After production, an article is observed for the first time. Once the article is packed into a carton, a second read event is triggered to indicate which article is stored in which carton. The carton then is brought to the cross docking facility. This results in a read event at goods issue at the supplier and a read event at goods receipt at the cross docking facility. The carton then is loaded onto a container. This, again, results in a read event to determine that the carton is on this particular container. When the container leaves the cross docking facility, it is recognized by a reader and a read event is stored in the cross docking owners EPCIS repository. Subsequently, the container arrives at a distribution center and is recognized by a reader again. Once the carton is unloaded another read event is stored in the distribution centers' EPCIS repository. Finally, the carton leaves the distribution center which, again, is stored in the distribution centers EPCIS repository. Finally the article reaches a retailer and causes a read event at goods receipt. Then, there might be several read events at the retailer before the RFID tag maybe is deactivated.

Gerry Weber places subscriptions at each partner in the supply chain regarding all garments produced for Gerry Weber. Thus, each and every read event is forwarded to the read event repository of Gerry Weber and can be processed by its SCSEM system.

## **4.3 Discussion of a SCSEM System Based on the EPC Network**

Applying the EPC network as data source for a SCSEM system offers the opportunity to get reliable and fine-granular data even at the beginning of the supply chain. At present, the supply chains of the fashion industry before goods receiving at the distribution centers is not adequately transparent as the only source of information consists of manually transmitted data via e-mail, fax, or web-based platforms. However, having available multiple read points with the manufacturer and cross docking centers at which all passing garments can be detected, it is appropriate to concatenate the captured EPCs with context information data (such as Purchase Orders or Advanced Shipping Notices), whereby deviations can be discovered in real-time. Thus, all supply chain partners can be alerted if significant variations are observed which offers the opportunity to readjust scheduled processes in order to satisfy the customers in delivering the items in the right quantity at the right time to the right destination in the right condition.

To give an indication of the potential benefit of this edge on information: research findings from the accomplished case study showed that a bigger part of merchandise

is produced in Far East and mostly transported by sea [2]. Assuming to be informed about all delayed orders, the process owner can switch to air transport and reduce lead-time by more than 20 days.

Not only the detection of critical supply chain incidents (“monitor”) can be improved by adopting the EPC network, but also the optimization of the supply chain by recognizing certain patterns or systematic weak points, e. g. (“measure”). The latter one can be achieved by analyzing all incoming EPCIS messages, which provide information about time, place, and business context of detected Stock Keeping Units (SKU). In this way, for instance, a company is able to discover who or what causes delays, which offers the opportunity to eliminate or attenuate interfering factors (traffic jams at certain ports, poor performing suppliers or forwarders, etc.) in the medium or long term.

## 5 Simulation of Gerry Weber’s RFID-Enabled Supply Chain

We now simulate the implementation of the RFID and EPC network enabled supply chain of Gerry Weber to foresee the impact of the RFID introduction at the producers and consolidation centers. In this section, we describe the information included in the simulation implementation considerations, as well as the resulting simulation model including its parameters and results.

### 5.1 Base Data

The data for our simulation is based on two sources: results from a case study with seven fashion companies (Galeria Kaufhof, Gardeur, Gerry Weber, Maerz Muenchen, René Lezard, Seidensticker and Van laack) and – in more detail – data directly from Gerry Weber. As the first RFID read point is with the supplier and as an edge on information is more advantageous the earlier an event has been detected, we decided to display examples of events and countermeasures (see Table 1) only for the supply chain sections production and transport.

**Table 1.** Examples of Events and Counteractions

	Typical event ( $E_i$ )	Countermeasure	Probability
Production	$E_1$ : Delay through capacity constraint due to other customer orders or machine break-down	Alteration of transportation mode	2.00 %
	$E_2$ : Sudden employee leave/ strike	Relocation of production	0.25 %
	$E_3$ : Force majeure (thunderstorm, earthquake, power outage, e. g.)	Situational reaction	0.25 %
	$E_4$ : Too late arrival of fabrics/ trimmings	Prioritize production	5.00 %
Transport	$E_5$ : Strike (forwarder, sea-/ airport staff)	Switch to alternative forwarder/ sea-/ airport	1.00 %
	$E_6$ : Missing customs documents	Notify agency	1.50 %
	$E_7$ : Lost of maritime container/ ship foundering	None	0.05 %
	$E_8$ : Traffic jam	None	1.50 %

## 5.2 Implementation Considerations

For the implementation of the DES, we use Rockwell Arena 12.0, which is a standard product for DES [18]. We extend the standard capabilities of Arena to be able to simulate an RFID-enabled supply chain [18]. For this purpose, we implement a RFID tagging module, a RFID reader module, an aggregation module, and a disaggregation module. For detailed information, we refer to [11].

## 5.3 DES Model of the Fashion Supply Chain

Using the capabilities of Arena and our implemented modules, we model the Gerry Weber supply chain, as it would look like if complete adoption of RFID in the supply chain took place. Reflecting the large data set, which is created by our simulation, we limit the simulation to 1/10th of the size of Gerry Weber's supply chain.

**Simulation Parameters and Assumptions.** We assume an annual production of 2,500,000 garments (57 % laying, 43 % hanging), which are produced by 20 equally large producers. Our model includes two cross docking facilities (one in Far East and one in Turkey). The number of distribution centers in Europe is 6. We assume that there are 178 retailers.

Sea transport accounts for 77 % of the total production in Far East. The rest is transported via air transport. Transportation from Far East takes place in containers. In Turkey, the garments are transported to the cross docking facility via small trucks and then are reloaded to large trucks. The producers in Eastern Europe directly transport the garments to the European distribution centers using small trucks. Then, the garments are transported to the retailers using large trucks.

The air transport from cross docking to the distribution centers takes 5 days. Sea transport takes 30 days in average (Gaussian Distribution (GD), Standard Deviation (SD) of 0.5 days). The transportation from the cross docking facilities in Turkey and the producers in Eastern Europe to the distribution centers also takes 5 days in average (GD, SD of 0.2 days). The handling, warehousing and distribution to the retailers takes 10 days in average (GD, SD of 2 days). Between time of arrival at the retailers and time of sale, 40 days of time elapse in average (GD, SD of 8 days).

For the aggregation and disaggregation, the capacity of cartons etc. is important. We assume that 40 items of lying garments fit into a carton. One small truck has the capacity for 125 cartons of laying garments or 40 bars of hanging garments respectively. We assume that a large truck has the same capacity as a container. 4,200 hanging garments and 250 cartons of laying garments fit into a large truck/ container.

In Table 1, we presented the probabilities of events. All these events result in delays. In our model, the unit always is hours. These delays are modeled using a triangular distribution, which is presented in the form of three numbers, e. g., {10, 20, 30}. That means that the probability of a lower than 10-hour delay or a higher than 30-hour delay is zero. The highest probability is a 20-hour delay. The delays of events  $E_1 - E_4$  (production) are {3, 120, 336}. A strike or missing customs documents ( $E_{5,6}$ ) causes a delay of {48, 96, 144}. If a container is lost ( $E_7$ ), it is deleted from the system and never occurs again. The transportation delays caused by traffic jams ( $E_8$ ), i. e., all ground transportation in our model, are {6, 12, 24}.

**Simulation Results.** Table 2 shows read events and data volume that occurred in the simulation. We assume that Gerry Weber subscribed to all read events at all supply chain partner's EPCIS repositories. Thus, each EPCIS event (1,257 Bytes in size [16]) is pushed into the EPCIS repository of Gerry Weber. The SCEM system uses this repository as data source. Table 2 also lists the number of read events indicating delays, which give insights that may improve supply chain transparency.

The simulation results and the analysis of key figures, resulting read events, and data volume answer our research question RQ<sub>1</sub>. The raw data of the simulation is available online at <http://epic.hpi.uni-potsdam.de/Home/SOCLOG09Simul>.

#### 5.4 Simulation Implications

As it can be derived from the simulation results of Gerry Weber's supply chain, the EPC network indeed can contribute to a significantly enhanced supply chain visibility. In comparison to the current situation, all 1,229,522 item-level delays are monitored by the system. Since EPCIS inherently answers the what, when, why, and where of read events, it allows for various statistical analysis. Hence, research question RQ<sub>2</sub> can be affirmed: the fine granular and in real-time available data through the EPC network are highly appropriate as data source for the monitoring and measuring function of SCEM systems.

**Table 2.** Read Events and Resulting Data Volume

<b>Id</b>	<b>Key Figure</b>	<b>All Read Events</b>	<b>Data Volume (in GB)</b>	<b>Read Events Indicating Delays</b>
1	Total number	13,303,720	15.57	1,229,522
2	Read events at producers	5,118,817	5.99	373,751
3	in Far East	2,816,304	3.30	204,841
4	in Turkey	1,023,981	1.20	75,004
5	in Eastern Europe	1,278,532	1.50	93,906
6	Read events at cross docking	1,712,091	2.00	151,243
7	in Far East	1,252,366	1.47	109,148
8	in Turkey	459,725	0.54	42,095
9	Read events at distribution centers	2,148,807	2.52	205,030
10	Read events at retailers	4,324,005	5.06	499,498

## 6 Calculation of Benefits

For practitioners, profitability of investments in RFID/ EPC network is of high interest. However, determining the benefits of an edge of information is always company-specific. It especially depends on the kind of events the company has to struggle with as well as on the availability of counteractions in the case of an early knowledge of critical supply chain incidents.

As for the determination of the benefits, we suggest the decision-maker proceeding as follows: identify all critical potential events in the supply chain, prioritize them (according to their potential impact, probability of occurrence, and probability of detection, e. g.) and inquire the measures to be taken in each case. Afterwards, it is to be distinguished between situations that help a company because appropriate measures

can be taken and those where an edge on information would make no difference (lost of a maritime container, e. g., see “E<sub>7</sub>” in Table 1).

An event that was named by all fashion companies in the course of our prior case study research is a delay at the manufacturer (see “E<sub>1</sub>” in Table 1). Here, an early knowledge would be indeed beneficial as altering the transportation mode is an appropriate countermeasure in order to meet the delivery date.

For a fashion company, a delay in production has multiple negative impacts which can be taken into account for the business case: loss in sales (as a fashion line only sells well if all of its combinable trousers, blazers, shirts, skirts, accessories, etc. are available on the sales floor), costs for subsequent deliveries, and contract penalties for delivery date exceeding, e. g.. On the basis of those considerations, we suggest a formula that can be applied to estimate the benefit out of an early detection of “E<sub>1</sub>”:

$$B(E_1) = (\alpha - \beta) \cdot a \cdot n \cdot [(p \cdot \gamma) + c_S + c_P - c_T]$$

The symbols are defined as follows:

$B(E_1)$	early detection benefit using the example of event E <sub>1</sub>
$\alpha$	approx. part of events able to be detected through EPC network utilization
$\beta$	approx. part of events able to be detected by hitherto means
$a$	part of units where counteractions can be applied to compensate the delay
$n$	total number of delayed units with the manufacturer’s site p. a.
$p$	average price per unit
$\gamma$	approx. loss in sales due to incomplete fashion lines
$c_S$	costs for subsequent deliveries
$c_P$	costs for contract penalties
$c_T$	costs to compensate the delay

Due to competitive reasons, we are not able to use genuine data (prices, costs, etc.) to illustrate the calculation. Nevertheless, we present realistic assumptions.

For our example we presume that a company is able to detect around 90 % of all supply chain events by conventional technologies like telephone or e-mail and that by utilization of the EPC network it is expected to increase that by 9 %. We assume that the company has the opportunity to switch from standard to express transport mode (for instance, from sea to air transport) in seven out of ten cases. The annual number of produced garments is the same as applied in our simulation: 2,497,599.

For lost sales and contract penalty it is necessary to set an average price for a unit, which for our cause is 50 \$. In the case of an incomplete fashion line we assume a loss in sales of 5 %. In our scenario, a retailer is permitted to request a discount of 10 % relating to the price. As for the expenses for a subsequent delivery, we assume additional costs of 2 \$ per unit for all additional administrative, handling and transportation processes.

As a rule of thumb, costs for air transport exceed those for carriage by sea by a factor of 5. Nevertheless, cargo rates always vary due to the fluctuating demand and supply ratio or oil prices. As a simplification, we assume the same price difference for all other transport mode alterations. The savings resulting from the utilization of the EPC network in combination with SCEM systems can be determined as follows:

$$B(E_1) = (99\% - 90\%) \cdot 70\% \cdot 99,525 \cdot [(50\$ \cdot 5\%) + 2\$ + (50\$ \cdot 10\%) - 0.80\$] = 54,549.65\$$$



Although this sample calculation for the case of  $E_1$  alone marks considerable potential savings, it is to be reminded that our calculation encompasses only three exemplary benefits for one sole type of event and that it is based on a 10 times smaller volume as it is the case in reality. Though, fulfillment of target processes in the supply chain is jeopardized by a multitude of potential events. Hence, we see a much higher return when all of a company's most critical events are considered. As for  $RQ_3$ , we illustrated a way how the benefits of an early detection of critical supply chain events can be determined.

## 7 Conclusion

In this contribution, we presented a simulation based on results from a case study with seven fashion companies and data directly from Gerry Weber. We designed the EPC network for future fashion industry supply chains and, based on this, developed a simulation model in the factor of 1:10. The simulation results were presented in detail and they are available online for future researchers and their investigations.

## References

1. Melski, A., Müller, J., Zeier, A., Schumann, M.: Assessing the Effects of Enhanced Supply Chain Visibility through RFID. In: 14th Americas Conference on Information Systems, Toronto, Canada (2008)
2. Goebel, C., Tröger, R., Tribowski, C., Günther, O., Nickerl, R.: RFID in the Supply Chain: How To Obtain A Positive ROI, The Case of Gerry Weber. In: International Conference on Enterprise Information Systems, Milan (2009)
3. Thiesse, F., Floerkemeier, C., Harrison, M., Michahelles, F., Roduner, C.: Technology, Standards, and Real-World Deployments of the EPC Network. *IEEE Internet Computing* 13(2), 36–43 (2009)
4. EPCglobal Inc.: The EPCglobal Architecture Framework. version 1.3 (2009)
5. Ngai, E., Moon, N., Riggins, F., Yi, C.: RFID Research: An Academic Literature Review (1995-2005) and Future Research Directions. *Int. Journal of Production Economics* (2008)
6. GS1 Germany et al.: Supply Chain Management in the European Textile Industry. Technical reports (2007)
7. Dittmann, L.: Der angemessene Grad an Visibilität in Logistik-Netzwerken. Die Auswirkungen von RFID. Deutscher Universitäts-Verlag, Wiesbaden (2006)
8. Meydanoglu, E.: RFID-Systeme als Datenlieferant von SCEM-Systemen. *PPS-Management* (13), 41–44 (2008)
9. Heusler, K., Stoelzle, W., Bachmann, H.: Supply Chain Event Management. Grundlagen, Funktionen und potenzielle Akteure. *WiSt* (1), 19 (2006)
10. Ijioui, R., Emmerich, H., Ceyp, M., Diercks, W.: Auf Überraschungen vorbereitet. Transparenz durch Supply Chain Event Management. *REFA-Nachrichten* (2) (2007)
11. Müller, J., Poepke, C., Urbat, M., Zeier, A., Plattner, H.: A Simulation of the Pharmaceutical Supply Chain to Provide Realistic Test Data. In: 2009 International Conference on Advances in System Simulation, Porto, Portugal (2009)
12. Chang, Y., Makatsoris, H.: Supply Chain Modeling Using Simulation. *International Journal of Simulation* 2(1), 24–30 (2001)

13. Fujimoto, R.M.: Parallel Discrete Event Simulation. In: 21st Conference on Winter Simulation, pp. 19–28. ACM, New York (1989)
14. Yin, R.: Case Study Research. Design and Methods. Applied Social Research Method Series. Sage Publications, Beverly Hills (1984)
15. Weber, G.: Annual Report 2007/2008 (2008)
16. EPCglobal Inc.: EPC Information Services (EPCIS). version 1.0.1 (2007)
17. Müller, J., Oberst, J., Wehmeyer, S., Witt, J., Zeier, A.: An Aggregating Discovery Service for the EPCglobal Network. In: Proceedings of the 43th Hawai'i International Conference on System Sciences (HICSS), Koloa, Hawaii, USA (2010)
18. Kelton, W.D., Sadowski, R.P., Sturrock, D.T.: Simulation with Arena, 4th edn. (2006)

# Using Automated Analysis of Temporal-Aware SLAs in Logistics\*

Carlos Müller, Manuel Resinas, and Antonio Ruiz-Cortés

Dpto. Lenguajes y Sistemas Informáticos  
ETS. Ingeniería Informática - Universidad de Sevilla, Spain - España  
41012 Sevilla, Spain - España  
{cmuller,resinas,aruiz}@us.es

**Abstract.** Service level agreements (SLAs) establish the terms in which a logistics service may be provided or consumed. During the last years we have been studying techniques to perform an automated analysis of expressive and realistic SLAs, which makes the agreement creation process easier for involved parties. Firstly, we extended WS-Agreement specification to allow to apply any type of validity periods to SLA terms. Later, we dealt with the automated analysis of SLAs by proposing the explaining of SLAs inconsistencies and non-compliance scenarios. In this paper we show how these contributions are necessary to enable a logistic scenario of package tracking by providing examples for each proposal. We also include a final discussion on the convenience of performing a merge of all contributions to enable a better application of SLAs to logistic scenarios.

## 1 Introduction

Service-Oriented Computing with its existing set of standards, promotes adaptive supply chain management concepts, flexible and re-configurable logistics service provisioning in supply chains. Thus, service level agreements (SLAs) establish the terms in which a logistics service may be provided or consumed. During the last years we have been studying techniques to perform an automated analysis of expressive and realistic SLAs which makes the agreement creation process easier for involved parties [3,5,4]. In order to apply our theories, we used WS-Agreement specification [1], which defines an XML-based language and a protocol for advertising the capabilities and preferences of services providers in templates, and creating agreements based on them. In IC-SOC'07 [3], we proposed a temporal domain specific language (DSL) which increases the temporal-awareness of WS-Agreement specification. Such temporal DSL, allows expressive periodical/non-periodical and disjoint/non-disjoint

---

\* This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT projects Web-Factories (TIN2006-00472), and SETI (TIN2009-07366); and project P07-TIC-2533 funded by the Andalusian local Government.

validity periods to the terms of SLAs. Later in ICSOC'08 [5] and ICSOC'09 [4], we dealt with the automated analysis of SLAs proposing the explaining of SLAs inconsistency and non-compliance scenarios respectively. We also provide a constraint-based solution model and two proof-of-concepts, available for testing at <http://www.isa.us.es/wsag>. Mentioned [5,4] papers were inspired by previous papers [2,6], which focus on checking whether an SLA is compliant with another one but without providing any explanation for the non-compliance, if any.

**Contribution:** This paper is focused on applying our previous contributions on automated analysis of SLAs in logistics domain. To this end, we take our previous work in [3,5,4] separately, and we apply each one into a package tracking scenario to validate our contributions in logistics domain. Furthermore, we provide a final discussion on the convenience of performing a merge of all contributions to a better application to logistics domain.

The remainder of the paper is organized as follows: Section 2 describes the used subset of WS-Agreement; Section 3 includes the agreement creation process in a motivating scenario of package tracking providing; Section 4 applies to logistics our contributions of: temporal-aware SLAs in Section 4.1, explaining SLA inconsistencies in Section 4.2, and explaining the non-compliance between templates and agreement offers in Section 4.3, and finally Section 5 concludes this paper and raises a discussion on future work.

## 2 WS-Agreement\* in a Nutshell

Due to the flexibility and extensibility of WS-Agreement, in [5,4] we focused on WS-Agreement\*, which is a subset of WS-Agreement (cf. <http://www.isa.us.es/wsag> for details about these differences). WS-Agreement\* just imposes several restrictions on some elements of WS-Agreement but it keeps the same syntax and semantics, therefore any WS-Agreement document that follows these restrictions is a WS-Agreement\* document. Furthermore, note that, although WS-Agreement\* is not as expressive as WS-Agreement, it does allow to express complex agreement documents as those in Figure 1, in which the elements of several WS-Agreement\* documents in a packing tracking services providing scenario are depicted.

- **Name & Context** identifies the agreement and other information such as a template name and identifier, if any, referring to the specific name and version of the template from which the current agreement is created. For instance, context of Figure 1(c) refers to Template of Figure 1(a).
- **Terms** can be composed using the three term compositors described in [1]: **All** ( $\wedge$ ), **ExactlyOne** ( $\oplus$ ), and **OneOrMore** ( $\vee$ ). All terms in the document must be included into a main **All** term compositor. Figure 1(a) includes **All** and **ExactlyOne** term compositors. Terms can be divided into: **Service Terms** including:

- **Service properties** must define all variables that are used in the guarantee terms and other agreement elements, explained later. In Figure [II\(a\)](#), the variables are the *availability of the computing service* (Availability), the *response time for a request from server, without considering network traffic delays* (ResponseTime), and the *initial cost for the service* (InitCost). The type and general range of values for each variable is provided in an external document such as the ad-hoc XML document depicted in Figure [II\(b\)](#).
- **Service description terms** provide a functional description of a service, i.e. the information necessary to provide the service to the consumer. They may set values to variables defined in the service properties (e.g. InitCost=20 in Figure [II\(a\)](#)) or they may set values to new variables. Type and domains are defined in external files such as XML Schemes (e.g. Carrier=MyCarrier in Figure [II\(a\)](#)).

**Guarantee terms** describe the service level objectives (SLO) that a specific obligated party must fulfill, and a qualifying condition that specifies the validity condition under which the SLO is applied. For instance the Lower-Availability guarantee term included in Figure [II\(a\)](#).

A WS-Agreement template is an agreement document with the structure of a WS-Agreement document as described in previous section, but including agreement creation constraints that should be taken into account during the agreement creation process [II](#). These **Creation Constraints** describe the variability allowed by the party that makes the template public. They include (1) **Constraints** involving the values of one or more terms, for instance the FinalCost definition of “Constraint 1” of Figure [II\(a\)](#); or (2) **Items** specifying that a particular variable of the agreement must be present in the agreement offer, typically as a service description term, and its range of values. For instance, the item elements of Figure [II\(a\)](#) define three variables: the *number of dedicated GPS satellites used for locating packages* (GPSs), the *increase of the cost due to the selected ResponseTime* (ExtraRespTimeCost), and the *final cost for the service* (FinalCost).

### 3 Creation of SLAs in a Motivating Logistics Scenario

A typical interaction process to create agreements using templates and offers, applied to a package tracking service providing scenario, could be as follows: (1) an initiator party, which needs a package tracking service, takes the public template depicted in Figure [II\(a\)](#) from a responder party. This template describes the agreement terms and some variability that must be taken into account by the initiator in order to achieve an agreement; (2) the initiator creates an agreement offer based on the public template, as Figures [II\(c\)](#) and [II\(d\)](#) depicts, and sends it to the responder party; (3) finally, the responder party may accept or not the agreement offer received.

However, an agreement signed by all interested parties should be carefully created because a failure to specify their terms could carry penalties to the

initiating or responding party. Therefore, agreement terms of templates and agreement offers should be specified in a consistent way, avoiding contradictions between them. In case of inconsistent documents an explanation of why it is inconsistent would be very appealing. For instance, the agreement offer of Figure 1(c) includes inconsistent terms emphasized with a cross. Moreover, once established that the agreement offer and the initial template are consistent, parties must ensure the compliance between agreement templates and offers. If they are not compliant, an explanation would make it easier to solve problems between parties. Figure 1(d) depicts a non-compliant agreement offer with template of Figure 1(a), because of the underlined term.

Furthermore, SLAs of logistics services usually require a high degree of temporal-awareness. For instance, the package tracking service scenario of Figure 1 could be included in a supply chain with more tasks such as the SMS sending to the final user with package tracking information. If we know that in Christmas period the tracking information is highly demanded by people who want to know whether their packages will arrive on time or not, we need control when and how the service can be requested to satisfy our users. Therefore, we need to include validity periods applied to the whole SLA and to concrete SLA terms.

## 4 Our Contributions

### 4.1 Applying Temporal-Aware SLAs in Logistics

Figure 2 depicts the package tracking service providing scenario of Figure 1 but including the periodical and disjoint validity periods: “Global Period”, “LessRespTime Period” and “MoreRespTime Period”, depicted in Figure 2(c). “Global Period” informs about *when* the service can be requested -it is active on 2010 and it just stops the last hour of every Sunday due to maintenance operations-; while “LessRespTime Period” and “MoreRespTime Period” inform about *how* service can be requested depending on the available number of dedicated servers and GPS satellites used to locate the package: On Monday-Friday from 8:00 to 18:00 the ResponseTime decreases due to a higher number of dedicated servers and GPS satellites used to locate the package at work hours; and the ResponseTime increases at home hours because of a lower number of GPS satellites. In the example we have removed the XML definitions for the validity periods for simplicity (cf. [3], for details on XML validity periods definitions). Figure 2(b) depicts a compliant agreement offer for template of Figure 2(a). In the agreement offer, it is included two optional groups of terms with different values for ResponseTime and GPSs service properties, depending on when the service is requested.

The study of temporal aspects in web services specifications starts with Octavio Martín Díaz’s PhD and in ICSOC’05 [2], we overcome the problem of the temporal covering, which involves covering a temporal-aware service demand with several temporal-aware service offers when none of the offers covers totally the demanded period. Later, in ICSOC’07 [3], we present our graphical representation for temporality and we develop a temporal DSL to specify any type of

**Template "id:Template v1.0"**

**Name** PackTrackingUpTo5GPSs

**Context**  
 - AgInitiator: IneedTracking Corp.  
 - ServiceProvider: AgreementResponder

**ServiceProperties**  
 - Availability "metricXML:Percentage"  
 - ResponseTime "metricXML:Time"  
 - InitCost "metricXML:Cost"

**ServiceDescriptionTerm**  
 - InitCost = 20 ...  
 - Carrier = MyCarrier

**GuaranteeTerm "GuaranteedRespTime"**  
 - SLO: ResponseTime >= 5 & <= 60

**Exactly One (xor)**

**GuaranteeTerm "LowerAvailability"**  
 - QualifCondition: ResponseTime >= 10  
 - SLO: Availability >= 90 & <= 100

**GuaranteeTerm "HigherAvailability"**  
 - QualifCondition: ResponseTime < 10  
 - SLO: Availability >= 95 & <= 100

**CreationConstraints**

**Item 1** - GPSs: integer [1,5]

**Item 2** - ExtraRespTimeCost: integer [1, ∞]

**Item 3** - FinalCost: integer [1, ∞]

**Constraint 1**  
 FinalCost = InitCost + ExtraRespTimeCost + GPSs x 10

**Constraint 2**  
 ResponseTime < 10 → ExtraRespTimeCost = 15

**Constraint 3**  
 ResponseTime >= 10 → ExtraRespTimeCost = 0

(a) A WS-Agreement template with general and item constraints.

**MetricXML**

- Percentage: integer [1,100]
- Time: integer [1,∞]
- Cost: integer [1,∞]

(b) Content of the ad-hoc XML document for the variable domains of Figures "a", "c", and "d".

**AgreementOffer "id:InconsistentOffer"**

**Name** Bad Required ResponseTime

**Context**  
 - AgInitiator & ServiceProvider same as in template  
 - TemplateID: Template v1.0  
 - TemplateName: PackTrackingUpTo5GPSs

**ServiceProperties** same as in template

**ServiceDescriptionTerm**  
 - InitCost = 20  
 - ResponseTime = 4 ❌  
 - GPSs = 3  
 - ExtraRespTimeCost = 15  
 - FinalCost = 65 (20 + 15 + 3 x 10)  
 - Carrier = MyCarrier

**GuaranteeTerm "GuaranteedRespTime"**  
 - SLO: ResponseTime >= 5 & <= 60 ❌

**GuaranteeTerm "HigherAvailability"**  
 - QualifCondition: ResponseTime < 10  
 - SLO: Availability >= 95 & <= 100

(c) An offer inconsistent with itself.

**AgreementOffer "id:Non-CompliantOffer"**

**Name** More GPSs Demanded

**Context**  
 - AgInitiator & ServiceProvider same as in template  
 - TemplateID: Template v1.0  
 - TemplateName: PackTrackingUpTo5GPSs

**ServiceProperties** same as in template

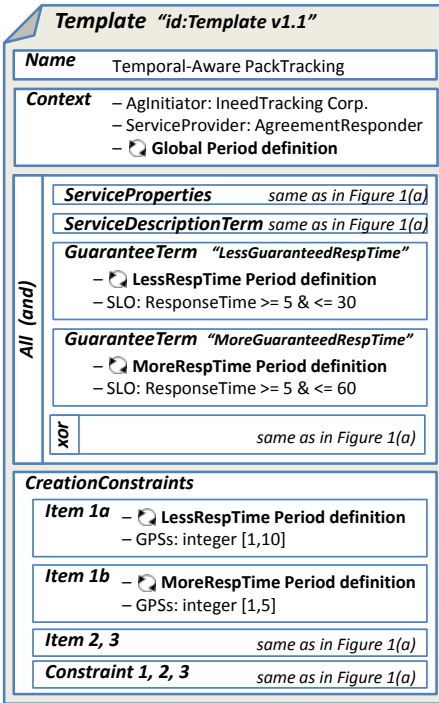
**ServiceDescriptionTerm**  
 - InitCost = 20  
 - ResponseTime = 50  
 - GPSs = 10 ❌  
 - ExtraRespTimeCost = 0  
 - FinalCost = 120 (20 + 0 + 10 x 10)  
 - Carrier = MyCluster

**GuaranteeTerm "GuaranteedRespTime"**  
 - SLO: ResponseTime >= 5 & <= 60

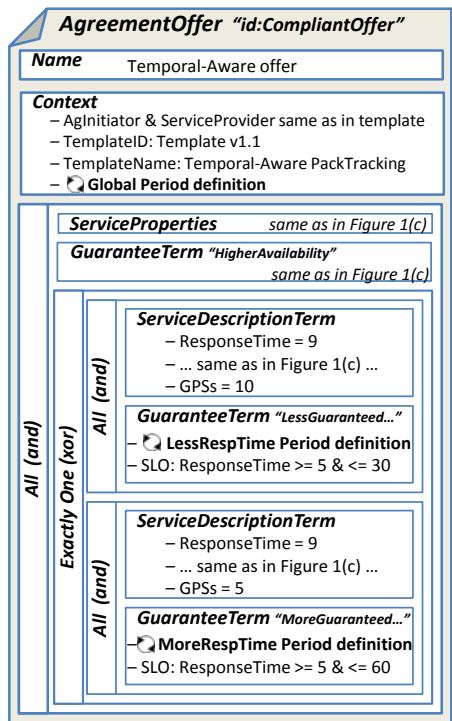
**GuaranteeTerm "LowerAvailability"**  
 - QualifCondition: ResponseTime >= 10  
 - SLO: Availability >= 90 & <= 100

(d) An offer non-compliant with template "a", demanding more dedicated GPS satellites.

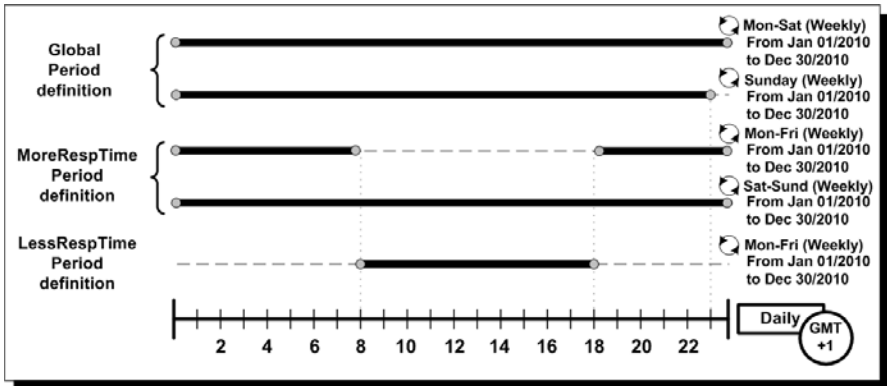
Fig. 1. Template and Offers WS-Agreement\* documents



(a) A WS-Agreement template with validity periods for the whole SLA, GuaranteeTerms, and Items.



(b) A Compliant offer with template "a", considering the higher value for GPSs and ResponseTime at each validity period.



(c) Validity periods definitions for the whole SLA, GuaranteeTerms and Items of Figures "a", and "b".

Fig. 2. Temporal-Aware Template and Offer WS-Agreement\* documents



validity periods: (1) periodical/non-periodical, repeated or not on time with a concrete frequency, and (2) disjoint/non-disjoint, with gaps inside the period or not. Such temporal DSL is defined as an XML schema which can be downloaded from <http://www.isa.us.es/wsag>, but represented in [3] as an UML diagram. We also discuss in [3] how to apply such temporal DSL to the whole agreement or to single terms of WS-Agreement documents.

## 4.2 Applying the Explanation of SLA Inconsistencies in Logistics

The first operation of automated analysis of SLAs studied by us is “the explaining of SLA inconsistencies”. Such analysis operation would be of great help for the management of supply chains with logistics services regulated by means of SLAs. Mainly, because a failure to specify the SLAs terms could carry penalties to the initiating or responding party. Therefore, agreement terms should be specified in a consistent way, avoiding contradictions between them. However, depending on the complexity of the agreement, this may become a challenging task. The application of our contribution in [5] would obtain error-free SLAs specifications in logistics scenarios. For instance, Figure II(c) depicts an agreement offer for template of Figure II(a). However, the agreement offer is not consistent due to a bad value assignment to the ResponseTime property. Thus, if the provider misses this mistake and sign the agreement, the consumer could claim for a compensation and vice versa.

The study of this first operation of automated analysis of SLAs started in [5] where we present a rigorous mapping from WS-Agreement\* subset of WS-Agreement to constraint satisfaction problems with the objective of analyze such resulting problem and give explanations for inconsistencies, if any. A proof-of-concept is developed and available for testing at <http://www.isa.us.es/wsag>.

## 4.3 Applying the Explanation of Non-compliance between Templates and Agreement Offers in Logistics

The second operation of automated analysis of SLAs studied by us is “the explaining of non-compliance between several SLAs”. Such analysis operation would be of great help for the management of supply chains with logistics services regulated by means of SLAs. Mainly, because it allows to identify why a template and an agreement offer are not compliant. This information can be used to provide solutions by means of checking the reported errors or by means of a negotiation process. In the logistics scenario of Figure II, a supply chain manager could be interested in the package tracking service described in template II(a), but during the agreement offer specification depicted in Figure II(d) he may commit a mistake in the GPSs value assignment. The error is to assign 10 to GPSs, without considering the template creation constraint “Item 1”, in which 5 is the higher value allowed for GPSs. In other cases a negotiation process could be needed if the provider is interested in a non-totally-compliant agreement offer.

The study of this second operation of automated analysis of SLAs started in [4] where we take the mechanism of templates and agreement offers of

WS-Agreement as reference, we include some rigorous definitions of compliance between templates and agreement offers, and the explaining of non-compliance scenarios. A proof-of-concept based on constraint satisfaction problems has been developed and it is available for testing at <http://www.isa.us.es/wsag>.

## 5 Conclusions and Discussion on Future Work

In this paper we have motivated the need for our recent contributions [3,5,4] in logistics scenarios and we have proof that they are applicable by means of examples for each contribution in a package tracking service proving scenario. More specifically, we use the mechanism of templates and agreement offers of WS-Agreement specification to present: (1) a temporal-aware scenario in Figure 2 with any kind of validity periods applied to the whole SLA or to different SLA terms; and (2) an inconsistent and non-compliant scenario in Figure 1 depicting an inconsistent agreement offer and a non-compliant agreement offer with an initial template.

Such proof has been performed for each contribution on its own. Therefore, we raise now the following question: *Is it necessary to merge our contributions to a better application to logistics?*

It is obvious that performing a merge of the mentioned contributions we would obtain an *automated analysis of SLAs with a high degree of temporal-awareness* which is very appealing in logistics scenarios because it allows the explaining of inconsistent and non-compliant SLAs with a high degree of temporal-awareness. Thus, explaining inconsistent or non-compliant temporal-aware scenarios we could: (1) solve contradictory terms with overlapped validity periods, as for instance “ResponseTime $\geq$ 20” from Monday to Friday and “ResponseTime $<$ 20” from Friday to Sunday; and (2) allow contradictory terms with non-overlapped validity periods, as for instance “ResponseTime $\geq$ 20” from Monday to Friday and “ResponseTime $<$ 20” from Saturday to Sunday.

However, the merging problem can easily be described, but it has a complex solution because we use constraint-based problems as paradigm to solve the explaining of SLAs errors. The inclusion of periodical and disjoint validity periods in such constraint-based problem may obtain a problem too complex to be solved. A possible solution could be to perform a pre-processing of the validity periods which is still under study. Therefore, at the moment we have focused on the study of main operations of automated analysis of SLAs without considering the complex temporal-awareness studied in [3]. In addition, we also have more analysis operations to study, as for instance the analysis of overlapping or differences in an SLA or several SLAs.

## References

1. Andrieux, et al. of the OGF Grid Resource Allocation Agreement Protocol WG. Web Services Agreement Specification (WS-Agreement), v. gfd.107 (2007)
2. Martín-Díaz, O., Ruiz-Cortés, A., Durán, A., Müller, C.: An approach to temporal-aware procurement of web services. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSSOC 2005. LNCS, vol. 3826, pp. 170–184. Springer, Heidelberg (2005)

3. Müller, C., Martín-Díaz, O., Ruiz-Cortés, A., Resinas, M., Fernández, P.: Improving Temporal-Awareness of WS-Agreement. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 193–206. Springer, Heidelberg (2007)
4. Müller, C., Resinas, M., Ruiz-Cortés, A.: Explaining the Non-Compliance between Templates and Agreement Offers in WS-Agreement\*. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 237–252. Springer, Heidelberg (2009)
5. Müller, C., Ruiz-Cortés, A., Resinas, M.: An Initial Approach to Explaining SLA Inconsistencies. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 394–406. Springer, Heidelberg (2008)
6. Ruiz-Cortés, A., Martín-Díaz, O., Durán, A., Toro, M.: Improving the Automatic Procurement of Web Services using Constraint Programming. *Int. Journal on Co-operative Information Systems* 14(4) (2005)

# Flexible SLA Negotiation Using Semantic Annotations

Henar Muñoz<sup>1</sup>, Ioannis Kotsiopoulos<sup>2</sup>, András Micsik<sup>3</sup>, Bastian Koller<sup>4</sup>,  
and Juan Mora<sup>1</sup>

<sup>1</sup> Telefónica Investigación y Desarrollo S.A, Valladolid, Spain  
{henar,mora}@tid.es

<sup>2</sup> The University of Manchester, United Kingdom  
ioannis@cs.man.ac.uk

<sup>3</sup> MTA SZTAKI, Budapest, Hungary  
micsik@sztaki.hu

<sup>4</sup> High Performance Computing Center Stuttgart, Germany  
koller@hlrs.de

**Abstract.** Moving towards a global market of services requires flexible infrastructures that will deal with the inevitable semantic heterogeneity that occurs during the negotiation that precedes the trading of a service. In order to reach an agreement, the negotiating parties need to understand the concepts describing the Quality of Service (QoS) terms which are part of the Service Level Agreement (SLA). The use of semantic annotations can increase the level of flexibility and automation, allowing the two parties to use their own terminology as long as it is related to the commonly understood conceptual model. This paper discusses how SLA negotiation will benefit from the use of a lightweight backwards compatible semantic annotation mechanism.

**Keywords:** Service Level Agreement (SLA), semantic annotations, SLA negotiation, ontologies.

## 1 Introduction

Since the 1980s, Service Level Agreements (SLAs) were established as tools for stating the Quality of a Service. Mainly used in the Telecommunication domain, and used as paper print-outs there was a tendency in the research community to try to adapt the SLA concepts on other domains [1]. Considerable research on SLAs and related technologies has been carried out in the Grid domain, especially in projects dealing with Virtual Organizations and Grids supporting relationships of its users.

Despite the extensive research on SLAs, the exact coverage of an SLA in terms of electronic contracts is still under debate. Some argue to use SLA as a complete replacement of contracts, but in reality the complexity and the sheer size of contracts has led us to assume that SLAs will cover statements of parts of electronic contracts defining the QoS terms, as well as the obligations on all involved business parties. The existence of SLAs is based on the assumption that an agreement can be reached. This may be simple to achieve when we consider services which can be represented on plain system parameters (e.g. lead time or number of containers in the domain of

logistics) or when the business participants speak a common language. But with the involvement of different language domains the creation of SLAs gets more complex and with this increased complexity it loses the interest of business end-users.

The increase of the service market and the emergence of cloud computing requires tools that will efficiently handle the whole SLA lifecycle without the need for human interaction. One of the major obstacles in this attempt for automation is the lack of formal semantics associated with the SLA terminology. Ontologies and semantic technologies can address this obstacle to a large extent as we will demonstrate with this paper. The rest of the paper is structured as follows, Section 2 discusses related work. Section 3 presents the proposed specification, Section 4 the related ontologies and in Section 5 we explain its applicability for SLA negotiation. Section 6 provides the details about our implementation experiments. Finally, our conclusions and ideas for further work are presented in Section 7.

## 2 Related Work

The presented work targets mainly two different areas; SLA Negotiation protocols and Semantic Annotations of SLAs. In the following we give a short overview of identified work in these areas.



**Fig. 1.** Layers in SLA negotiation

With the advent of the usage of SLAs in e-business, several attempts have been made to support the creation (negotiation) of Service Level Agreements in an optimal way for all involved business entities. Hung et al [4] describe WS-Negotiation as an independent declarative XML language for Service Providers and Customers to come to an agreement. Their model was based on three layers (cf. Figure 1), the Negotiation message (describing the contents of the exchanged message), the Negotiation Protocol (describing how to exchange these messages) and the Negotiation Decision Making (describing the decision taking based on internal strategies of the entity).

Defining the Negotiation Message, the most mature approaches are WSLA [3] and WS-Agreement [2]. WSLA, which was published by IBM in 2003 (no further updates), provides a specification for the definition and monitoring of SLAs in a Web Service environment. The other well established specification was provided by the Grid Resource Allocation Agreement Protocol Working Group of OGF (GRAAP-WG). WS-Agreement defines a language and protocol to represent the services of providers, creating agreements based on offers and also monitors the

agreement compliance at runtime. Both specifications were already taken up by different research communities which tend currently to use a mix of both specifications. The value of using them together seems to be much higher than building only on one or the other individually (cf. TrustCoM [5] and BREIN [6] projects).

With respect to the Negotiation Protocol, a variety of specifications for that area are available. The most referenced one is WS-AgreementNegotiation [7], even though it is still under discussion. So far it follows the Discrete Offer (also known as Take-It-Or-Leave-It) approach which is simple to implement but not efficient and flexible enough for complex application areas. Alternatives like multi-round negotiation or auctions are also examined, but have never reached the status of a well balanced specification for the Web Service domain. Especially auctions [8] are mainly settled in the multiagent domain but rarely examined by Web Service research activities.

Even though there were many thoughts about semantic enhancements of SLAs, not many activities really provided results originating from this research topic. Frankova et al [9] present an approach showing a formal definition of WS-Agreement, based on the concepts of finite automata. Even though the approach is very interesting, it uses semantic concepts in a different way and therefore a comparison with the work shown in this paper is nearly impossible.

Oldham et al [10] present the Semantic WS-Agreement Partner Selection (SWAPS). This approach covers how to overcome limitations of WS-Agreement with respect to syntactical matching; it enables intelligent partner selection based on SLA technologies. These limitations exist due to the scanty defined XML domain vocabulary of WS-Agreement. The main focus of SWAPS is on closing these gaps, by adding more structure to the original specification, to enable automatic parsing and reasoning. This additional structure is based on WSLA and introduces the OntConcept element in the schema, which clarifies the ambiguities of agreements by linking expression parameters directly to the concrete ontology concepts. This paper works further in the direction of SWAPS by using semantic annotation and also extending the idea towards negotiation.

### 3 The SA-SLA Specification

Customers and service providers need a common language for SLA negotiation, in order to understand each other's offers and bids. Current SLA specifications only define the format of expressing an SLA offer, but the content can use different languages, terms and metrics. To solve the problem, one possibility could be to construct an XML schema definition of SLA metrics, but the general experience is that a semantic definition supports better the re-use, re-combination and translation of descriptive elements.

In this paper we propose the usage of semantic annotations inside current SLA descriptions. A semantic annotation is additional information that identifies or defines a concept in a semantic model in order to describe part of any document element. This way of annotating does not require an extension of the XML Schema, but it uses the extensibility elements of the XML schema as placeholders for the annotations. Thus, annotations are simply ignored by tools that cannot interpret the additional semantics,

qos:Kilogram	a	owl:Class ; rdfs:subClassOf qos:WeightUnit .	OWL QoS Ontology
qos:WeightCapacity	a	owl:Class ; rdfs:subClassOf qos:Capacity .	
qos:WeightCapacityMetric	a	owl:Class ; rdfs:subClassOf qos:CapacityMetrics .	
<pre>&lt;xsi:attribute name="modelReference" type="listOfAnyURI" /&gt; &lt;xsi:attribute name="liftingSchemaMapping" type="listOfAnyURI" /&gt; &lt;xsi:attribute name="loweringSchemaMapping" type="listOfAnyURI" /&gt;</pre>			SA-SLA specification
<pre>&lt;wsla:SLAParameter name="MaxWeight" sawsdl:modelReference="qos:WeightCapacity" type="long" unit="kg"&gt;   &lt;wsla:Metric&gt;MaxWeightMetric&lt;/wsla:Metric&gt;   ... &lt;/wsla:SLAParameter&gt; &lt;wsla:Metric name="MaxWeightMetric" sawsdl:modelReference="qos:WeightCapacityMetric" type="long"&gt;   &lt;MeasurementDirective xsi:type="kg" sawsdl:modelReference="qos:Kilogram"&gt; &lt;/wsla:Metric&gt; &lt;wsag:ServiceLevelObjective&gt;   ...   &lt;wsla:Predicate xsi:type="wsla:Greater"&gt;     &lt;wsla:SLAParameter&gt;MaxWeight&lt;/wsla:SLAParameter&gt;     &lt;wsla:Value&gt;3000&lt;/wsla:Value&gt;   &lt;/wsla:Predicate&gt;   ... &lt;/wsag:ServiceLevelObjective&gt;</pre>			SLA template annotated with SA-SLA

**Fig. 2.** Samples for the SA-SLA approach

allowing a “mixed economy” environment where annotated and non annotated SLAs can co-exist.

This work proposes a specification for semantic annotations in SLA files called Semantic Annotations for Service Level Agreement (SA-SLA), which is based on the Semantic Annotations for Web Service Description Language (SA-WSDL) [11]. SA-SLA provides a description format extending current WS-Agreement specification [2] with semantic annotations in order to provide dynamically linkable domain vocabularies for the SLA format. Thus, SLA elements can be linked to semantic concepts defined externally to the SLA.

As well as SA-WSDL, this specification is comprised mainly of: i) model reference, which is an association between a SLA schema and a concept in some semantic model, and ii) schema mappings, specifying mappings between semantic data and XML [11].

Figure 2 shows how it is possible to link elements in a SLA template file such as SLA Parameters or metrics (MaxWeight) with concepts belonging to the OWL QoS ontology (WeightCapacity, Kilogram) by using the SA-SLA specification. As a result, we have a SLA template annotated semantically.

Although we use the same elements as SA-WSDL, our approach is different since the annotated elements are different. The former is using WSDL elements while SA-SLA annotates WS-Agreement and WSLA elements. Concretely, these elements are service descriptions (terms related to functional properties), SLA parameters, metrics and service properties, since they constitute the basic vocabulary inside the SLA message structure.

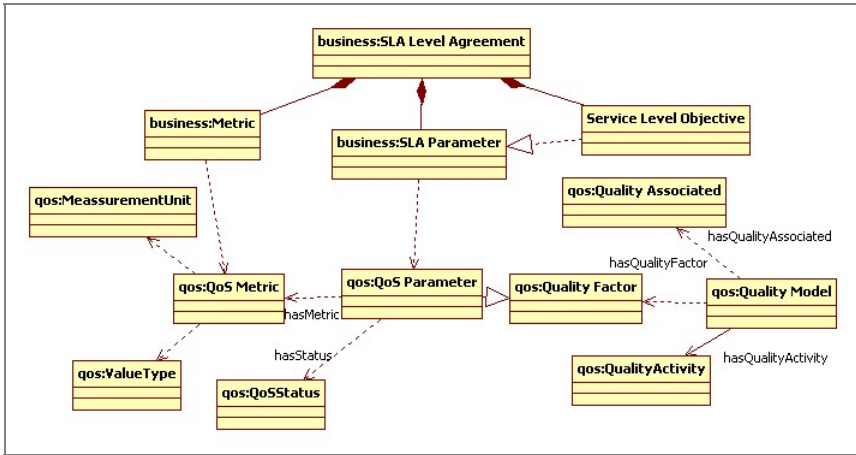


Fig. 3. Key concepts of our conceptual model

## 4 SA-SLA Related Ontologies

The SA-SLA allows to link SLA elements with concepts from a conceptual model formalized as ontologies. Thus, we also provide a common conceptual model for SLAs, which can be used to annotate SLA offers and bids, and which can be extended in local repositories according to the parties' requirements. The common conceptual model used in SA-SLA is mainly composed of concepts belonging to the core BREIN ontologies [6]; the business ontology and QoS ontology, whose main elements are summarized in Figure 3.

The business ontology enables the representation of SLAs as OWL instances in the sense of WS-Agreement and WSLA specifications. The emphasis is made on the semantic representation of Service Level Objectives (SLOs). The internals of a SLO are left undefined in WS-Agreements. Thus, they are modeled according to WSLA (as under SLA parameter), and include the name and value of the quality attribute, plus the metric used to calculate the value. Moreover, the SLA parameter is related to the QoS Parameter in the QoS ontology.

The QoS ontology, which is also part of the business ontology, collects the metrics and quality attributes to be used in SLOs. The basic concepts are taken from the quality model defined by OASIS in the Web Services Quality Model (WSQM) specification [12]. WSQM complements existing SLA-related specifications with a general view on quality related roles, processes and attributes. WSQM uses the term Quality Factor for QoS parameters and further categorizes it into sub-factors and layers concerning the user's view, interoperability and management. In our ontology, each QoSParameter is associated with a Metric characterized by ValueType (float, integer, boolean, etc.), a Value and a MeasurementUnit (e.g. euro, kB, ms). Finally, the QoSParameter can have several statuses depending on if it is requested by a customer, or offered by a provider, etc.

Involved parties can extend the common ontology to consider their internal requirements forming local ontologies. These local ontologies extend the common ontology



with locally used parameter types and also with local technological knowledge, which enables a better understanding of SLA requests by the provider in terms of its own local infrastructure. Service providers can add here the definition of locally used QoS parameters, metrics or measurement units. They can also add descriptions about local environment, such as available resource types and their parameters, licenses or platform dependencies. Furthermore, the mapping of received SLAs into the local environment can be supported with such local ontologies, either by new instances (such as conversion rates) or by conversion rules. In the next sections we present examples of these rules and demonstrate that the approach works on available Semantic Web technologies and it provides an adaptable solution without changing program code.

## 5 Application of SA-SLA for Negotiation

Based on WS-Negotiation [4], our negotiation approach follows the three-layered approach (Figure 1), so that it provides a protocol-independent solution on the level of Negotiation Message and a practical framework on the level of Negotiation Decision Making. For the Negotiation Message layer initially a merge of WS-Agreement [2] and WSLA [3] is used, allowing rich SLA definition.

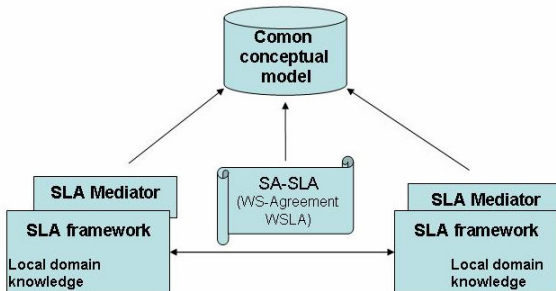


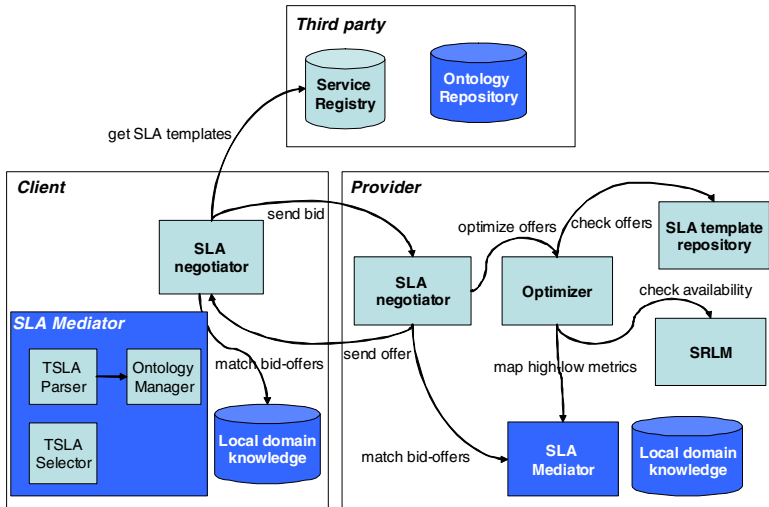
Fig. 4. SLA negotiation framework enhanced to use SA-SLA

The application of the framework for SLA negotiation is depicted on Figure 4. The common part contains the SA-SLA specification defining how to annotate SLA files and the common conceptual model expressed as an ontology. The model consists of a generic quality model and the structure of SLA documents as well as metrics and measurement units for SLA parameters.

The parties involved in SLA negotiation usually have their existing frameworks supporting SLA management. The negotiation may take place between customer and provider and also between two providers (outsourcing scenario), by exchanging SA-SLA files. The role of the SLA Mediator component is to import received SLAs into the local SLA model, which is used by the SLA framework. To achieve this, the SLA Mediator combines the common model with local knowledge.

This approach allows to keep the original standard SLA formats, and to explain its elements semantically through semantic annotations in SLA template files. The

ontologies used for annotations provide the common language of SLA negotiations, while the XML format files provide the common format or syntax. By this current approach, interoperability between entities involved in SLA negotiation is improved in a backward compatible way, so that components non aware of semantic annotations can continue working with SLA files. Furthermore, it is a lightweight and scalable approach, where participants can choose between various levels of adaptation to SA-SLAs.



**Fig. 5.** Concrete implementation of the SLA negotiation framework

In the following we present the concrete implementation of the previously described generic approach on the basis of an existing negotiation framework, which is represented in Figure 5 with lighter shade. The framework is composed of the following components; the SLA Negotiator in the customer side which acts on behalf of the customer to achieve the agreement with the provider. It has to both create bids and evaluate the offers received from providers. The SLA Negotiator in the Provider side processes incoming SLA requests from customers. The SLA Negotiator has been implemented with the Globus Toolkit 4 middleware. It follows the principles and designs of the WS-Agreement and provides the means of optional agent usage for negotiation. With this the possibility of going away from the traditional one-phase commit protocol to multiphase negotiation (e.g. with FIPA Iterated Contract Net Protocol [13]) is given. Moreover, it optimizes the offers sent using the Optimizer component, which is able to map the high level terms of the SLA Offer to the low level infrastructure level (checking the resources availability with the Semantic Resource Lifecycle Manager, SRLM) and available templates in the repository).

The base framework is extended with new components represented in Figure 5 by the darker shade. A third party hosts the common conceptual model in the Ontology Repository, which stores the ontologies used to annotate the SLA templates files. With that it provides the semantic model in charge of improving the interoperability.

Each of the entities involved in negotiation has a Local Domain Knowledge, which includes all business and negotiation rules and enterprise knowledge involved in the negotiation. Finally, the SLA Mediator matches SA-SLA requests and offers, by importing SLAs into the local SLA model supported and used within the framework.

The SLA Mediator allows for matching customer SLA requests with providers' offers. It is implemented in Java, using Jena as the semantic platform. First, the TSLA Parser parses the incoming SLA files performing two kinds of matchmaking depending on whether the metrics are annotated or not: i) syntactically, which involves a keyword search and ii) semantically, which interacts with the ontology manager in order to match the semantic QoS metrics with the customer' requests.

The Parser creates the OWL representation of the incoming SLA by extending the local domain ontology with new facts through the Jena OntModel API [15].

The Ontology Manager (Jena) allows the management of the local domain ontology. Pellet [16] is run inside Jena to provide OWL DL inferencing and SWRL support. The domain rules are formalized using SWRL [17]. The TSLA Selector provides a comparison (ranking) of local SLA templates (offers) and the incoming customer bid. This ranking is a number between 0 and 1 representing how well the template fits the request. This outcome is then used by the SLA Negotiator to create the offer and send it as a response to the customer.

## 6 Experimentation

The scenario chosen to demonstrate the applicability of the approach is obtained from the Airport Scenario of the BREIN project [6]. This scenario tries to demonstrate the coordination mechanism for ad-hoc resource allocation.

Different resource providers (ground-handling companies) are involved in the collaboration value chain of the airport ground handling. Among the different providers and customers an agreed SLA is required. Figure 6 shows examples for the parameters and metrics the customers and providers may use. It is apparent that the parameters are related, but they are not uniformly expressed, which demonstrates the heterogeneity problems in terms of:

- Different **service attribute and value abstractions**: e.g. “capacity” (abstract) vs. “max. number of passengers” (concrete), “jet-propelled aircraft” (abstract) vs. “B747-400” (concrete)
- Different **service attribute definitions**: e.g. “price” without or with VAT or “price” based on distance from gate to position vs. flat distance,
- Different **service attribute names**: e.g. “max. passengers”, “passengers, max.”, “passengers no.”, and the problem of multi-linguality in names,
- Different **units of measurement (UOM) for service attributes**: e.g. kilograms, tons, pounds, park position code vs. GPS coordinate.

In order to improve interoperability between all involved parties, providers and customers define the SLA template files relating their parameters and metrics to a common conceptual model by following SA-SLA (cf. Figure 2). Both the customer's requests and providers' templates are formalized as WS-Agreement (plus WSLA elements) extended by the SA-SLA specification. Coming back to the example

provided in Figure 6, there are two providers with associated templates, which express the metrics and parameters commented in the example by using the SA-SLA specification. On the other side, the customer expresses the SLA customer request by using the same format. In order to formalize the local domain knowledge, the provider creates a set of rules by using concepts from the common ontology.

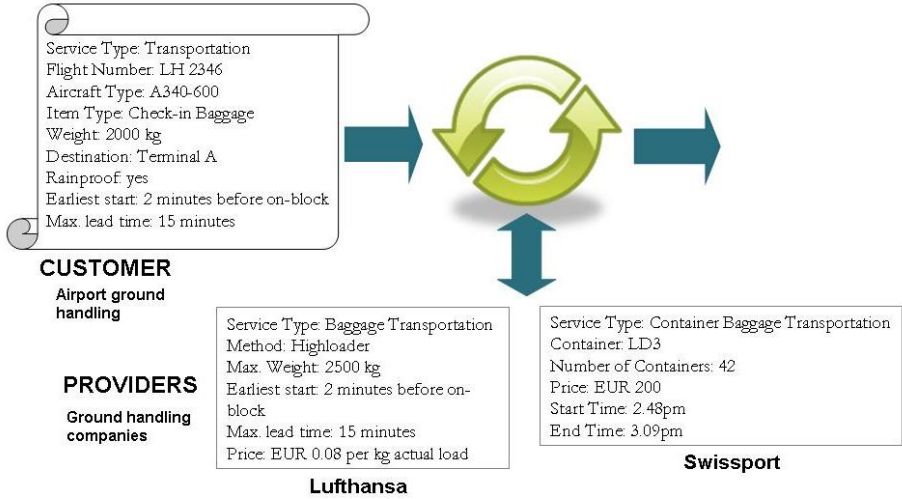


Fig. 6. Examples for SLA bid and offers

These SA-SLA files are then applied in the SLA negotiation, where the semantic annotations in SA-SLA files are used, so that the provider can understand the customer requirements and search for a relevant offer according to the customer bid. Steps required in the process include the following:

*Create SLA instances:* All the SLA templates in the provider side are stored as OWL instances in the local domain knowledge repository.

*Receive bid:* when receiving a bid the (provider) SLA Negotiator evaluates if it can provide an offer to satisfy the bid. For this, it calls the SLA Mediator.

*Bid conversion:* The customer SLA request is parsed by the SLA Mediator and then converted into an OWL representation. An OWL instance represents the SLA request, containing instances for each SLO.

*SLO translation:* The SLOs of the bid are translated into the corresponding local SLOs using local metrics and units of measurement. In this phase SWRL rules are used to fill 'localized' instance properties. An example given here is a generic conversion rule where conversion 'rates' are defined in OWL:

```
qos:WeightCapacityMetric(?param) ^ qos:hasUnit(?param, t) ^
qos:parameterValue(?param, ?v1) ^ swrlb:multiply(?v2,?v1,1000) →
hasLocalUnit(?param, kg) ^ hasLocalValue(?param, ?v2)
```

Naturally, the provider can define additional, custom rules to provide more complex metric conversions.

*Matching phase:* Available offers need to be related to the request in this step. First, a new Match instance is created for each offer template using Java code. Then, comparable pairs of SLOs in request and potential offers (from templates) are detected using an SWRL rule:

$$\begin{aligned} & \text{qos:QualityModel(?b)} \wedge \text{qos:hasQualityFactor(?b, ?p1)} \wedge \text{qos:hasMetric(?p1, ?m1)} \wedge \\ & \text{hasLocalMUnit(?p1, ?u1)} \wedge \text{QualityModelTemplate(?t)} \wedge \text{qos:hasQualityFactor(?t, ?p2)} \wedge \\ & \text{qos:hasMetric(?p2, ?m1)} \wedge \text{qos:hasMUnit(?p2, ?u1)} \wedge \text{differentFrom(?p1, ?p2)} \\ & \rightarrow \text{isMatchedTo(?p1, ?p2)} \end{aligned}$$

The pairs identified by “isMatchedTo” are evaluated one by one, and accepted if the SLO of the offer is stronger than the SLO of the request, with rules like this one:

$$\begin{aligned} & \text{isMatchedTo(?p1, ?p2)} \wedge \text{hasLocalValue(?p1, ?v1)} \wedge \text{qos:hasOperation(?p1,} \\ & \text{qos:greaterEqual)} \wedge \text{qos:parameterValue(?p2, ?v2)} \wedge \text{qos:hasOperation(?p2, qos:greaterEqual)} \\ & \wedge \text{swrlb:greaterThanOrEqual(?v2, ?v1)} \\ & \rightarrow \text{isFitting(?p1, ?p2)} \end{aligned}$$

This is again a place for local adaptation by rules; for example the provider may define a 5% threshold of SLO comparison instead of strict  $\leq$  or  $\geq$  comparisons.

*Template Scoring:* The previous evaluation is used to assign a numeric, normalized score to each SLA template, which represents the fitness of the template for the customer request.

*Respond with offer:* Finally, the SLA Negotiator collects the ranked list of templates, takes the one with the best score, and creates an offer from it, which is sent to the customer.

## 7 Conclusions

The paper presents a novel approach towards the improvement of SLA negotiation, which was driven from the need to improve the flexibility of existing SLA negotiation protocols by infusing semantic annotations in a non-intrusive manner. The annotation technique adopted in the proposed SA-SLA specification extends the idea introduced in SAWSDL. Our experimentation with SA-SLA and testing with existing SLA negotiation components proved its backwards compatibility which was the first objective of our approach. The second objective of our experimentation was to demonstrate how semantic interoperability issues found in the annotated WS-Agreement documents were automatically dealt with by the SLA mediator. Appropriate improvements to the SLA negotiation component were made in order to allow the use of the semantic annotations while new ontologies and rules were developed to capture the conceptual model related to the negotiation of logistics services. The conceptual model for our experimentation was derived from the analysis of real requirements from ground-handling services in an airport. The modified SLA negotiation component successfully used semantic technologies to overcome the semantic interoperability issues in an automated fashion.

## Acknowledgment

This work has been supported by the BREIN project (<http://www.gridsforbusiness.eu>) and has been partly funded by the European Commission under contract FP6-034556.

## References

1. Mitchell, B., Mckee, P.: SLAs a Key Commercial Tool Innovation and the Knowledge Economy: Issues, Applications. Case Studies (2005)
2. GRAAP-WG: Web Services Agreement Specification (WS-Agreement) (March 14, 2007), <http://www.ogf.org/documents/GFD.107.pdf>
3. Ludwig, H., Keller, A., Dan, A., King, R.P., Franck, R.: Web service level agreement (WSLA) language specification, <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>
4. Hung, P., Li, H., Jeng, J.-J.: WS-Negotiation: An Overview of Research Issues. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, HICSS 2004 (2004)
5. TrustCom consortium: The TrustCom Project, <http://www.eu-trustcom.com/>
6. BREIN consortium: The BREIN Project, <http://www.eu-brein.com/>
7. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Pruyne, J.: Web Services Agreement Negotiation Specification (WS-AgreementNegotiation) (2004)
8. Sandholm, T.W.: Distributed rational decision making. Multiagent systems: a modern approach to distributed artificial intelligence, pp. 201–258. MIT Press, Cambridge (1999)
9. Frankova, G., Malfatti, D., Aiello, M.: Semantics and Extensions of WS-Agreement. Journal of Software 1(1) (July 2006)
10. Oldham, N., Verma, K., Sheth, A., Hakimpour, F.: Semantic WS-agreement partner selection. In: Proceedings of the 15th international conference on World Wide Web, pp. 697–706. ACM, New York (2006)
11. Kopecky, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Computing 11(6), 60–67 (2007)
12. Kim, E.: OASIS Quality Model for Web Services. Version 2.0 (2005)
13. FIPA Iterated Contract Net Interaction Protocol Specification, <http://www.fipa.org/specs/fipa00030/SC00030H.html>
14. Munoz, H., Kotsiopoulos, I., Vaquero, L.M., Rodero, L.: Enhancing Service Selection by Semantic QoS. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 565–577. Springer, Heidelberg (2009)
15. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations, New York, NY, USA, pp. 74–83 (2004)
16. Evren, S., Bijan, P., Bernardo Cuenca, G., Aditya, K., Yarden, K.: Pellet: A practical OWL-DL reasoner. Web Semant. 5(2), 51–53 (2007)
17. O’connor, M., Knublauch, H., Tu, S., Grosz, B., Dean, N., Grosso, W., Musen, M.: Supporting rule system interoperability on the semantic web with SWRL, pp. 974–986 (2005)

# Runtime Prediction of Service Level Agreement Violations for Composite Services

Philipp Leitner<sup>1</sup>, Branimir Wetzstein<sup>2</sup>, Florian Rosenberg<sup>3</sup>, Anton Michlmayr<sup>1</sup>, Schahram Dustdar<sup>1</sup>, and Frank Leymann<sup>2</sup>

<sup>1</sup> Distributed Systems Group  
Vienna University of Technology  
Argentinierstrasse 8/184-1  
A-1040, Vienna, Austria  
`lastname@infosys.tuwien.ac.at`

<sup>2</sup> Institute of Architecture of Application Systems  
University of Stuttgart  
Stuttgart, Germany

`lastname@iaas.uni-stuttgart.de`

<sup>3</sup> CSIRO ICT Centre  
GPO Box 664  
Canberra ACT 2601, Australia  
`florian.rosenberg@csiro.au`

**Abstract.** SLAs are contractually binding agreements between service providers and consumers, mandating concrete numerical target values which the service needs to achieve. For service providers, it is essential to prevent SLA violations as much as possible to enhance customer satisfaction and avoid penalty payments. Therefore, it is desirable for providers to predict possible violations before they happen, while it is still possible to set counteractive measures. We propose an approach for predicting SLA violations at runtime, which uses measured and estimated facts (instance data of the composition or QoS of used services) as input for a prediction model. The prediction model is based on machine learning regression techniques, and trained using historical process instances. We present the basics of our approach, and briefly validate our ideas based on an illustrative example.

## 1 Introduction

In service-oriented computing [1], finer-grained basic functionality provided using Web services can be composed to more coarse-grained services. This model is often used by Software-as-a-Service providers to implement value-added applications, which are built upon existing internal and external Web services. Very important for providers and consumers of such services are Service Level Agreements (SLAs), which are legally binding agreements governing the quality that the composite service is expected to provide (Quality of Service, QoS). SLAs contain Service Level Objectives (SLOs), which are concrete numerical target values (e.g., “maximum response time is 45 seconds”). For the provider

it is essential to not violate these SLOs, since typically violations are coupled with penalty payments. Additionally, violations can negatively impact service consumer satisfaction. Therefore, it is vitally important for the service provider to be aware of SLA violations, in order to react to them accordingly.

Typically, SLA monitoring is done *ex post*, i.e., violated SLOs can only be identified after the violation happened. While this approach is useful in that it alerts the provider to potential quality problems, it clearly cannot directly help preventing them. In that regard an *ex ante* approach is preferable, which allows to predict possible SLA violations before they have actually occurred. The main contribution of this paper is the introduction of a general approach to prediction of SLA violations for composite services, taking into account both QoS and process instance data, and using estimates to approximate not yet available data. Additionally, we present a prototype implementation of the system and an evaluation based on an order processing example. The ideas presented here are most applicable for long-running processes, where human intervention into problematic instances is possible. Our system introduces the notions of checkpoints (points in the execution of the composition where prediction can be done), facts (data which is already known in a checkpoint, such as the response times of already used services) and estimates (data which is not yet available, but can be estimated). Facts and estimates can refer to both typical QoS data (e.g., response times, availability, system load) and process instance data (e.g., customer identifiers, ordered products). Our implementation uses regression classifiers, a technique from the area of machine learning, to predict concrete SLO values.

## 2 Illustrative Example

To illustrate the ideas presented in this paper we will use a simple purchase order scenario (see Figure 2 below). In this example there are a number of roles to consider (reseller, customer, banking service, shipping service, and two external suppliers). Whenever the reseller service receives an order from the customer, it first checks if all ordered items are available in the internal stock. If this is not the case, it checks if the missing item(s) can be ordered from Supplier 1, and, if this is not the case, from Supplier 2. If both cannot deliver the order has to be cancelled, otherwise the missing items are ordered from the respective supplier. When all ordered items are available she will (in parallel) proceed to charge the customer using the banking service and initialize shipment of the ordered goods (using the Shipping Service). Please refer to [2] for more details on this case.

In this case study, the reseller has an SLA with its customers, with an SLO specifying that the end-to-end response time of the composition cannot be more than a certain threshold of time units. For every time the SLO is violated the customer is contractually entitled a discount for the order. Note that even though our explanations in this paper will be based on just one single SLO, our approach can be generalized to multiple SLOs. Additionally, even though we present our approach based on a numerical SLO, our ideas can be also applied to estimation of nominal objectives.



### 3 Predicting SLA Violations

In this section we present the core ideas of our approach towards prediction of SLA violations. Generally, the approach is based on the idea of predicting concrete SLO values based on whatever information is already available at a concrete point in the execution of a composite service. We distinguish three different types of information. (1) **Facts** represent data which is already known at prediction time. Typical examples of facts are the QoS of already used services, such as the response time of a service which has already been invoked in this execution, or instance data which has either been passed as input or which has been generated earlier in the process execution. (2) **Unknowns** are the opposites of facts, in that they represent data which is entirely unknown at prediction time. Oftentimes, instance data which has not yet been produced falls into this category. If important factors are unknown at prediction time the prediction quality will be very bad, e.g., in our illustrative example a prediction cannot be accurate before it is known whether the order can be delivered from the reseller’s internal stock. (3) **Estimates** are a kind of middle ground between facts and unknowns, in that they represent data which is not yet available, but can be estimated. This is often the case for QoS data, since techniques such as QoS monitoring [3] can be used to get an idea of e.g., the response time of a service before it is actually invoked. Estimating instance data is more difficult, and generally domain-specific.

The overall architecture of our system is depicted in Figure 1. The most important concept used is that the user defines **checkpoints** in the service composition, which indicate points in the execution where a prediction should be carried out. The exact point in the execution model which triggers the checkpoint is called the **hook**. Every checkpoint is associated with one **checkpoint predictor**. Essentially, the predictor uses a function taking as input all facts which are already available in the checkpoint, and, if applicable, a number of estimates of not yet known facts, and produces a numerical estimation of the SLO

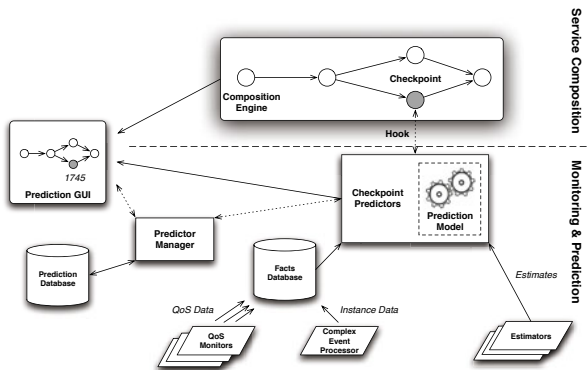


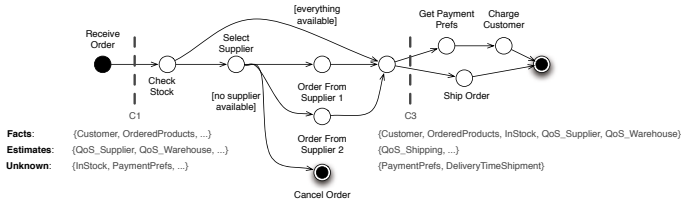
Fig. 1. Overall System Architecture

value(s). This function is generated using machine learning techniques. We refer to this function as the **prediction model** of a checkpoint predictor. Facts are retrieved from a **facts database**, which is filled using a number of **QoS monitors** (which provide QoS data) and a **Complex Event Processing (CEP)** engine (which extracts and correlates the instance data, as emitted by the process engine). A detailed discussion of our event-based approach to monitoring is out of scope of this paper, but can be reviewed in related work [2,4]. **Estimators** are a generic framework for components which deliver estimates. Finally, the prediction result is transferred to a graphical user interface (**prediction GUI**), which visualizes the predicted value(s) for the checkpoint. A **predictor manager** component is responsible for the lifecycle management of predictors, i.e., for initializing, destroying and retraining them. Additionally, predictions are stored in a **prediction database** to be available for future analysis.

### 3.1 Checkpoint Definition

At design-time, the main issue is the definition of checkpoints in the composition model. For every checkpoint, the following input needs to be provided: (1) The hook, which defines the concrete point in the execution that triggers the prediction, (2) a list of available facts, (3) a list of estimates, and the estimator component as well as the parameters used to retrieve or calculate them, (4) the retraining strategy, which governs at which times a rebuilding of the prediction model should happen, and (5) as a last optional step, a parameterization of the machine learning technique used to build the prediction model. After all these inputs are defined the checkpoint is deployed using the predictor manager, and an initial model is built. For this a set of historical executions of the composite service need to be available, for which all facts (including those associated with estimates) have been monitored. If no or too little historical data is available the checkpoint is suspended by the predictor manager until enough training data has been collected. The amount of data necessary is case-specific, since it vastly depends on the complexity of the composition. We generally use the Training Data Correlation as a metric for evaluating the quality of a freshly trained model (see below for a definition), however, a detailed discussion of this is out of scope of this paper. After the initial model is built the continuous optimization of the predictor is governed by the predictor manager, according to the retraining strategy. Finally, the checkpoint can be terminated by the user via the prediction GUI. We will now discuss these concepts in more depth.

*Hooks.* Hooks can be inserted either before or after any WS-BPEL activity (for instance, an *Invoke* activity). Generally, there is a tradeoff to take into account here, since early predictions are usually more helpful (in that they rather allow for corrections if violations are predicted), but also less accurate since less facts are available and more estimates are necessary. Figure 2 depicts the (simplified) example from Section 2, and shows two possible checkpoints. In  $C_1$  the only facts available are the ones given as input to the composition (such as a customer identifier, or the ordered products). Some other facts (mainly QoS metrics) can already



**Fig. 2.** Illustrative Example With Possible Checkpoints

be estimated, however, other important information, such as whether the order can be served directly from stock, is simply unavailable in  $C_1$ , not even as an estimate. Therefore, the prediction cannot be very accurate. In checkpoint  $C_3$ , on the other hand, most of the processes important raw data is already available as facts, allowing for good predictions. However, compared to  $C_1$ , the possibilities to react to problems are limited, since only the payment and shipping steps are left to adapt (e.g., a user may still decide to use express shipping instead of the regular one if a SLA violation is predicted in  $C_3$ ). Finding good checkpoints at which the prediction is reasonably accurate and still timely enough to react to problems demands for some domain knowledge about influential factors of composition performance. Dependency analysis as discussed in [4] can help providing this crucial information. Dependency analysis is the process of using historical business process instance data to find out about the main factors which dictate the performance of a process. When defining checkpoints, a user can assume that the majority of important factors of influence need to be available as either facts or at least as good estimates in order to achieve accurate predictions.

*Facts and Estimates:* Facts represent all important information which can already be measured in this checkpoint. This includes both QoS and instance data. Note that the relationship between facts and the final SLO values does not need to be known (e.g., a user can include instance data such as user identifiers or ordered items, even if she is not sure if this has any relevance for the SLO). However, dependency analysis can again be used to identify the most important facts for a checkpoint. Additionally, the user can also define estimates. In the example above, in  $C_1$  the response time of the warehouse service is not yet known, however, it can e.g., be estimated using a QoS monitor. Since estimating instance data is inherently domain-specific, our system is extensible in that more specific estimators (which are implemented as simple Java classes) can be integrated seamlessly. Estimates are linked to facts, in the sense that they have to represent an estimation of a fact which will be monitorable at a later point.

*Retraining Strategy:* Generally, the prediction model needs to be rebuilt whenever enough new information is available to significantly improve the model. The retraining strategy is used to define when the system should check whether rebuilding the prediction model is necessary. We currently support the following retraining strategies: (1) periodic retraining (retraining is done in fixed intervals),

(2) instance-based retraining (retraining is done whenever a certain amount of new instances have been monitored), (3) on demand retraining (retraining only on explicit user demand), (4) on error retraining (retraining when the mean prediction error  $\bar{e}$  exceeds a given threshold), and (5) custom retraining (retraining based on user-defined conditions).

*Prediction Model Parameterization:* A user can also define the machine learning technique that should be used to build the prediction model. This is done by specifying an algorithm and the respective parameterization for the WEKA toolkit<sup>1</sup>, an open source machine learning toolkit which we internally use in our prototype implementation. In this way the prediction quality can be tuned by a machine learning savvy user, however, we also provide a default configuration which can be used out of the box.

### 3.2 Run-Time Prediction

At runtime, the prediction process is triggered by lifecycle events from the WS-BPEL engine. These are events emitted by some engines, which contain lifecycle information about the service composition. Our approach is based on these events, therefore, a WS-BPEL engine which is capable of emitting these events is a preliminary of our approach. When checkpoints are deployed we use the hook information to register respective event listeners. For instance, for a checkpoint with the hook “After invoke CheckStock” we generate a listener for `ActivityExecEndEvents` which consider the invoke activity “CheckStock”. We show the sequence of actions which is triggered as soon as such an event is received in Figure 3.

After being triggered by a lifecycle event the checkpoint predictor first extracts some necessary correlation information from the event received. This includes the process instance ID as assigned by the composition engine, the instance start time (i.e., the time when the instance was created) and the timestamp of

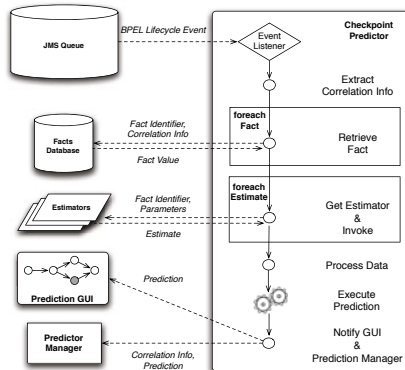


Fig. 3. Runtime View On Checkpoint Predictors

<sup>1</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

the event. This information is necessary to be able to retrieve the correct facts from the facts database, which is done for every fact in the next step (e.g., in order to find the correct fact “CustomerNumber” for the current execution the process instance ID needs to be known). When all facts have been gathered, the predictor also collects the still missing estimates. For this, for every estimate the predictor instantiates the respective estimator component (if no instance of this estimator was available before), and invokes it (passing all necessary parameters as specified in the checkpoint definition). The gathered facts and estimates are then converted into the format expected by the prediction model (in the case of our prototype, this is the WEKA Attribute-Relation File Format ARFF<sup>2</sup>), and, if necessary, some data cleaning is done. Afterwards, the actual prediction is carried out by passing the gathered input to the prediction model producing a numerical estimation of the SLO value. This prediction is then passed to the prediction GUI (for visualization) and the prediction manager.

Note that the “intelligence” that actually implements the prediction of the SLO values is encapsulated in the prediction model. Since we (usually) want to predict numerical SLO values the prediction model needs to be a regression model. We consider the regression model to be a black-box function which takes a list of numeric and nominal values as input, and produces a numeric output. Generally, our approach is agnostic of how this is actually implemented. In our work we use multilayer perceptrons (a powerful variant of neural networks) to implement the regression model. Multilayer perceptrons are trained iteratively using a back-propagation technique (maximization of the correlation between the actual outcome of training instances and the outcome that the network would predict on those instances), and can (approximately) represent any relationship between input data and outcome.

### 3.3 Evaluation of Predictors

Another important task of the prediction manager is quality management of predictors, i.e., continually supervising how predictions compare to the actual SLO values once the instance is finished. Generally, we use three different quality metrics to measure the quality of predictions in checkpoints. The first metric, *Training Data Correlation* ( $corr = \frac{cov(P,M)}{\sigma_p \sigma_m}$ ), is a standard machine learning approach to evaluating regression models. We use it mainly to evaluate freshly generated models, when no actual predictions have yet been carried out. This metric is defined as the statistical correlation between all training instance outcomes and the predictions that the model would deliver for these training instances. The definition given is the standard statistical definition of the correlation coefficient between a set of predicted values  $P$  and a set of measured values  $M$ . However, note that this metric is inherently overconfident in our case, since during training all estimates are replaced for the facts that they estimate (i.e., the training is done as if all estimates were perfect). Therefore, we generally measure the

<sup>2</sup> <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>

prediction error later on, when actual estimates are being used. However, a low training data correlation is an indication that important facts are still unknown in the checkpoint, i.e., that the checkpoint may be too early. The actual quality of the prediction is measured using the *Mean Prediction Error*  $\bar{e} = \frac{\sum_{i=0}^n |m_i - p_i|}{n}$ , which is the average difference between predicted and monitored values.  $n$  is the total number of predictions,  $p_i$  is a predicted value, and  $m_i$  is the measured value to prediction  $p_i$ . Finally, we use the *Prediction Error Standard Deviation* ( $\sigma = \sqrt{\frac{\sum_{i=0}^n (e_i - \bar{e})^2}{n}}$ ) to describe the variability of the prediction error (i.e., high  $\sigma$  essentially means that the actual error for an instance can be much lower or higher than  $\bar{e}$ ).  $e_i$  is the actual prediction error for a process instance ( $m_i - p_i$ ). These metrics are mainly used to give the user an estimation of how trustworthy a given prediction is. Additionally, the **on error** retraining strategy triggers on  $\bar{e}$  exceeding a certain threshold.

### 4 Experimentation

In order to provide a first validation of the ideas presented we have implemented the illustrative example as discussed in Section 2, and ran some experiments using our prototype tool. All experiments have been conducted on a single test machine with 3.0 GHz and 32 GByte RAM, running under Windows Server 2007 SP1.

For reasons of brevity we focus on prediction accuracy in this paper. To measure accuracy, we have implemented five checkpoints in the illustrative example (see top of Figure 4): C1 is located directly after the order is received, C2 after the internal warehouse is checked, C3 after eventual orders from external suppliers have been carried out, C4 during the payment and shipment process, and finally C5 when the execution is already finished. In each of those checkpoints we have trained a prediction model using 1000 historical process instances, and

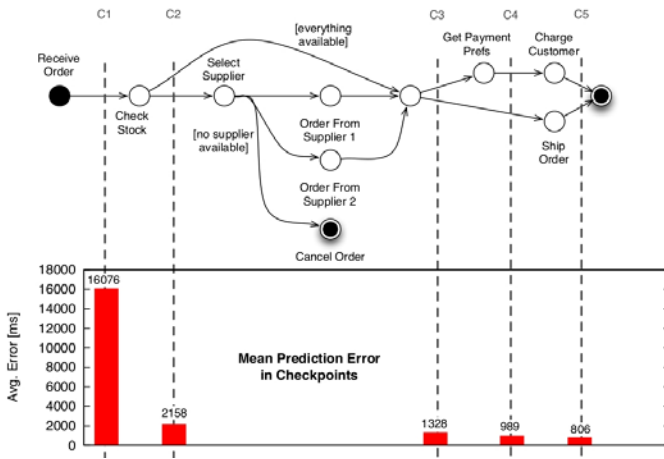


Fig. 4. Prediction Error in Checkpoints

have specified all available data as facts. For not yet available QoS metrics we have used the average of all previous invocations as estimate. Missing instance data has been treated as unknown. We have used each of those checkpoints to predict the outcome of 100 random executions, and calculated  $\bar{e}$ . As expected,  $\bar{e}$  is decreasing with the amount of factual data available. In C1, the prediction is mostly useless, since no real data except the user input is available. However, in C2 the prediction is already rather good. This is mostly due to the fact that in C2 the information whether the order can be delivered directly from stock is already available. In C3, C4 and C5 the prediction is continually improving, since more actual QoS facts are available, and less estimates are necessary. Speaking in absolute values,  $\bar{e}$  in e.g., C3 is 1328 ms. Since the average SLO value in our illustrative example was about 16000 ms, the error represents only about 8% of the actual SLO value, which seems satisfactory. We have also carried out experiments to determine the time necessary to build prediction models and to do predictions at runtime. These results are not discussed here because of space restrictions.

## 5 Related Work

The work presented in this paper is complementary to the more established concept of SLA management. SLA management incorporates the definition and monitoring of SLAs, as well as the matching of consumer and provider templates [5,6]. In our work we add another facet to the more general area of SLA management, namely the prediction of SLA violations before they have actually occurred. Inherently, this prediction demands for some insight into the internal factors impacting composite service performance. In [7], the MoDe4SLA approach has been introduced to model dependencies of composite services on the used base services, and to analyze the impact that these dependencies have. Similarly, the work we have presented in [2] allows for an analysis of the impact that certain factors have on the performance of service compositions. SLA prediction as discussed in this paper has first been discussed in [8], which is based on some early work of HP Laboratories on SLA monitoring for Web services [9]. In [8], the authors introduced some concepts which are also present in our solution, such as the basic idea of using prediction models based on machine learning techniques, or the trade-off between early prediction and prediction accuracy. However, the authors do not discuss important issues such as the integration of instance and QoS data, or strategies for updating prediction models. Additionally, this work does not take estimates into account, and relatively little technical information about their implementation is publicly available. A second related approach to QoS prediction has been presented recently in [10]. In this paper the focus is on KPI prediction using analysis of event data. Generally, this work exhibits similar limitations as the work described in [8], however, the authors discuss the

influence of seasonal cycles on KPIs. This facet has not been examined in our work, even though seasons can arguably be integrated easily in our approach as additional facts.

## 6 Conclusions

In this paper we have presented an approach to runtime prediction of SLA violations. Central to our approach are checkpoints, which define concrete points in the execution of a composite service at which prediction has to be carried out, facts, which define the input of the prediction, and estimates, which represent predictions about data which is not yet available in the checkpoint. We use techniques from the area of machine learning to construct regression models from recorded historical data to implement predictions in checkpoints. Retraining strategies govern at which times these regression models should be refreshed. Our Java-based implementation uses the WEKA Machine Learning framework to build regression models. Using an illustrative example we have shown that our approach is able to predict SLO values accurately.

## Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement 215483 (S-Cube).

## References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. *IEEE Computer* 11 (2007)
2. Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Leymann, F., Dustdar, S.: Monitoring and Analyzing Influential Factors of Business Process Performance. In: *EDOC 2009: Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference* (2009)
3. Michlmayr, A., Rosenberg, F., Leitner, P., Dustdar, S.: Comprehensive QoS Monitoring of Web Services and Event-Based SLA Violation Detection. In: *MW4SOC 2009: Proceedings of the 4rd International Workshop on Middleware for Service Oriented Computing* (2009)
4. Wetzstein, B., Strauch, S., Leymann, F.: Measuring Performance Metrics of WS-BPEL Service Compositions. In: *ICNS 2009: Proceedings of the Fifth International Conference on Networking and Services*. IEEE Computer Society, Los Alamitos (2009)
5. Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web Services on Demand: WSLA-Driven Automated Management. *IBM Systems Journal* 43(1), 136–158 (2004)
6. Tomic, V., Pagurek, B., Patel, K., Esfandiari, B., Ma, W.: Management applications of the web service offerings language (wsol). *Information Systems* 30(7), 564–586 (2005)



7. Bodenstaff, L., Wombacher, A., Reichert, M., Jaeger, M.C.: Monitoring Dependencies for SLAs: The MoDe4SLA Approach. In: SCC 2008: Proceedings of the 2008 IEEE International Conference on Services Computing, pp. 21–29 (2008)
8. Castellanos, M., Casati, F., Dayal, U., Shan, M.C.: Intelligent Management of SLAs for Composite Web Services. In: Bianchi-Berthouze, N. (ed.) DNIS 2003. LNCS, vol. 2822, pp. 28–41. Springer, Heidelberg (2003)
9. Sahai, A., Machiraju, V., Sayal, M., Moorsel, A.P.A.v., Casati, F.: Automated SLA Monitoring for Web Services. In: Feridun, M., Kropf, P.G., Babin, G. (eds.) DSOM 2002. LNCS, vol. 2506, pp. 28–41. Springer, Heidelberg (2002)
10. Zeng, L., Lingenfelder, C., Lei, H., Chang, H.: Event-Driven Quality of Service Prediction. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 147–161. Springer, Heidelberg (2008)

# A Framework for Multi-level SLA Management<sup>\*</sup>

Marco Comuzzi<sup>1</sup>, Constantinos Kotsokalis<sup>2</sup>, Christoph Rathfelder<sup>3</sup>,  
Wolfgang Theilmann<sup>4</sup>, Ulrich Winkler<sup>4</sup>, and Gabriele Zacco<sup>5</sup>

<sup>1</sup> Eindhoven University of Technology, The Netherlands

<sup>2</sup> Dortmund University of Technology, Germany

<sup>3</sup> FZI Research Center for Information Technology, Karlsruhe, Germany

<sup>4</sup> SAP Research, Karlsruhe, Germany

<sup>5</sup> Fondazione Bruno Kessler, Povo (Trento), Italy

m.comuzzi@tue.nl, wolfgang.theilmann@sap.com

**Abstract.** Service-Oriented Architectures (SOA) represent an architectural shift for building business applications based on loosely-coupled services. In a multi-layered SOA environment the exact conditions under which services are to be delivered can be formally specified by Service Level Agreements (SLAs). However, typical SLAs are just specified at the customer-level and do not allow service providers to manage their IT stack accordingly as they have no insight on how customer-level SLAs translate to metrics or parameters at the various layers of the IT stack. In this paper we present a technical architecture for a multi-level SLA management framework. We discuss the fundamental components and interfaces in this architecture and explain the developed integrated framework. Furthermore, we show results from a qualitative evaluation of the framework in the context of an open reference case.

**Keywords:** Service Level Agreement (SLA), Service-Oriented Infrastructure (SOI), e-Contracting, Adaptive Infrastructures, Manageability, Non-Functional Properties.

## 1 Introduction

The paradigm of *Service-Oriented Architectures* (SOA) has changed the way for building IT-based systems [2]. Initially SOA was mainly applied to restructure the IT stack within an organisation. More recently it has also evolved as a common paradigm for cross-organisational service landscapes where services are considered as tradeable goods. Consequently, services operate under a strong business context where service customers can expect services to be provided under well-defined and dependable conditions and with clearly associated costs.

*Service Level Agreements* (SLAs) are a common way to formally specify the exact conditions (both functional and non-functional behaviour) under which services are or shall be delivered. However, the current SLAs in practice are just specified at the customer-level interface between a service provider and a

---

<sup>\*</sup> Presented by the authors on behalf of the SLA@SOI consortium [1].

service customer. Customer-level SLAs can be used by customers and providers to monitor whether the actual service delivery complies with the agreed SLA terms. In case of SLA violations, penalties or compensations can be directly derived. Customer-level SLAs do not allow service providers to either plan their IT landscapes according to possible, planned or agreed SLAs; nor do they allow understanding why a certain SLA violation might have occurred. The reason for this is that SLA guarantee terms might not be explicitly or directly related to actual performance metrics or configuration parameters. This makes it difficult for service providers to derive proper configuration parameters from customer-level SLAs and to assess (lower-level) monitoring metrics against customer-level SLAs. Overall, the missing relation between customer-level SLAs and (lower-level) metrics and parameters is a major hurdle for managing IT stacks in terms of IT planning, prediction or adjustment processes and in accordance with possible, planned or actual SLAs.

As part of the European Research project SLA@SOI [1], we developed the vision to use the paradigm of SLAs for managing a complete IT stack in correlation with customer-level SLAs which are agreed at the business level. This complies with the current technical trend to apply the paradigm of service-orientation across the complete IT stack, i.e. infrastructure/platform/software as a service, but also with the organisational trend in IT companies to organise different departments as service departments, providing infrastructure resources, middleware, applications or composition tools as a service. SLAs will be associated with multiple elements of the stack at multiple layers, e.g. SLAs for elements of the physical/virtual infrastructure, middleware, application and process-level. Such internal SLAs describe the contract between the lower-level entities and a higher-level entities consuming the lower ones. More precisely, the SLAs specify the required or agreed performance metrics but also the related configuration parameters.

This paper presents the detailed conception and implementation of a multi-level SLA management framework and it is built on a previous discussion of a purely conceptual architecture [3]. The remainder of this paper is organised as follows. Section 2 introduces the developed framework while Section 3 provides evaluation results in the context of a case study. Section 4 concludes with a brief summary and outlook. A full version of this paper including a discussion of related work can be found at [4].

## 2 SLA Management Framework

The design and implementation of our SLA management framework is based on the conceptualisation presented in [3]. There, we introduced the following core concepts:

- The four roles of service customer, software provider, service provider and infrastructure provider;
- The three layers of business, software, and infrastructure management;

- A service lifecycle model including design, negotiation, provisioning, operation, i.e. monitoring and adjustment, and decommissioning;
- Conception of relevant basic data store entities covering design-time and run-time data for the various layers and roles; and
- Conception of functional flows for the lifecycle phases of negotiation, provisioning, and operation.

In the following we now show a concrete technical architecture which implements the previous conceptualisation.

## 2.1 Technical Architecture

The technical architecture is presented in two steps. First we show how the architecture is split into different modules, each of them having a clear responsibility and interface. Second, we show how the envisioned SLA management procedures are realised in terms of scenario flows. The main challenges for such an architecture are (1) a clear separation of concerns between concepts that are highly interrelated and (2) a reasonable abstraction and flexibility that can easily support different target scenarios.

The technical architecture of the SLA management framework consists of six modules. Out of them, four modules are primarily responsible to orchestrate the core activities of the SLA negotiation, SLA provisioning, and SLA operation phase:

- The *negotiation module* is responsible for conducting multi-level SLA negotiation. It contains the design time repository (which allows for model-based performance predictions), the SLA template registry (that stores all the possible/offered templates), and it is also aware of the rules that support SLA translation between layers.
- The *provisioning module* co-ordinates the provisioning of SLA-specified services. It contains the SLA registry (storing the established SLAs), the software landscape (storing all software related artefacts for provisioning), and a scheduling component that makes sure temporal and other kinds of dependencies between services are honoured.
- The *monitoring module* is responsible for the decomposition of a SLA into monitorable rules and the analysis of a flow of incoming monitoring events against these rules. Eventual violations can be reported again on the level of SLAs.
- The *adjustment module* is responsible to collect all detected SLA violations or warnings and to trigger adjustment actions. It makes use of a set of adjustment patterns and a manageability model to achieve this.

The architecture is complemented by two modules which address the pure business / infrastructure view.

- The *eContracting module* is responsible for managing the business relationship with service customers. It contains the notion of *business offers* (products) and *policies* (how individual customers are processed in terms of pricing etc).

- The *infrastructure module* is responsible for managing all the infrastructure resources in a cloud-like manner. It contains the landscape of existing physical resources and possible virtualised counterparts.

Last, a technically motivated module realises an *event bus* to support asynchronous communication between the other modules.

Figure 1 provides an overview of the technical architecture, its modules, interfaces, and main relationships.

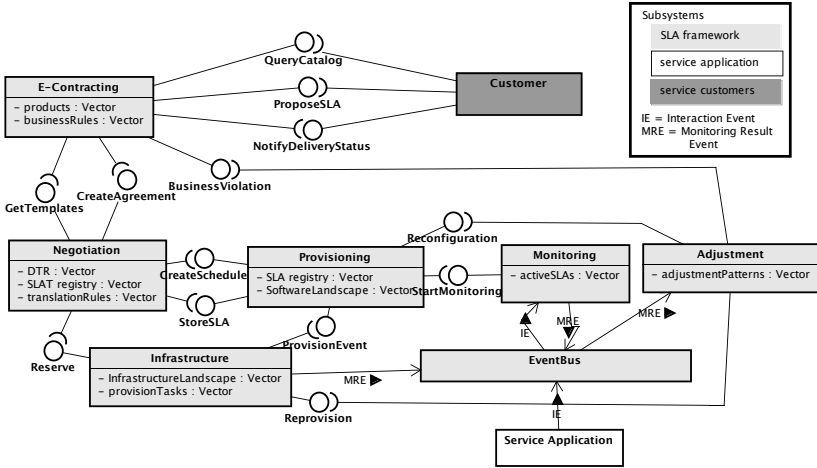
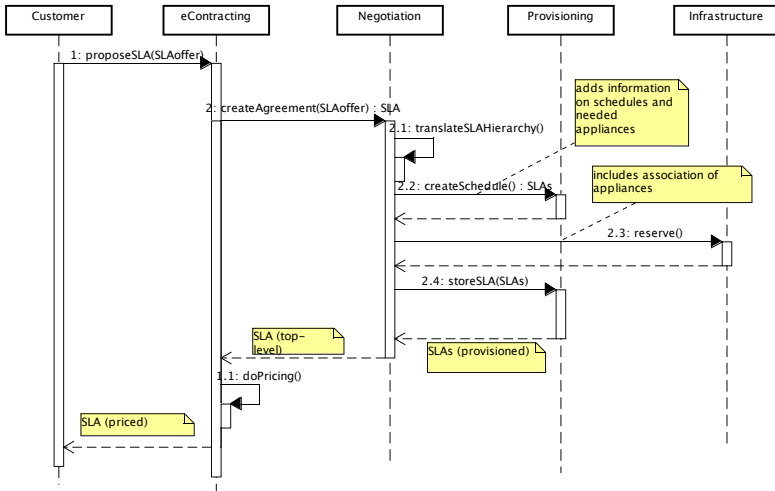


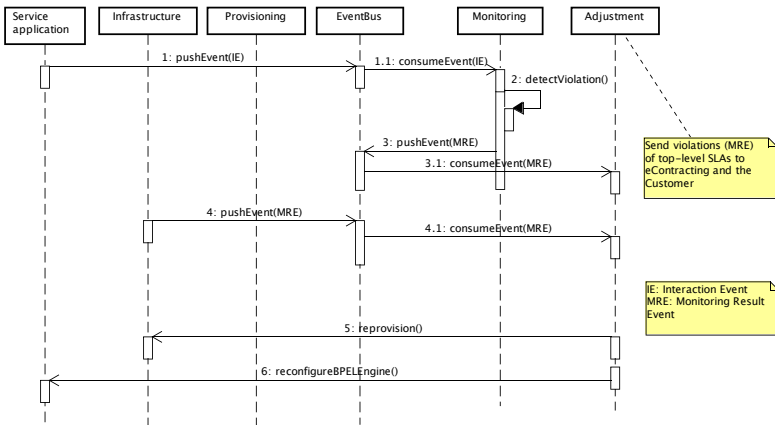
Fig. 1. Module Architecture

The dynamics of the SLA management framework are described along two scenarios. The first scenario concerns the negotiation and provisioning of the SLA, whereas the second scenario refers to runtime support, and, in particular, the monitoring, which detects SLA violations, and the adjustment, which reshapes services deployment configuration in relation to detected SLA violations.

The first scenario (see Figure 2) starts with the customer proposing an SLA offer to the SLA Framework, by invoking the eContracting module interface. The scenario implies an initial phase where the customer queries the eContracting module for retrieving the available product templates, which is not represented in Figure 2. The SLA offer proposed by the customer is accepted if it can be provisioned by the SLA framework according to the current status of software and infrastructure resources. The SLA offer is, therefore, forwarded to the Negotiation module, which creates the SLA hierarchy and issues requests for provisioning the software services and reserving the virtual machines required for execution of the services. If both requests are successful, the SLA is stored and then sent back to the customer. Before being sent back to the customer, the SLA is completed with pricing information. Note that we are using WS-Agreement for expressing SLAs and the usage of WS-Agreement has a counter intuitive side-effect: Since



**Fig. 2.** The negotiation and provisioning scenario



**Fig. 3.** The monitoring and adjustment scenario

there are no counter-offers foreseen in the specification, it is not possible for the initiator (here: the customer) to reject the proposed price. The upcoming WS-Agreement-Negotiation specification will resolve this issue.

The second scenario (see Figure 3) starts when an SLA becomes active and it is therefore submitted to the Monitoring module. Monitoring at the software layer is performed by checking the rules derived from the SLA against the *Interaction Events* (IE) produced by the services' execution environment, e.g. timestamped service operation calls and responses. When an SLA violation is detected, e.g. the average completion time of an operation exceeds the threshold reported in the

SLA, a *Monitoring Result Event* (MRE) is pushed on the Event Bus. Note that violations of terms at the infrastructure layer, e.g. abnormal CPU or memory usage, are directly detected by the Infrastructure module and pushed as MREs on the Event Bus. Violations (MREs) are read by the eContracting module, in order to be shown to the Customer, and by the Adjustment module. Adjustment, in particular, may decide to apply some corrective actions to overcome SLA violations (e.g. by reconfiguring software components or resizing hardware capacity).

## 2.2 Integration Foundations

A solid foundation for building a common approach for all modules is represented by the adoption of WS-Agreement as SLA modelling solution. WS-Agreement defines a signalling protocol for establishing SLAs. As such, it is domain-agnostic and provides generic data types to describe agreement templates and offers. WS-Agreement defines *Agreement Templates* to be roughly containers of terms, and constraints on those terms. A template consists of 4 top-level elements: *name*, *context*, *terms* and *creation constraints*, where terms are subdivided into *service description* terms and *guarantee* terms. Agreement documents themselves have the same structure, without containing any constraints section though.

For the terms, which form the content of the agreement depending on the domain at hand, the project developed a comprehensive core meta-model to be used as a basis by all modules. This core meta-model defines a number of essential constructs (e.g. temporal units, quantitative resource descriptions, etc) and binary operators for them. Using those constructs, it is possible to define the agreements' higher level terms, that can be understood in the same way by all components of our system.

The second foundation for a common approach of the modules is the adoption of a framework that eases the assembly of the architecture modules into the final platform. From a technological point of view, this is achieved using Spring [5], an open source application framework that provides core features services as well as advanced solutions for building complex applications.

## 3 Prototype Implementation and Case Study

The implemented framework has been evaluated against a reference application, the so-called Open Reference Case (ORC). This section briefly sketches the structure of this application, explains the related SLA hierarchy that we established, and provides results from a qualitative evaluation.

### 3.1 Open Reference Case

The ORC is a service-oriented software system supporting a retail chain scenario. The ORC extends the Common Component Modelling Example (CoCoME) [6], which represents a component-based trading system dealing with the various

aspects of handling sales at a supermarket. This includes the interaction at the cash desk with the customer, including product scanning and payment, as well as accounting the sale at the inventory.

The ORC is about a *Software as a Service* (SaaS) scenario where a service providers offers a solution to a number of different customers. Thus the service provider negotiates an SLA with each customer. The service provider in turn relies on a Software Provider (delivering the ORC application), a cloud-like Infrastructure Provider and External Service Providers which offer complementing functionality.

The ORC itself consists of a bundle of atomic and composite software services (for inventory, ordering and payment) which in turn may depend on other external software services (such as credit card validation). Technically, the ORC is offered with 2 deployment options (addressing small and large customers): *all in one*, with the service containers, BPEL engine, and database running on the same virtual machine; *separated database*, with the database running on a different virtual machine; and *separated database and BPEL engine*.

### 3.2 SLA Hierarchy

Generally, a hierarchy of SLAs is a set of SLAs that are associated in a way that captures some kind of dependency of one on another. This kind of association / dependency is not always straightforward. It may be the case that the reduced capacity of a provider forces it to rent capacity from another provider, to serve its clients as per the standing agreements. In a different scenario, the dependency might be due to a request for fail-over redundancy, in which case the failure of an agreement of the provider may not affect at all the agreements of its customer. Dependencies may just as well be related to functionality that a provider cannot offer by default, and that therefore must be outsourced.

SLA dependencies may also be internal to a single provider. This is the case where a department of a provider relies on another department of the same provider, in order to accomplish its tasks. Very often this has to do with a business department relying on the IT department, for back-office or other functions.

Our evaluation follows the scenario of a single provider with 3 internal departments (business department, software IT department and infrastructure department) relying on each other via internal SLAs.

Figure 4 illustrates the realized SLA hierarchy. It shows the 3 departmental layers where the top-level business SLA describes the complete offered Retail-as-a-Service solution, including software but also relevant business aspects (legal conditions, support agreements, etc.). The software service bundle relies on a hierarchy of lower-level software services. The hierarchy here allows the service provider to precisely understand and monitor his software landscape in relation to top-level business SLAs. Last, the collection of software services relies on the infrastructure where the SLA specifies the nature and conditions of the infrastructure resources (here virtual machines) needed to host the software.



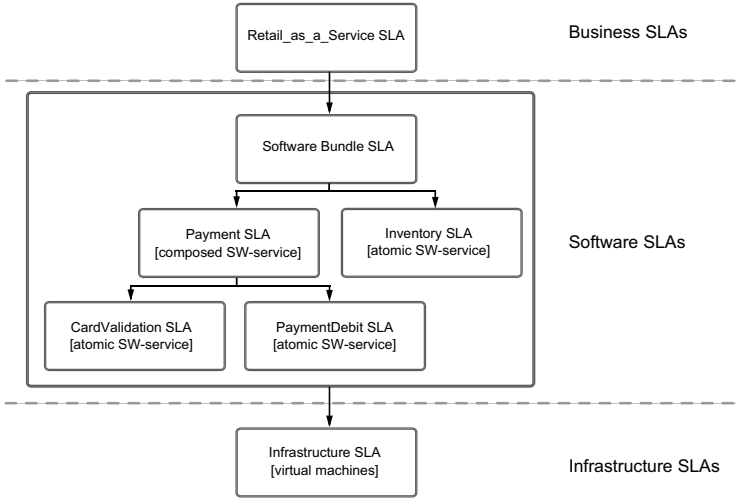


Fig. 4. Implemented SLA hierarchy

### 3.3 Evaluation

Our experiments with the SLA Framework cover the two main scenarios already described in Section 2.1.

For the negotiation and provisioning scenario, we experimented with different SLA templates, deployment options and infrastructure options. We decided to differentiate services starting from the highest service layer, the business services layer, and we adopted the widely used metaphor of *gold*, *silver* and *bronze* class services. Hence, we run our experiments with three distinguished business level SLA templates. Although the three SLA templates provide default values, the customer in our scenarios should be able to request individual non-functional properties, i.e. completion times and arrival rates (as *qualifying condition*) for each service operation accessible by the customer. Note, that the *qualifying condition* is understood as “customer obligation” within an SLA, meaning that if the arrival rate of service requests is higher than agreed, the service provider is no more bound to sustain the agreed completion time. In this case it is simply the customer who violated the SLA. At the software level we used individual templates for each service in order to support flexible service composition and provisioning, and to gain fine grained SLA hierarchies as depicted in Figure 4. The infrastructure template allows resource configuration (comprising multiple virtual machines) and enables the service provider to request guarantees from the infrastructure provider on various virtual resources, such as CPU or memory.

The core of the actual negotiation procedure is now the translation of terms along the SLA hierarchy, the evaluation of different composition/deployment/infrastructure options and the selection of the best suited ones. The basic element to this procedure is a prediction service which, based on a PCM model [7] of the

ORC, can simulate the non-functional behaviour of all these options. Taking the results of this prediction, the SLA hierarchies of possible solutions are associated with terms on their non-functional behaviour and costs. Last, the negotiation component selects the cheapest option that still satisfies the requested top-level SLA. Experiments show the feasibility of this approach and the selection of well-suited system setups which realize the requested customer SLAs in a cost efficient manner.

For what concerns the monitoring and adjustment scenario, a software layer MRE, e.g. a too high average response time of an operation call, is identified by the name of the guarantee term and the unique id of the SLA to which the guarantee term belongs. The violation, for instance, may refer to the guarantee term on the average completion time of the operation `bookSale` of the service `InventoryService` in the ORC. This information is sufficient for the Adjustment module to retrieve the violated SLA and the related SLA hierarchy and to decide eventual control actions. Infrastructure violations report the id of the resource on which the violation has occurred and information on the parameter, e.g. CPU or memory utilisation, which has been violated.

Adjustment actions have been realised at various layers relying on the analysis done within the adjustment module. SLA violations may result from faulty customer behaviour, e.g. where a customer exceeds the SLA-agreed maximum workload. In this case a message is sent via the eContracting module to the customer, to inform him/her about this.

Concerning the infrastructure, Adjustment may trigger the re-provisioning of the virtual machines on which services are executed. In particular, if the Adjustment receives a violation of the software-level guarantee term on a service operation completion time and, at the same time, a violation of the CPU utilisation on the virtual machine on which the service is executed, then it commands the Infrastructure to re-provision the virtual machine on which the service is executing with a higher CPU share.

Concerning the software layer, the corrective action implemented at the current stage is the reconfiguration of the BPEL engine in which composite services are executed. The reconfiguration is triggered when a software-level guarantee term on completion time of an operation of a composite service is violated, and when it is not possible to find a corresponding infrastructure violation of the virtual machine where the service is executed. The Adjustment detects, in this case, that the problem belongs to the BPEL engine in which the service is executing; the reconfiguration of the engine involves the increase of thread pool of the composite service for which the violation has been detected.

Last and if no other adjustment action can be detected, violations (MRE) are reported as fault of the service provider and the customer is informed accordingly, including the acceptance of possibly agreed penalties.

## 4 Conclusions

This paper presents a technical architecture and implementation for a multi-level SLA management framework. We discussed the fundamental components

and interfaces in the architecture. The framework dynamics were described via two fundamental scenarios, which cover the core SLA management lifecycle including negotiation, provisioning, monitoring and adjustment. Furthermore, the main technical and integration aspects of the developed framework have been described. The framework has been successfully applied within the context of a reference application. The qualitative evaluation includes a description of the actually realised SLA hierarchy and the details about the concrete scenario steps. A full version of this paper including a discussion of related work can be found at [4].

Although the experiments have been successful as a feasibility study, they also revealed some areas that require further improvement. The most important among these are a clearer separation of concerns between the horizontal aspect of system layers and the vertical aspect of SLA management. Second, the framework must support more flexible integration techniques for different target scenarios. Besides improving the architecture along the aforementioned lines, we will also start a thorough evaluation based on five industrial use cases, which represent a broad set of relevant but also distinct scenarios.

## Acknowledgements

The research leading to these results is partially supported by the European Community's Seventh Framework Programme (FP7/2001-2013) under grant agreement no.216556.

## References

1. SLA@SOI project: IST- 216556; Empowering the Service Economy with SLA-aware Infrastructures, <http://www.sla-at-soi.eu/>
2. Papazoglou, M., van den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal* 16(3), 389–415 (2007)
3. Theilmann, W., Yahyapour, R., Butler, J.: Multi-level sla management for service-oriented infrastructures. *Towards a Service-Based Internet*, 324–335 (2008)
4. Comuzzi, M., Kotsokalis, C., Rathfelder, C., Theilmann, W., Winkler, U., Zacco, G.: A framework for multi-level sla management. Technical Report 2010-1, SLA@SOI project (April 2010)
5. Spring framework: Eliminating Enterprise Java Complexity, <http://www.springsource.com/>
6. Rausch, A., Reussner, R., Mirandola, R. (eds.): *The Common Component Modeling Example*. LNCS, vol. 5153. Springer, Heidelberg (2008)
7. Becker, S., Koziolok, H., Reussner, R.: The Palladio component model for model-driven performance prediction. *Journal of Systems and Software* 82(1), 3–22 (2009)

# EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications

Jiang Dejun<sup>1,2</sup>, Guillaume Pierre<sup>1</sup>, and Chi-Hung Chi<sup>2</sup>

<sup>1</sup> VU University Amsterdam

<sup>2</sup> Tsinghua University Beijing

**Abstract.** Cloud computing is receiving increasingly attention as it provides infinite resource capacity and “pay-as-you-go” resource usage pattern to hosted applications. To maintain its SLA targets, resource provisioning of service-oriented applications in the cloud requires reliable performance from the cloud resources. In this paper, we study performance behavior of small instances in Amazon EC2. We demonstrate that the performance of virtual instances is relatively stable over time with fluctuations of mean response time within at most 8% of the long-term average. Moreover, we also show that different supposedly identical instances often have very different performance, up to a ratio 4 from each other. We consider this as an important issue that must be addressed, but also as an opportunity as it allows one to assign each instance with a task that matches its own performance profile.

## 1 Introduction

Cloud computing is emerging today as a new paradigm for on-demand resource provisioning for service-oriented applications. Clouds are attractive to service-oriented application hosting as such applications often observe large fluctuations in their workload. Clouds allow one to add resources very quickly to a hosted application when the performance is about to violate certain criteria such as Service Level Agreements (SLA) or system load. Similarly, clouds offer the opportunity to release resources when the load decreases. Service Level Agreements typically define requirements regarding the performance, security, availability and the like from a user perspective. We focus here on the performance aspects.

Cloud platforms typically do not give access to actual physical machines but rely heavily on virtualization techniques for reasons of cost effectiveness and technical flexibility. Virtual machine monitors such as Xen allow fine-grained performance isolation between multiple virtual instances sharing the same physical resource [1]. The usual wisdom is that CPU performance can be isolated very effectively, while I/O performance is harder to isolate [2,3].

This paper addresses the following question: *how effective can the virtual machine based cloud be for SLA-aware resource provisioning of service-oriented applications?* Resource provisioning mechanisms traditionally rely on two fundamental performance properties of the available resource units [4]:

- Performance stability: the performance of the provisioned resource units should remain constant over time. In the virtual machine based clouds, the performance of the same virtual machine should be stable without being affected by the activity of other virtual machines on the same hardware.
- Performance homogeneity: the performance of different resource units should be predictable through the profiling of current deployed resource units. It requires that the performance behavior of resource units are homogeneous. In real-world applications, cloud providers commonly provide users with a set of different virtual machine types, each of which has different resource capacities in terms of CPU capacity, RAM size, disk I/O bandwidth and the like. The performance of different types of virtual machines are obviously heterogeneous. However, the performance of multiple virtual machines of the same type should be similar. Otherwise, it becomes very hard for one to quantify the number of virtual machines to be provisioned such that the performance of hosted applications meet its SLA targets.

Although these two properties are typically true for cluster-based systems where identical physical resources are exclusively dedicated to a single application, the introduction of virtualization in the cloud requires a re-examination of these properties. This paper studies the above two performance properties of virtual instances provided by Amazon EC2 [5]. We then evaluate the possibility of providing SLA-aware resource provisioning to service-oriented applications in the cloud.

As the workloads of commonly-used applications (such as three-tier web applications) can be either CPU-intensive at the application server tier or I/O-intensive at the backend database server tier, we benchmark virtual instances on EC2 using three synthetic web applications that exhibit different types of workload pattern. Based on the measured performance data, we find performance of virtual instances to be relatively stable. The mean response time of a single small instance fluctuates by at most 8% around its long-term average. On the other hand, multiple instances of the same type show very heterogeneous performance profiles, up to a ratio 4 in response time from each other. We also observed that the CPU and I/O performances of different “identical” virtual instances are not correlated. We consider this as a very important issue that must be addressed to allow effective resource provisioning in the cloud. At the same time, we see this as an opportunity to exploit these differences and assign each virtual instance with a task that matches its own performance profile.

The remain of this paper is organized as follows: Section 2 introduces research efforts related to our work. Section 3 presents our methodology to benchmark virtual instances on Amazon EC2. Section 4 discusses performance analysis results. Finally, Section 5 concludes our work and proposes future research directions for resource provisioning in virtualized cloud environments.

## 2 Related Work

A number of recent research works made efforts towards resource provisioning of virtual machines to multi-tier web applications. For instance, [6] assumes the

existence of a performance model which determines the number of virtual machines at each tier of the application. It then formalizes the mapping problem of assigning a set of virtual machines to a pool of physical hosts with the goal of maximizing resource usage. Similarly, [7] assumes that the most-demanding resource at each tier of the application has a well-defined capacity in terms of CPU, I/O and network. The paper then presents a general performance model for multi-tier web applications. The model determines the capacity of virtual machines provisioned to each tier with the goal of optimizing certain utility functions, such as revenues of provisioned resource. Although these works propose resource provisioning solutions that rely on virtual machines, they do not investigate the actual performance behavior of virtual machines. On the other hand, this paper focuses specifically on the performance of virtual machines in a real-world environment and aims at proposing practicable solutions for virtual machine provisioning.

To our best knowledge, few works focus on performance analysis of the virtual machines provided by clouds. For example, [8] analyzes the performance of Amazon EC2 using micro-benchmarks, kernels, and e-Science workloads. This analysis targets the evaluation of usefulness of EC2 as a scientific computing platform. In this paper we also take Amazon EC2 as our experimental cloud platform, however we focus on the challenges and opportunities of providing SLA-aware resource provisioning to service-oriented applications in the cloud.

Finally a number of works analyze the performance impact caused by the inherent virtualization mechanisms used in commercial clouds. [9] analyzes the impact of different choices of CPU schedulers and the parameters on application performance in Xen-based virtualized environments. [3] presents a system-wide statistical profiling toolkit called Xenoprof, which is implemented for the Xen virtual machine environment. Xenoprof is then used to quantify Xen's performance overheads for network I/O processing. Similarly, [10] characterizes network I/O performance in a Xen virtualized environment. [2] uses Xen Virtual Machine Monitors to measure CPU overheads when processing a set of disk I/O and network I/O intensive workloads. [11] compares the scalability of four virtual machine technologies in terms of CPU, memory, disk and network. These works help us to understand the performance behaviors of virtual machines. However, our work differs from these as we profile virtual machines at a macroscopic level with the premise of taking virtual machines as black boxes, and we finally aim to provision resources based on them.

### 3 Methodology

In this section we introduce our experimental environment on Amazon EC2 and present the detail of our experimental setup.

Amazon EC2 provides 5 types of virtual instances, each of which has different capacities in terms of CPU capacity, RAM size and I/O bandwidth. Table I shows the announced capacity details of virtual instances on EC2. To provide fault-tolerance, EC2 provides its virtual instances across multiple data centers

**Table 1.** Capacity detail of virtual instances on EC2

Instance type	Compute units	RAM	I/O performance
Small	1	1.7GB	Moderate
Medium - high CPU	5	1.7GB	Moderate
Large	4	7.5GB	High
Extra large	8	15GB	High
Extra large - high CPU	20	7GB	High

**Table 2.** Software environment in all experiments

Application server	Database server	JDK	Operating system	Kernel
Tomcat 6.0.20	MySQL 5.1.23	JDK 6 update 14	Ubuntu 8.10	Linux 2.6.21

organized in so-called availability zones<sup>1</sup>. Two virtual instances running in different availability zones are guaranteed to be executed in different data centers. Of the six availability zones, four are located in the U.S. and the other two are in Europe.

In this paper, we examine the performance of small instances on EC2 as they are the most widely used. To demonstrate that the same performance features appear on different types of virtual instances as well, we also partially benchmark medium instances with high CPU<sup>2</sup>.

In practice, service-oriented applications are commonly deployed in different data centers for fault tolerance and to deliver good quality of service to users in different locations. To match this common case we examine the performance of small instances in all six availability zones. This also allows us to make sure that the experimental instances do not interfere with each other.

In order to provision virtual machines for service-oriented applications, it is important for one to predict its future performance if given one more or one less resource. This performance predictability in turn requires that performance of the same virtual machine remains constant over time. In addition, it requires that the performance of newly allocated virtual instances is similar to that of currently-deployed instances. We therefore carry out three groups of experiments to benchmark small instances on EC2.

*Performance stability:* The first group of experiments studies the performance stability of small instances under the constant workload intensity. As workloads of service-oriented applications can be CPU-intensive and database I/O intensive, we develop the following three synthetic web applications to simulate different types of workload patterns:

<sup>1</sup> There are four zones located in the United States (US-EAST-1A, US-EAST-1B, US-EAST-1C and US-EAST-1D) and two zones in Europe (EU-WEST-1A and EU-WEST-1B).

<sup>2</sup> We leave the experiments on other types of instances for future work.

- T1: a CPU-intensive web application. This application consists of a servlet processing XML transformation based on client inputs. It issues no disk I/O (except for reading configuration file when starting up) and very little network I/O (each request returns one html page of size around 1,600 bytes). The request inter-arrival times are derived from a Poisson distribution. The average workload intensity is 4 requests per second.
- T2: a database read-intensive web application. This application consists of a servlet and a database hosted on two separate virtual instances. The database has 2 tables: “CUSTOMER” and “ITEM.” The “CUSTOMER” table holds 14,400,000 records while the “ITEM” table holds 50,000,000 records. The size of data set is 6.5 GB, which is nearly 4 times the RAM size of small instances. The servlet merely issues SQL queries to the backend database. It first gets customer order history based on customer identification, and then fetches items related to those ones in customer’s historical orders. Here as well, the request inter-arrival times are derived from a Poisson distribution. The average workload intensity is 2 requests per second.
- T3: a database write-intensive web application. This application consists of a servlet and a database hosted on two separate virtual instances. The servlet issues UDI (Update, Delete and Insert) queries to execute write operations on 2 tables: “CUSTOMER” and “ITEM.” The servlet first inserts 1,440,000 records into “CUSTOMER” table and then inserts 1,000,000 records into “ITEM” table. After populating the two tables, the servlet sends queries to sequentially update each record. Finally, the servlet sends queries to delete the two tables.

In this group of experiments, we randomly select one small instance in each availability zone and run T1, T2 and T3 on each instance separately. Each run of the tested application lasts for 24 hours in order to examine the potential interference of other virtual instances on the tested application performance. We compare the statistical values of mean response time of each hour within the whole experiment period to evaluate the performance stability of small instances.

*Performance homogeneity:* The second group of experiments evaluates the performance homogeneity of different small instances. We randomly select one small instance in each availability zone and run T1 and T2 on each instance separately. Each run of the tested application lasts 6 hours such that database caches can fully warm up and the observed performance becomes stable. We repeat this process 5 times at a few hours interval such that we acquire different instances on the same availability zone<sup>3</sup>. We compare the mean response times of tested application among all tested instances in order to evaluate the performance homogeneity of different instances.

*CPU and I/O performance correlation:* The third group of experiments studies the correlation between the CPU and I/O performance of small instances.

---

<sup>3</sup> If one requested a virtual instance very quickly after another one is released, Amazon EC2 might recycle the virtual instances and return the previous one.



**Table 3.** Response time of T1 on small instances

	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D	EU-WEST 1A	EU-WEST 1B
Mean value (mean)	684.8ms	575.5ms	178ms	185.2ms	522.9ms	509.9ms
Standard deviation (std)	46.7ms	31.3ms	4.99ms	3.5ms	20.6ms	18.9ms
std/mean	6.8%	5.4%	2.8%	1.9%	3.9%	3.7%

**Table 4.** Response time of T2 on small instances

	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D	EU-WEST 1A	EU-WEST 1B
Mean value (mean)	75.9ms	76.3ms	79.9ms	71.8ms	83.8ms	71.6ms
Standard deviation (std)	1.28ms	2.05ms	6.36ms	1.14ms	2.1ms	2.34ms
std/mean	1.7%	2.7%	8.0%	1.6%	2.5%	3.3%

The experiment process is similar to the second group. Instead of running T1 and T2 separately, we run them sequentially on the same instance, each one for 6 hours. We finally correlate the CPU performance and I/O performance in all tested instances to observe their relationships.

In all above experiments the clients run on a separate virtual instance in the same availability zone as the virtual instances hosting application servers and database servers. Table 2 shows the software environment used in all experiments.

## 4 Evaluation

This section first presents the results of CPU and disk I/O performance stability on small instances in each availability zone. We then show the performance behavior of different small and medium instances. Finally, we discuss the implications of the (lack of) correlation between CPU performance and disk I/O performance of small instances for resource provisioning. We measure the response time at the server side in all experiments in order to avoid the latency error caused by the network between servers.

### 4.1 Performance Stability Evaluation

To study the performance stability of small instances, we measure the response time of each request and calculate the mean response time at a one hour granularity. We then compute the standard deviation of the 24 mean response times (each for 1 hour in the whole 24-hours experiment period). Table 3 shows the mean value and the standard deviation of the 24 mean response times for T1 in all zones.

From the perspective of long-running time periods the CPU performance is quite stable. As shown in Table 3, the standard deviation of mean response times of each hour is between 2% and 7% of the mean value, which may be acceptable in real-world hosting environments.

**Table 5.** Response time for INSERT operation of T3 on small instances

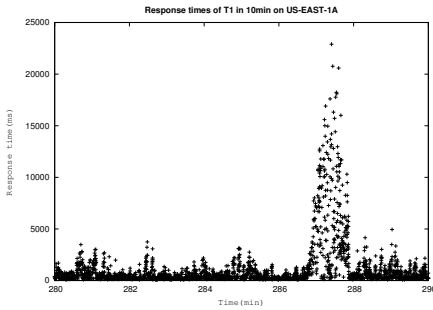
	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D	EU-WEST 1A	EU-WEST 1B
Mean value (mean)	0.33ms	0.33ms	0.33ms	0.53ms	0.47ms	0.46ms
Standard deviation (std)	0.0012ms	0.0006ms	0.0022ms	0.0021ms	0.0019ms	0.0044ms
std/mean	0.4%	0.2%	0.7%	0.4%	0.4%	0.9%

**Table 6.** Response time for UPDATE operation of T3 on small instances

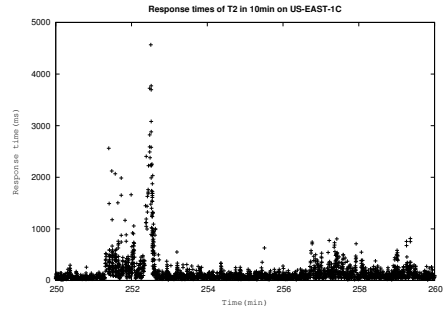
	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D	EU-WEST 1A	EU-WEST 1B
Mean value (mean)	3.84ms	3.5ms	2.67ms	3.09ms	3.72ms	3.91ms
Standard deviation (std)	0.026ms	0.035ms	0.061ms	0.03ms	0.005ms	0.026ms
std/mean	0.7%	1%	2.3%	1.0%	0.1%	0.6%

**Table 7.** Response time for DELETE operation of T3 on small instances

	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D	EU-WEST 1A	EU-WEST 1B
Mean value (mean)	30.1ms	21.7ms	30.3ms	17.8ms	15.9ms	21.7ms
Standard deviation (std)	21.4ms	0.52ms	20.2ms	4.6ms	0.33ms	0.71ms
std/mean	71.1%	2.4%	66.7%	25.8%	2.1%	3.3%



(a) Response time samples of T1



(b) Response time samples of T2

**Fig. 1.** Response time samples of T1 and T2 over a period of 10min

However, we also observed that the CPU performance could be temporarily affected by the underlying resource sharing mechanism. Figure 1(a) shows the response time of T1 over a period of 10 minutes. We observe short periods during which the response time significantly increase. The duration of such peaks is relatively short (on average 1 to 2 minutes). We attribute these peaks to external factors such as the creation of a new virtual instance in the same physical machine. (the duration of such peaks is similar to the observed delay for creating a new virtual instance). Apart from these short peaks, the CPU-capacity sharing

has little impact on the performance stability of CPU-intensive web applications when considering performance behavior in long-running periods.

We also observed that the performance of multiple small instances vary wildly from each other, from 185ms to 684ms of average response time. We further compare the performance of different small instances in section 4.2.

Similarly, we measure the response time of application T2 and compute the mean response times of each hour. Table 4 shows the mean value and the standard deviation of mean response times of each hour within the 24-hours experiment period. The database read performance of small instances is also very stable from the perspective of long-running time periods. For all tested instances, the mean response time of each hour deviates between 1.6% and 8% from the mean value. Similarly to CPU performance, the database read performance is affected by short interferences with the underlying I/O virtualization mechanism. Figure 1(b) shows the peak of response time of T2 over 10 minutes. The disturbances are also short, in the order of 2 minutes.

The database read performance of different small instances (such as in different zones) are also different from each other. We further examine database read performance homogeneity in section 4.2.

We finally evaluate the performance stability of database write-intensive workloads. Tables 5 to 7 show response time statistics for database UDI operations. As shown in Tables 5 and 6, the performance of database INSERT and UPDATE operations is very stable. We observed that the standard deviation of those mean response times is small, between 0.2% and 2.3%.

However, Table 7 shows that the performance of database DELETE operation varies a lot. The cause of this performance behavior of database DELETE operations remains to be found. Similarly to CPU and database read operations, the performance of database UDI operations on different small instances are different from each other.

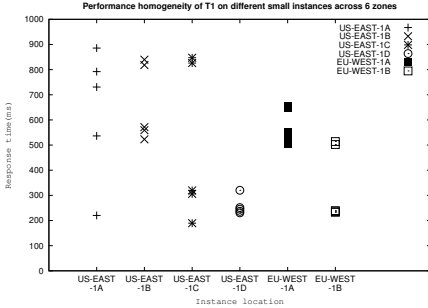
To demonstrate that other types of virtual instances on EC2 exhibit similar performance features, we partially benchmark medium instance with high CPU on EC2. We run T1 to profile CPU performance of medium instances after adjusting the request rate to match the capacity of medium instances. Table 8 shows the statistical values of mean response times of each hour on medium instances across the four US availability zones. Similar to the performance behavior of small instances, CPU performance is also relatively stable. The standard deviation of mean response time of each hour is between 2% and 10%. The CPU performance of different medium instances also varies a lot. One would however need more samples to fully explore the performance behavior of medium and large instances.

## 4.2 Performance Homogeneity Evaluation

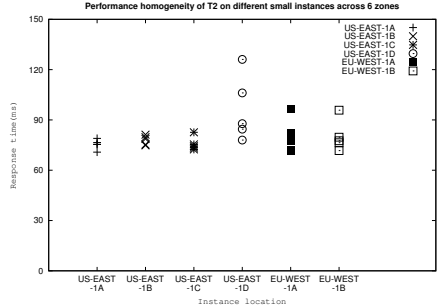
So far we observed that the CPU performance and disk I/O performance of the same small instance are relatively stable from the perspective of long-running time periods (except for database DELETE operation). However, typical resource provisioning algorithms also expect that different small instances have

**Table 8.** Mean response time of T1 on medium instances (high CPU)

	US-EAST 1A	US-EAST 1B	US-EAST 1C	US-EAST 1D
Mean value (mean)	307.7ms	791.5ms	197.3ms	199.8ms
Standard deviation (std)	27.4ms	26.1ms	3.6ms	5.8ms
std/mean	9.6%	3.3%	1.8%	2.9%



(a) CPU performance homogeneity



(b) Disk I/O performance homogeneity

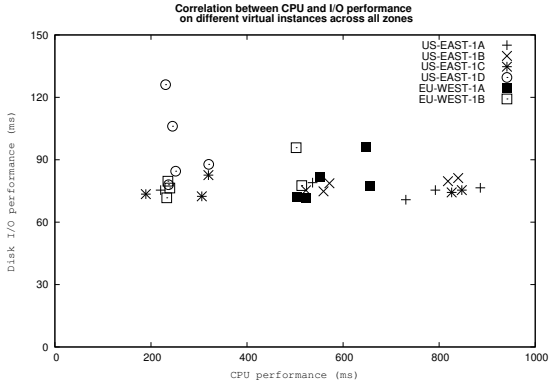
**Fig. 2.** Performance homogeneity of different small instances across all zones

homogeneous performance behavior such that the performance of future small instances is predictable based on current performance profiles. Thus, we evaluate the performance homogeneity of different small instances through the second group of experiment.

Figure 2(a) shows the mean response times of T1 for 30 different small instances across all zones (5 instances for each zone). Different instances clearly exhibit very different CPU performance when serving the exact same workload. Response times of different virtual instances vary up to a ratio 4. The same pattern appears both inside each zone and between different zones. Figure 2(b) shows similar heterogeneous performance behavior of different small instances for disk I/O operations, even though the variations are less important than for CPU. Thus, when provisioning small instances to host service-oriented applications, it will be very hard for one to predict the performance of the newly-allocated virtual instances based on the observed performance profiles of currently deployed ones. This property challenges traditional resource provisioning approaches which assume that the underlying infrastructure provides homogeneous resources.

### 4.3 Implications for Resource Provisioning

As we observed that different small instances behave differently when serving CPU-intensive and disk I/O intensive workloads, we further explore this phenomenon and run the third group of experiment to check if the CPU and disk I/O performances are correlated on supposedly identical small instances.



**Fig. 3.** Correlation between CPU and I/O performance of small instances

Figure 3 shows 30 samples of the correlation of CPU performance and disk I/O performance on identical small instances across all availability zones. Each point depicts the CPU and I/O performances of a single virtual instance. We do not observe any obvious correlation between the respective CPU and I/O performances of single instances. On the other hand, small instances can clearly be classified into three or four clusters with similar performance. Often (but not always) instances from the same availability zone are clustered together.

These results suggest that different small instances on Amazon EC2 may be suitable to process different types of workload. Within commonly-used multi-tier service-oriented applications, different tiers have different workload patterns. For example, an application server tier is commonly CPU-intensive while a database server tier is rather I/O intensive. Thus, one may consider employing well-suited small instances to provision resources to hosted applications such that each instance runs a task that matches its own performance profile. A virtual instance with fast CPU could be given an application server to run, while an instance with fast I/O would run a database server and a virtual instance with slow CPU and I/O may carry a modest task such as load balancing. We name this performance feature of small instances on EC2 as workload affinity. Although adapting resource provisioning algorithm will be a challenge, we believe that exploiting such workload affinity properties could result in improvement of the overall resource usage, even compared with a fully homogeneous case.

## 5 Conclusion

Cloud platforms such as Amazon EC2 are attracting attention in the service-oriented application hosting community as they provide near-infinite capacity and a “pay-as-you-go” business model. Good usage of cloud facilities may help application providers to reduce their IT investments as well as operational costs.

However, fluctuating workloads and the necessity to maintain SLAs require clouds to provide SLA-aware resource provisioning to hosted applications.

To dynamically provision virtual machines to hosted applications with guaranteed performance, it is necessary to understand the performance behavior of virtual instances provided by clouds. In this paper, we took the popular commercial cloud Amazon EC2 as an example, and evaluated the performance stability and homogeneity of small instances on EC2. We demonstrated that the CPU and disk I/O performance of small instances are relatively stable from the perspective of a long-running periods. However, the performance behavior of multiple “identical” small instances is very heterogeneous. We claim that this property challenges the effectiveness of current resource provisioning approaches if employed in the virtual machine based cloud as these approaches assume homogeneous sets of underlying resources. We consider this as an important issue that must be addressed by the resource provisioning community. At the same time, we believe that this variety of workload affinity provides opportunities for future algorithms to make more effective use of the resources offered by the cloud.

## References

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Proc. SOSP (2003)
2. Cherkasova, L., Gardner, R.: Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In: Proc. USENIX Annual Technical Conf. (2005)
3. Menon, A., Santos, J.R., Turner, Y., Janakiraman, G.J., Zwaenepoel, W.: Diagnosing performance overheads in the Xen virtual machine environment. In: Proc. Intl. Conf. on Virtual execution environments (2005)
4. Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P.: Dynamic provisioning of multi-tier internet applications. In: Proc. Intl. Conf. on Autonomic Computing (2005)
5. Amazon.com: Amazon Elastic Compute Cloud, <http://aws.amazon.com/ec2/>
6. Campegiani, P., Presti, F.L.: A general model for virtual machines resources allocation in multi-tier distributed systems. In: Proc. Intl. Conf. on Autonomic and Autonomous Systems, pp. 162–167 (2009)
7. Wang, X., Du, Z., Chen, Y., Li, S.: Virtualization-based autonomic resource management for multi-tier web applications in shared data center. *Journal of Systems and Software* 81(9), 1591–1608 (2008)
8. Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., Epema, D.: An early performance analysis of cloud computing services for scientific computing. Technical Report PDS-2008-006, Delft University of Technology (December 2008)
9. Cherkasova, L., Gupta, D., Vahdat, A.: When virtual is harder than real: Resource allocation challenges in virtual machine based it environments. Technical Report HPL-2007-25, HP Laboratories Palo Alto (February 2007)
10. Apparao, P., Makineni, S., Newell, D.: Characterization of network processing overheads in Xen. In: Proc. Intl. Workshop on Virtualization Technology in Distributed Computing (2006)
11. Quetier, B., Neri, V., Cappello, F.: Scalability comparison of four host virtualization tools. *Journal of Grid Computing* 5(1), 83–98 (2007)

# On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management

Patrícia Silveira<sup>1</sup>, Carlos Rodríguez<sup>1</sup>, Fabio Casati<sup>1</sup>, Florian Daniel<sup>1</sup>,  
Vincenzo D'Andrea<sup>1</sup>, Claire Worledge<sup>2</sup>, and Zouhair Taheri<sup>3</sup>

<sup>1</sup> University of Trento, Italy

{silveira, crodriguez, casati, daniel, dandrea}@disi.unitn.it

<sup>2</sup> Deloitte Conseil, Paris, France

cworledge@deloitte.fr

<sup>3</sup> PricewaterhouseCoopers Accountants, Rotterdam, Netherlands

zouhair.taheri@nl.pwc.com

**Abstract.** Assessing whether a company's business practices conform to laws and regulations and follow standards, i.e., compliance governance, is a complex and costly task. Few software tools aiding compliance governance exist; however, they typically do not address the needs of who is in charge of assessing and controlling compliance, that is, compliance experts and auditors. We advocate the use of compliance governance dashboards, whose design and implementation is however challenging for these reasons: (i) it is fundamental to identify the right level of abstraction for the information to be shown; (ii) it is not trivial to visualize distinct analysis perspectives; and (iii) it is difficult to manage the large amount of involved concepts, instruments, and data. This paper shows how to address these issues, which concepts and models underlie the problem, and, how IT can effectively support compliance analysis in SOAs.

## 1 Introduction

*Compliance* is a term generally used to refer to the conformance to a set of laws, regulations, policies, or best practices. *Compliance governance* refers to the set of procedures, methodologies, and technologies put in place by a corporation to carry out, monitor, and manage compliance.

Compliance governance is an important, expensive, and complex problem to deal with: It is *important* because there is increasing regulatory pressure on companies to meet a variety of policies and laws (e.g., Basel II, SOX). This increase has been to a large extent fueled by high-profile bankruptcy cases (Parmalat, WorldCom, the recent crisis) or safety mishaps (the April 2009 earthquake in Italy has already led to stricter rules and procedures for construction companies). Failing to meet these regulations means safety risks, hefty penalties, loss of reputation, or even bankruptcy [9].

Managing and auditing/certifying compliance is a very *expensive* endeavor. A report by AMR Research [5] estimates that companies will spend US\$32B only on governance, compliance, and risk in 2008 and more than US\$33B in 2009. Audits are themselves expensive and invasive activities, costly not only in terms of auditors' salaries but also in terms of internal costs for preparing for and assisting the audit – not to mention the cost of non-compliance in terms of penalties and reputation.

Finally, the problem is *complex* because each corporation has to face a large set of compliance requirements in the various business segments, from how internal IT is managed to how personnel is trained, how product safety is ensured, or how (and how promptly) information is provided to shareholders. As a result, compliance governance requires understanding/interpreting requirements and implementing and managing a large number of control actions on a variety of procedures across the business units of a company. Each compliance regulation and procedure may require its own control mechanism and its own set of indicators to assess the compliance status of the procedure [1]. Today, compliance is to a large extent managed by the various business units in rather ad-hoc ways (each unit, line of business, or even each business process has its own methodology, policy, controls, and technology for managing compliance). Hence, it is very hard for any CFO or CIO to answer questions such as [16]: *Which rules does my company have to comply with? Which processes are following regulations? Where do violations occur? Which processes do we have under control?* Even more, it is hard to do so from a perspective that not only satisfies the company but also the company's *auditors*, since they are the ones that certify compliance.

This paper presents a conceptual model for compliance and for *compliance governance dashboards* (CGDs), along with a dashboard architecture and a prototype implementation. The aim of our CGD is to report on compliance, to create an awareness of possible problems/ violations, and to facilitate the identification of root-causes for non-compliant situations. The dashboard is targeted at several classes of users: chief officers of a company, line of business managers, internal auditors, and external auditors (certification agencies). Typically, the two latter focus on a narrow set of processes and historical data to verify non-compliant situations and how they have been dealt with. Via the CGD, they also have *access to key compliance indicators* (KCIs). Managers are interested in a much broader set of compliance regulations and at quasi-real time compliance information that allows them to detect problems (unsatisfactory KCIs) as they happen and identify the causes (drill-down to the root of the problem), so that they can take decisions before they become (significant) violations. They have access and navigate through the entire set of regulations, business processes, and business units and also observe the overall compliance status (through KCIs).

Technically, building a dashboard that shows a bunch of indicators and allows drill-downs is easy. Indeed, the main challenges are *conceptual* more than technological [15] and constitute the contributions of this paper as follows:

1. Provide a *conceptual model for compliance and for compliance dashboards* that covers a broad class of compliance issues. It is important to identify the key abstractions and their relationships; otherwise the dashboard loses its value of single entry point for compliance assessment.
2. Combine the above *broadness with simplicity and effectiveness*. The challenge here is to derive a model that, despite being broad, remains simple and useful. If the abstractions are not carefully crafted and kept to a minimum, the dashboard will be too complex and remain unused. As we have experienced, this problem may seem easy but is instead rather complex, up to the point that discussions on the conceptual model in the projects took well over a year. There is no clarity in this area, and this is demonstrated by the fact that while everybody talks about compliance, there are no generic but simple compliance models available.

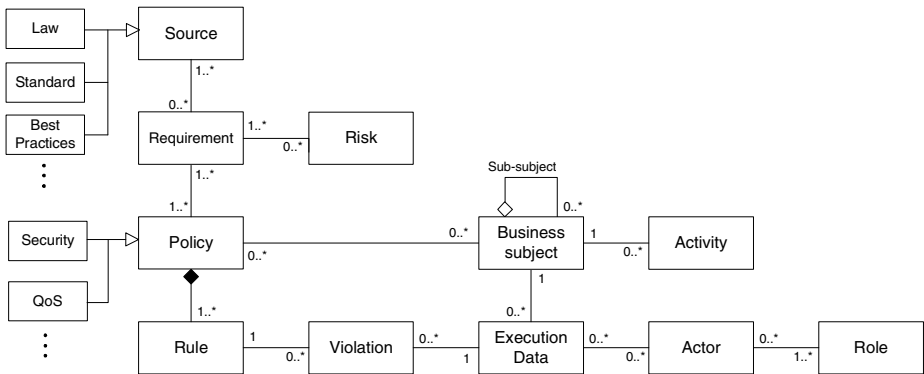


3. Define a *user interaction and navigation model* that captures the way the different kinds of users need to interact with the dashboard, to minimize the time to accesses spent in getting the information users need and to make sure that key problems do not remain unnoticed.
4. Derive a model aligned with the *criteria and approach that auditors have* to verify compliance. In this paper, this latter is achieved “by design”, in that the model is derived via a joint effort of two of the major auditing companies and reflects the desired method of understanding of and navigation among compliance concerns.

## 2 The Problem of Compliance Management

Despite the increasing awareness of compliance issues in companies and the recognition that part of the compliance auditing task can be automated, i.e., assisted by software tools [9][12][13], there is still a lot of confusion around. This is especially true for the IT community, which would actually be in charge of aiding compliance governance with dedicated software. To help thinking in terms of auditing, in the following we aim to abstract a wide class of compliance problems into a few key concepts that are also the ones understood by auditors. The resulting model (see Fig. 1) does not cover all possible compliance problems, but our goal is to strike a balance between coverage and simplicity. So far, we didn’t find any such model in literature.

At the top-left corner: The *Regulation* entity generalizes all documents that provide guidelines for the good conduct of business in a given domain. Examples of regulations are legislations (e.g., MiFID, The Electronic Commerce Directive), laws (e.g., SOX, HIPAA), standards (e.g., CoBIT, ISO-9001), and SLAs. Typically, a regulation defines a set of rules in natural language, which constrain or guide the way business is conducted. *Complying* with a regulation means satisfying its rules. The selection of the pertaining ones represents the *requirements* for compliance management, usually expressed in terms of control objectives and activities. A regulation expresses multiple requirements, and a requirement might relate to one or more regulations.



**Fig. 1.** Conceptual model of the compliance management problem

Assessing compliance demands for an interpretation and translation of the requirements provided in natural language in an actionable rule description (especially in the case of principle-based regulations) [7][8]. This is modeled by the *Rule* entity, which represents actionable rules expressed either in natural language (using the company's terminology and telling exactly how to perform work) or, as desirable in a formalism that facilitates its automated processing (e.g., Boolean expressions over events generated during business execution). Rules are then grouped into *policies*, which are the company-internal documents that operatively describe how the company intends achieving compliance with the selected requirements. Typically, policies group requirements into topics, e.g., security policies, QoS policies, and similar.

At a strategic level, compliance is related to the concept of risk. Non-compliant situations expose a company to risks that might be mitigated (e.g., a non-encrypted message that is sent through the network might violate a security compliance rule, which might put at risk sensitive information). Risk mitigation is the actual driver for internal compliance auditing. The *Risk* entity represents the risks a company wants to monitor; risks are associated with compliance requirements. For the evaluation of whether business execution is compliant, we must know which rules must be evaluated in which business context. We therefore assume that we can associate policies with specific *business processes*. Processes are composed of *activities*, which represent the atomic work items in a process.

The actual evaluation of compliance rules is not performed on business processes (that is, on their models) but on their concrete executions (their instances). Executing a business process means performing activities, invoking services, and produced business data (captured by the *Execution data* entity). In addition, e.g., separation of duties, it is necessary to track the *actors* and *roles* of execution of activities. When evaluation of a rule for a process/activity instance is negative, it corresponds to *violations*, which are the core for assessing compliance level and computing KCIs.

The model in Fig. 1 puts into context the most important concepts auditors are interested in when auditing a company. Indeed, the typical auditing process looks at a the company decides which regulations are pertaining, how it implements its business processes, how it checks for violations, and so on. In short, the auditing process is embedded in a so-called compliance management life cycle [18].

### 3 Designing Compliance Governance Dashboards (CGDs)

To aid the internal evaluation and to help a company pass external audits, a concise and intuitive visualization of its compliance state is paramount. To report on compliance, we advocate the use of a web-based CGD, whose good design is not trivial [4][14]. It is important to understand how: i) the information auditors expect to find look like; ii) large amounts of data can be visualized in an effective manner, and how data can be meaningfully grouped and summarized; and iii) to structure the available information into multiple pages, that is, how to intuitively guide the user through the wealth of information. Each page of the dashboard should be concise and intuitive, yet complete and expressive. It is important that users are immediately able to identify the key information in a page, but that there are also facilities to drill-down to details.

Designing a CGD requires mastering some new concepts in addition to those discussed above. Then, the new concepts must be equipped with a well-thought navigation structure to effectively convey the necessary information. Here, we do not focus on how data are stored and how rules are evaluated; several proposals and approaches have been conceived so far for that (see Section 4), and we build on top of them.

### 3.1 A Conceptual Model for CGDs

In Fig. 2 we extend the conceptual model (Fig. 1) to capture the necessary constructs for the development of a CGD (bold lines represent new entities and their respective interrelations). The extensions aim at (i) providing different *analysis perspectives*, (ii) *summarizing* data at different levels of abstraction, and (iii) enabling drill-down/roll-up features (from aggregated data to detailed data, and vice versa).

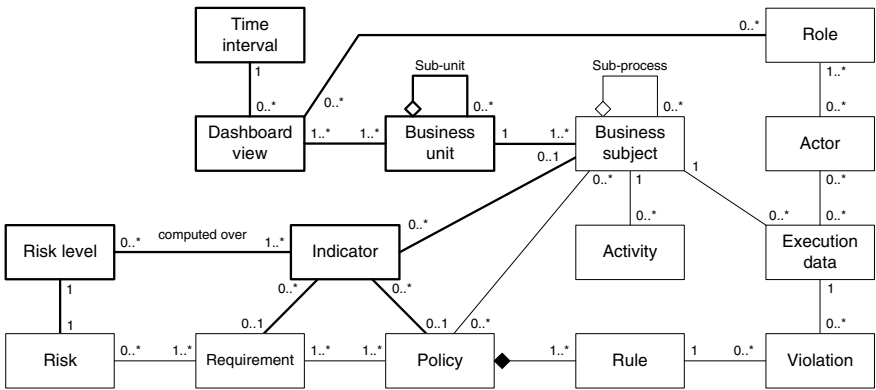


Fig. 2. Conceptual model for CGDs (dashboard-specific constructs are in bold)

The *Dashboard view* entity represents individual views over the compliance status of the company. A view is characterized by the user role that accesses it, e.g., IT specialists, compliance experts, or managers. Each of these roles has different needs and rights. For instance, managers are more interested in aggregated data, risk levels, and long time horizons (to take decisions); IT personnel are interested in instance-level data and short time spans (to fix violations). A view is further characterized by the *time interval* considered for showing data (e.g., day or year), also providing for the historical analysis (e.g., last year) and supporting different reporting purposes (operative, tactical, strategic). Finally, a view might be restricted to some of the company’s *business units*, based on the role of the user. In summary, views support distinct summarization levels of the available data, ranging over multiple granularity levels.

Effective summarization of data is one of the most challenging aspects in the design of a CGD instrumented with indicators [11]. An *indicator* is a quantitative summarization of a particular aspect of interest in the business, i.e., a metric of how well an objective is being reached. Typically, KPIs (key performance indicators), are used to summarize the level at which business objectives are reached. In our context, we speak about KCIs, referring to the achievement of the stated compliance objectives (e.g., the number of unauthorized accesses to our payroll data).

In general, indicators are computed out of a variety of data and functions; in the context of compliance assessment, however, indicators can typically be related to the ratio of encountered violations vs. compliant instances of a process or activity. As an abstraction of indicator values, we can define taxonomies (e.g., low, medium, high) and use colors (e.g., red, yellow, green) for their intuitive visualization.

### 3.2 Navigation Design

After discussing the *static* aspects of the design of CGDs, we now focus on the *dynamic* aspect, i.e., on how to structure the interaction of users with the dashboard, and on how users can explore the data underlying the dashboard application. Specifically, on top of the conceptual model for CGDs we now describe how complex data can be organized into hypertext pages and which navigation paths are important.

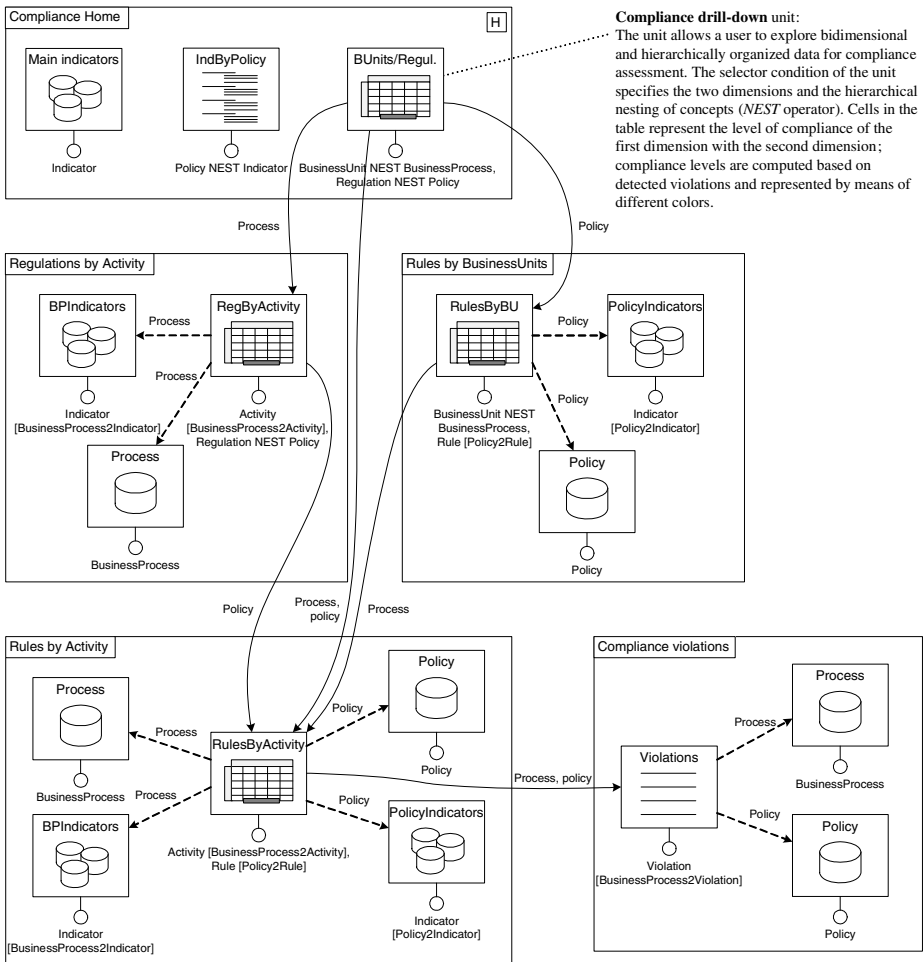
For this purpose, we adopt the Web Modeling Language (WebML [2]), a conceptual modeling notation and methodology for the development of data-intensive web applications. We use the language for the purpose of illustration only (we show a simplified, not executable WebML schema) and intuitively introduce all the necessary constructs along with the description of the actual CGD navigation structure.

The WebML hypertext schema (Fig. 3) describes the organization of our ideal CGD. It consists of five *pages* (the boxes with the name labels in the upper left corner), Compliance Home being the home page. Each page contains a number of *content units*, which represent the publication of contents from the data schema in Fig. 2 (the *selector* condition below the units indicates the source data entity). Usually, there are many *hyperlinks* (the arrows) in a hypertext schema, representing the navigations a user might perform, but, for simplicity, we limit our explanation to only those links that represent the main navigation flow. Links carry *parameters*, which represent the selection done by the user when activating a link (e.g., the selection of a process from a list). For the purpose of reporting on compliance, we define a new content unit (not part of WebML), the *compliance drill-down* unit, which allows us to show compliance data in a table-like structure (see the screenshots in Fig. 4).

Let's examine the CGD's structure (Fig. 3): The home page of the CGD provides insight into the compliance state of the company at a glance. It shows the set of most important indicators (Main indicators *multidata* unit) and a set of indicators grouped by policy (IndByPolicy *hierarchical index* unit). Then, we show the (BUnits/Regul.) unit that allows the user to drill-down from business units to processes and from regulations to policies. A click on one of: i) the processes leads the user to the Regulations by Activity page; ii) regulations leads her to the Rules by BusinessUnits page; and iii) the cell of the table leads her to the Rules by Activity page. After the selection of a process, in the Regulations by Activity page the user can inspect the compliance state of each activity of the selected process with the given regulations and policies (Reg-ByActivity), a set of related indicators (BPIndicators unit; the unit consumes the Process parameter), and the details of the selected process (Process *data* unit). Similar details are shown for policies in the Rules by BusinessUnits page, which allows the user to inspect the satisfaction of individual compliance rules at business unit or process level (RulesByBU). A further selection in the compliance drill-down units in these last two pages or the selection of a cell in the BUnits/Regul. unit in the home page

leads the user to the Rules by Activity page, which provides the user with the lowest level of aggregated information. It visualizes the satisfaction of the compliance rules of the chosen policy by the individual activities of the chosen process (RulesByActivity), along with the details of the chosen policy and process and their respective indicators. A further selection in this page leads the user to the Compliance violations page, which shows the details of the violations related to the chosen process/policy combination at an instance level in the Violations *index* unit.

The navigation structure in Fig. 3 shows one of the possible views over the data in Fig. 2, e.g., the one of the internal compliance expert. Other views can easily be added by restraining access to data and defining alternative navigation structures. Each page provides a distinct summarization level from high-level information to low-level details. The time interval for the visualization can be chosen in each of the pages.



**Fig. 3.** WebML hypertext schema structuring the navigation of CGD concepts and data

### 3.3 CGDs in Practice

In Fig. 4 we illustrate some screenshots from our prototype CGD, in order to illustrate its look and feel. The screenshots show views that consistently present our ideal CGD. Fig. 4(a) shows the Compliance Home page, Fig. 4(b) the Rules by Activity page, and Fig. 4(c) the Compliance violations page.

Compliance Home concentrates on the most important information at a glance, condensed into just one page. The five colored indicators (top left) are the most relevant, showing the most critical non-compliant regulations. The gray indicators (right) report on the compliance with the three main policies. In the bottom, there is the interactive compliance drill-down table containing the compliance performance of business units and processes (rows) in relation to regulations and policies (columns).



Fig. 4. Example CGD screenshots of our prototype implementation

The user can easily reach lower levels of granularity by drilling down on the table. For instance, the Rules by Activity page condenses lower level information concerning a combination of Business Process 1.1 and the company's SOX policy. The colors of the cells represent the compliance performance of each combination (e.g., the Business activity 32.1 presents a critical situation regarding Rule 3 of SOX - Section 301 (red cell) and weak performance regarding Rule 5, and Rule 6 (yellow cells)).

A drill-down on the red cell, for instance, leads us to the Compliance violations page, which provides the lowest level of abstraction in form of a table of event violations of the selected rule. The page illustrates the main information that must be reported to assist internal and external auditors. The data in the particular page reports all violations of one activity in Business Process 1.1 of Business Unit 1, detected considering Rule 3 of SOX - Section 301. Each row of the table represents a distinct violation and the columns contain the typical information required by auditors, e.g., responsible of activity, timestamps, mitigation action, cause of violation.

The amount and position of the graphical widgets for indicators, tables, summaries, and so on are chosen in accordance with our short-term memory and the convention of most western languages that are read from left to right and from top to bottom [4].

## 4 Related Work

To the best of our knowledge, there are only few works that deal with the problem we address in this paper. For example, [1] studies the problem of designing visualizations (i.e., the representation of data through visual languages) for risk and compliance management. Such study focuses on capturing the information required by users and on providing visual metaphors for satisfying those requirements. In [3], the performance reporting is provided in a model-driven fashion. The framework provides four models: data, navigation, report template, and access control, which jointly help designing a business performance dashboard.

Business Activity Monitoring (BAM) has gained attention in the last decade, and many tools support it. BAM aims at providing aggregated information suitable for performing analysis on data obtained from the execution of business activities. For example, tools such as Oracle BAM, Nimbus and IBM Tivoli aim at providing its users with real-time visual information and alerts based on business events in a SOA. The information provided to users comes in the form of dashboards for reporting on KPIs and SLA violations. The compliance management part of these tools (if any) comes in the form of monitoring of SLA violations, which need the SLA formal specifications as one of its inputs. In our work, we take a more general view on compliance (beyond SLAs, which are a special case to us) and cover the whole lifecycle of compliance governance, including a suitable CGD for reporting purposes.

## 5 Conclusions and Future Work

In this paper we discussed a relevant aspect in modern business software systems, i.e., compliance governance. Increasingly, industry and academia are investing money and efforts into the development of compliance governance solutions. Yet, we believe CGDs in particular, probably the most effective means for visualizing and reporting on compliance, have mostly been neglected so far. It is important to implement sophisticated solutions to check compliance, but it is at least as important (if not even

more) to effectively convey the results of the compliance checks to a variety of different actors, ranging from IT specialists to senior managers.

Our contribution is a conceptualization of the issues involved in the design of CGDs in service- and process-centric systems, the definition of a navigation structure that supports drill-down/roll-up features at adequate levels of detail and complexity, and a set of examples that demonstrate the concepts at work. Our aim was to devise a solution with in mind the real needs of auditors and – more importantly – with the help of people who are indeed involved every day in the auditing.

## References

1. Bellamy, R., Erickson, T., Fuller, B., Kellogg, W., Rosenbaum, R., Thomas, J., Vetting Wolf, T.: Seeing is believing: Designing visualizations for managing risk and compliance. *IBM Systems Journal* 46(2), 205–218 (2007)
2. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: *Designing Data-Intensive Web Applications*. Morgan Kaufmann Publishers Inc., USA (2002)
3. Chowdhary, P., Palpanas, T., Pinel, F., Chen, S.-K., Wu, F.Y.: Model-driven Dashboards for Business Performance Reporting. In: *Proceedings of the 10th IEEE EDOC*, pp. 374–386 (2006)
4. Few, S.: *Information Dashboard Design: The Effective Visual Communication of Data*, p. 223. O'Reilly Media, Inc., Sebastopol (2006)
5. Hagerty, J., Hackbush, J., Gaughan, D., Jacobson, S.: The Governance, Risk Management, and Compliance Spending Report, 2008-2009: Inside the \$32B GRC Market. *AMR Research* (2008)
6. Saqid, S., Governatori, G., Naimiri, K.: Modeling Control Objectives for Business Process Compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
7. Giblin, C., Müller, S., Pfitzmann, B.: From Regulatory Policies to Event Monitoring Rules: Towards Model-Driven Compliance Automation. *IBM Research Report* (October 2006)
8. Namiri, K., Stojanovic, N.: A Semantic-based Approach for Compliance Management of Internal Controls in Business Processes. In: *CAiSE 2007*, pp. 61–64 (2007)
9. Trent, H.: *Products for Managing Governance, Risk, and Compliance: Market Fluff or Relevant Stuff?* In-Depth Research Report, Burton Group (2008)
10. Lam, J.: *Operational Risk Management – Beyond Compliance to Value Creation*. White Paper, Open Pages (2007)
11. Imrey, L.: CIO Dashboards: Flying by Instrumentation. *Journal of Information Technology Management* 19(4), 31–35 (2006)
12. Evans, G., Benton, S.: The BT Risk Cockpit – a visual approach to ORM. *BT Technology Journal* 25(1) (2007)
13. Papazoglou, M.P.: Compliance Requirements for Business-process-driven SOAs. *E-Gov. Ict Professionalism and Competences Service Science* 280, 183–194 (2008)
14. Read, A., Tarrel, A., Fruhling, A.: Exploring User Preference for the Dashboard Menu Design. In: *Proceedings of the 42nd Hawaii Intern. Conf. on System Sciences*, pp. 1–10 (2009)
15. Allman, E.: Complying with Compliance. *ACM Queue* 4(7), 18–21 (2006)
16. Cannon, J., Byers, M.: Compliance deconstructed. *ACM Queue* 4(7), 30–37 (2006)
17. Oberortner, E., Zdun, U., Dustdar, S.: Tailoring a Model-Driven Quality-of-Service DSL for Various Stakeholders. In: *Workshop on Modeling in Software Engineering, MiSE (2009)*
18. Daniel, F., Casati, F., D'Andrea, V., Strauch, S., Schumm, D., Leymann, F., Mulo, E., Zdun, U., Dustdar, S., Sebahi, S., de Marchi, F., Hacid, M.: Business Compliance Governance in Service-Oriented Architectures. In: *Proceedings of AINA 2009*. IEEE Press, Los Alamitos (May 2009)



# Transformation of Intermediate Nonfunctional Properties for Automatic Service Composition

Haruhiko Takada and Incheon Paik

School of Computer Science and Engineering  
University of Aizu, Aizu-Wakamatsu, Fukushima, Japan  
harulin3@gmail.com, paikic@u-aizu.ac.jp

**Abstract.** Service-oriented computing provides an evolving paradigm for flexible and scalable applications of open systems. Web services and their automatic composition are in the mainstream of the evolution of new value-added services. Functional and non-functional aspects are considered together for automatic service composition (ASC). After locating suitable functionality for the required composition, non-functionalities are considered to select the final set of services. Non-functional properties (NFPs) obtained from users or identified during planning or discovery usually have abstract concepts that cannot be identified at the selection stage. In this paper, we propose a transformation technique for automatic composition that identifies binding information in the selection stage from intermediate abstract NFPs. The classification of abstraction level in NFPs, a model to define abstract and concrete NFPs, and an algorithm for transformation from intermediate to concrete level are presented. The identification of the binding information is based on domain ontologies for services. Evaluation in our algorithm according to characteristics of NFPs is shown. Our work will contribute to modeling and identification of NFPs for ASC.

**Keywords:** Automatic service composition, Non-functional property, Constraint, Semantic Web, Transformation.

## 1 Introduction

Service-oriented computing (SOC) enables new kinds of flexible and scalable business applications of open systems and improves the productivity of programming and administering applications in open distributed systems. Web services are already providing useful APIs for open systems on the Internet and, thanks to the semantic Web, are evolving into the rudiments of an automatic development environment for agents. To further this environment, the goal of automatic service composition (ASC) is to create new value-added services from existing services, resulting in more capable and novel services for users.

Services composition traditionally follows a four-stage procedure: planning, discovery, selection, and execution [1]. In the planning stage, an abstract workflow to satisfy the functionalities is created. In the discovery stage, service candidates are located for each task based on functional properties (FPs) or QoS requirements. In the selection

stage, a set of services that fulfill the non-functional properties (NFPs) is selected. Finally, the selected services are executed.

Many efforts have focused on the four stages or integrating them to improve the performance of automatic composition in several aspects. When a user gives a composition goal that includes functional and non-functional requirements to a composer, a sequence of abstract tasks to satisfy the user's functional requirements is generated, together with additional interim non-functional requirements, in the planning stage. The non-functional requirements may be described in high-level abstractions such as natural language, i.e., they do not have binding information, that is input and output of service instance located, that can be used in the selection stage. The composition result, an ordered set of services to satisfy all the requirements, is obtained by selecting the services that fulfill a set of NFPs (requested by the user or generated in the planning stage) from those discovered in the selection stage. The services in the selection stage should be bound and grounded for execution at the next stage, so they require binding information such as identification of service, operator, and parameters. As abstract NFP requirements from the user or the planner do not have this information, they must be related through transformations to the information before entering into the selection stage.

We present an approach for transforming abstract requirements into concrete ones that can be used in the selection stage. We suggest a model for transformation with classification of the levels of abstraction. Extracting proper semantics for requirements from natural-language requests imposes another heavy load of translation. We supply an NFP definition for abstract-level and intermediate-level that our algorithm can transform to concrete ones. Matching operations that bind abstract properties to concrete processes and properties require sophisticated procedures, because the context of non-functionality is based on large dynamics and semantics. Therefore, we exploit a transformation algorithm that works on domain ontologies constructed from information about real service domains.

The rest of the paper is organized as follows. Section 2 describes related work in the area of automatic service composition. Section 3 presents the data model of an NFP in Backus–Naur form (BNF). Section 4 explains the algorithm to transform intermediate level NFPs to concrete ones. Section 5 gives conclusions and mentions future work.

## 2 Related Work

In the selection stage of the four-stage composition architecture [1], optimized services are selected to satisfy overall NFPs (the term constraint is used to describe NFPs in many papers) with binding information. The selection problem is to find a set of services that satisfy the given constraints or boundary conditions from the candidates. Several approaches for selecting services have been investigated, such as the constraint satisfaction problem (CSP), the constraint optimization problem (COP) [2], and linear programming [3]. In those approaches, attribute–value clauses were used for NFPs, where an attribute identifies the NFP involved, and the value is the physical

value of the property offered by the related service. The policy-centered metamodel (PCM) [4] for NFPs was suggested to describe more general characteristics of properties in the real world. PCM can deal with constraint operators including range and can offer clustering. It is based on policies described by a BNF in the ontology and can be applied to general NFPs.

NFPs are described as attribute–value in the form of an ontology in the frameworks of OWL-S [5] and the Web service modeling language (WSML) (in description logic or first-order logic). The Web service modeling ontology (WSMO) [6] tries to add enriched semantic descriptions of existing Web services and defines an explicit conceptual model for semantic Web services based on the Web Service Modeling Framework. WSML was proposed for modeling Web services (functionality and non-functionality), ontologies, and related aspects based on WSMO, together with the formal grounding of the language based on logical formalisms such as description logic (DL), first-order logic (FOL), and logic programming.

Definition of abstractness and terms of constraints has been studied [7]. Our approach is based on the attribute–value scheme, which is the most representative and reasonable description for NFPs. As we defined the essential concept of non-functional characteristics, it can be extended easily. Automatic transformation from abstract NFPs to concrete ones with binding information to support the selection stage is presented in this paper.

### 3 Definition of Non-Functional Property

NFPs of services include QoS requirements, user preferences, and various constraints from internal or external requests. The constraints for QoS requirements consider execution cost, time, reputation, success rate, availability, reliability, and security (encryption, confidentiality), and attributes of each property can be classified as being of types resource or value [7,8]. The user preference constraints describe basic desire, priority, and choice in the form of FOL or DL. They can be an extended general formula to combine the atomic preference formulas. Many studies of service composition use the term constraint for NFP for a service. The term constraint can cover NFPs, QoS, preferences, and other constraints. In our research, we use constraint as a representative term and the attribute–value clause as the basic form for NFPs.

#### 3.1 Level of Constraints as NFPs

Constraints have two kinds of sources at three levels. Internal constraints are defined by the composer, and external constraints are imposed by the user. We divide the characteristics of the constraints into three levels: abstract, intermediate, and concrete. High-level constraints are highly abstract and visceral for users, while low-level constraints can be dealt with in the selection stage because they contain binding information.

**Level 3: Abstract Constraints.** The constraints are at a high abstraction level that is near humans' natural concepts. All terms are abstract, and the constraint may not be

defined in formal terms. They can be in natural language (NL) or may contain several complex meanings in a keyword.

**Level 2: Intermediate Constraints.** The intermediate constraints consist of a relation, two terms, context information, and an operator. They are generated by extracting abstract relations, terms, and context information from abstract terms (which may include context information) in NL or nonterminal terms at level 3. All the terms are terminal and have not yet been bound to concrete terms.

**Level 1: Concrete Constraints.** These have relations, terms as binding information, and indexes of abstract workflow.

### 3.2 Data Model of Property

The data model of composition properties in our ASC is based on the above constraints and other properties for all composition stages.

#### Definition 1: FP and NFP for Planner

*Request* ::= **pDomain** *Parameter*\*

*Parameter* ::= **name** **value**

A *Request* consists of FPs and NFPs for the planner. It is described in simple FOL style. The planner requires a problem domain **pDomain** as FP and parameters *Parameter*\* as FPs and NFPs. Each *Parameter* has a **name** and **value**.

An example of a *Request* is

(*trip* (*departurePlace* Aizu) (*arrivalPlace* SanFrancisco)  
(*priority* *earliness*) (*stay* 3))

This means that a user wants to go from Aizu to San Francisco as soon as possible and stay there for three nights. The *departurePlace*, *arrivalPlace* and *stay* are used as FPs and *priority* as an NFP. If the user sets (*priority* *cheapness*), the planner makes another plan, i.e., not by Shinkansen (high-speed train) but by ordinary railways or highway buses.

#### Definition 2: FP of Abstract Task and Workflow

*AbstractTask* ::= **sDomain** *Parameter*\*

*Workflow* ::= *AbstractTask*\*

An *AbstractTask* is an FP of an abstract task and has service domain **sDomain** and some parameters *Parameter*\*. The **sDomain** is a class in the service domain ontology described in the following section.

An example of an *AbstractTask* is

(*TrainService* (*departureStation* Tokyo) (*arrivalStation* Narita) (*line* NEX))

It means traveling by train from Tokyo to Narita using the Narita Express.

And a *Workflow* is sequence of *AbstractTask*(s).

**Definition 3: Abstract Constraint**

$$\text{AbstractConstraint} ::= \text{Relation NonTerminalTerm}_1 \text{ NonTerminalTerm}_2$$

$$\text{NonTerminalTerm} ::= \mathbf{w}_1 \mathbf{w}_2 \mathbf{w}_3 \dots \mathbf{w}_n$$

$$\text{Relation} ::= = | \neq | > | < | \geq | \leq$$

In this paper, an abstract constraint *AbstractConstraint* has two nonterminal terms *NonTerminalTerm* and one relation *Relation*. *NonTerminalTerm* has possibility that term has compound meaning. For example, *TotalCost* has means of the summation and meaning of the price. An example of an *AbstractConstraint* is

$$(< \text{TotalCost } 300000 \text{ yen})$$

It means that a user wants total cost is less than 300,000 yen.

**Definition 4: Intermediate Constraint**

$$\text{IntermediateConstraint} ::= \text{Relation IntermediateTerm}_1 \text{ IntermediateTerm}_2$$

$$\text{IntermediateTerm} ::= (\mathbf{vDomain} \text{ Context Operator}) | \mathbf{value}$$

$$\text{Operator} ::= \mathbf{sum} | \mathbf{average} | \mathbf{max} | \mathbf{min} | \mathbf{additional-function}$$

$$\text{Context} ::= \mathbf{index} | \mathbf{sDomain}$$

An intermediate constraint *IntermediateConstraint* has two intermediate terms *IntermediateTerm* and one relation *Relation*. An *IntermediateTerm* has a variable domain **vDomain**, a context term *Context*, and a term operator *Operator*, or there may be an intermediate term with a constant value **value**. The **vDomain** is a class of variable in the ontology shown in Section 4.3. The *Context* has two types of index: one is a pure index **index** for pointing and the other is a service domain **sDomain** that will give information through inference on the service domain ontology. The **index** points to an *AbstractTask*. The *Operator* is an aggregating operator such as **sum**, **average**, **max**, **min**, or some **additional-function**. This operator calculates using variables pointed to by **vDomain** and *Context*. The **sDomain** of the *Context* points to some abstract task belonging to this domain.

An example of *IntermediateConstraint* is

$$(< (\text{Cost AllServices sum}) 300,000\text{yen})$$

It has same meaning to abstract constraint's ones. This means that the total cost of all services in the abstract workflow must be less than 300,000 yen.

**Definition 5: Concrete Constraint**

$$\text{ConcreteConstraint} ::= \text{Relation ConcreteTerm}_1 \text{ ConcreteTerm}_2$$

$$\text{ConcreteTerm} ::= (\text{Operator Variable}^*) | \mathbf{value}$$

$$\text{Variable} ::= \mathbf{index} \mathbf{vDomain}$$

*IntermediateConstraint*\* are transformed to concrete constraints *ConcreteConstraint*\* by the transformer for the selection stage. A *ConcreteConstraint* has two concrete terms *ConcreteTerm* and a *Relation*. A *ConcreteTerm* has an index for a task

in the abstract workflow *index* and a variable domain in the ontology **vDomain**, or there may be a concrete term with constant value **value**.

An example of *ConcreteConstraint* is

(= (*nil* (2 *Seat*)) *economy*)

It means the *Seat* variable of the second abstract task is *economy* class. This concrete constraint is made from the following *IntermediateConstraint*.

(= (*Seat* *Airplane* *nil*) *economy*)

It means that the seat in an airplane must be *economy* class.

### Definition 6: Binding Information

*Variable* ::= **index** **vDomain**

*Candidate* ::= *Service*\*

*Service* ::= **sReference** *Attribute*\*

*Attribute* ::= **vDomain** **value** **aReference** | **pReference**

The candidate generator makes some service candidates *Candidate*\* as binding information. The number of *Candidate*\* is the same as the number of *AbstractTask*\*. An *AbstractTask* corresponds to a *Candidate*. A *Service* is selected by CSP from *Candidate* to satisfy *ConcreteConstraint*\*. A *Service* has binding information. The **sReference** is a reference to a service such as the URL of WSDL and some attributes. *Attribute*\* are input parameters and the output operator of a Web service. An *Attribute* has **vDomain**, **value**, and reference to a process **pReference** or parameter **aReference**. When the CSP solver receives a value from a service, if the **value** of *Attribute* is input, this **value** is used, but if the **value** of *Attribute* is not input (the input is *nil*), **pReference** is used as an operator of a Web service, and **aReference** is used as the parameter to obtain a value.

### 3.3 Construction of Ontology

Ontologies for the service domain, variable domain, operator, and pointing method are used in a transformer. Each Web service must belong to a class in the service domain ontology. Each parameter and operator of a Web service must belong to a class in the variable domain ontology. A child class in the service domain includes all attributes of its parent class. The classes, relations, and their hierarchy in the variable domain ontology can be used for several inference operations to extract knowledge about service variables. The ontology for operators includes all aggregating operators, such as *Summation* and *Average* that the service selector can treat, together with their aliases. The ontology for pointing includes a relational index for a workflow, such as *First* and *Last*. Our current transformation algorithm uses the ontologies to include all classes of the services and the variables being transformed. According to the characteristics of the various services domains, the ontologies for the domains can be changed.

## 4 Discovering Binding Information : Transformation of NFPs

To transform an intermediate constraint into concrete constraints, we must find binding information from the intermediate terms. In detail, the procedure must find a service identification (or index) in the abstract workflow (here, the workflow is presumed sequential) and information about the operation and parameters of the service. We presume that the intermediate constraint can be obtained from users directly, or by translating abstract constraints to the intermediate level.

The following algorithm transforms an intermediate constraint to some concrete constraints that have binding information. Let  $CC$  be a set of concrete constraints as a transformation result. Let  $it$  be an *IntermediateTerm* in this constraint. Let  $r$  be a *Relation* in the constraint.  $CC$  is constructed as:

$$CC = r \times \text{transform}(it_1) \times \text{transform}(it_2).$$

The function *transform* is shown in Algorithm 1 in Fig. 1 and transforms an intermediate term into concrete terms.

**Algorithm 1.** *transform(it)*  
**if**  $it$  is a constant value  
     **return**  $it$   
**else if**  $it.context$  is an *index*  
     **return**  $(nil, (it.context, it.vDomain))$   
**else if**  $it.operator = nil$   
     **return**  $\{ct: ct = (nil, (index\ of\ t, it.vDomain)),$   
          $t \in workflow, t.Context \subseteq it.Context\}$   
**else**  
     **return**  $\{(it.operator, \{v: v = (index\ of\ t,$   
          $it.vDomain), t \in workflow, t.Context \subseteq it.Context\})\}$   
**end**

**Fig. 1.** Algorithm for Transformation

The function *transform* diverges according to the context and the operator. In the first two cases, the intermediate term has constant or direct index information, and the transformation can be done simply. If an intermediate term has a constant value, it becomes a concrete term that has the same constant value. If an intermediate term has an index, it becomes the service identification index of a concrete term. In the next two cases, there is no direct information for service identification, so an inference on the service and the variable domain ontologies is required. Currently, our implementation matches the corresponding class by querying the class hierarchy of the ontology. If an intermediate term has no operator (only one service is involved) and a service domain, it becomes a concrete term that has a variable. The variable has an index of an abstract task in the service domain of the intermediate term. If an intermediate term has an operator and a service domain, the transformer collects all the services related to the operator in the workflow and applies the operator to the related services.

## 5 Example of Transformation Flow

In this section, we show an example of transformation flow using the travel scenario. There is a station at departure location A, and it connects to a station B at an airport (for departure). There is another airport (for arrival) at a location C. There are three trains from A to B on railroad line AB, and three airplanes from B to C. There are two hotels in C.

In the train, the fare for the line AB is 3000 and its timetable is described in the table 1. Three airplanes departure from Airport B at 11:00 and the cost and seat are described in the table 2. Information of two hotels in C is described in the table 3.

**Table 1.** Service candidates for trains from station A to station B

	Station A	Station B
Service <sub>1</sub>	7:10	8:10
Service <sub>2</sub>	9:10	10:10
Service <sub>3</sub>	11:10	12:10

**Table 2.** Service candidates for airplanes form the airport B to the airport C

	Seat	Cost
Service <sub>4</sub>	Economy	100,000
Service <sub>5</sub>	Business	200,000
Service <sub>6</sub>	Economy	150,000

**Table 3.** Service candidates for hotels in location C

	Cost (for 3 nights)
Service <sub>7</sub>	30,000
Service <sub>8</sub>	60,000

First, a user requests as follows.

*I want to make a trip from A to C after 2009/11/23 8:00 and stay 3 days in a hotel in C. Seat for airplane is economy. Total cost is less than 150,000.*

Next, a natural language processor translates it to some structural properties as follows.

*Request = (Trip ((departurePlace A)(arrivalPlace C) (stay 3)))*

*IntermediateConstraint<sub>1</sub> = ((TimeFrom 1 nil) >2009/11/23 8:00)*

*AbstractConstraint<sub>1</sub> = (SeatAirplane = economy)*

*AbstractConstraint<sub>2</sub> = (TotalCost < 150,000)*

Next, we presume that two abstract constraints were translated by a translation method. As an example, nonterminal term "TotalCost" is to be translated to intermediate



terms. All candidates of an intermediate term that is made to compound “TotalCost” are  $(TotalCost\ AllServices\ nil)$ ,  $(Total\ AllServices\ Cost)$ ,  $(Cost\ AllServices\ Total)$ ,  $(Total\ Cost\ nil)$  and  $(Cost\ Total\ nil)$ . The confided intermediate term is only  $(Cost\ AllServices\ Total)$  because “Cost” is in the variable domain ontology and “Total” is in operator ontology that describes “Total” is same to class “Summation”. Therefore the nonterminal term is translated to  $(Cost\ AllServices\ Summation)$ . Finally, two abstract constraints are translated as follow.

$IntermediateConstraint_2 = ((Seat\ Airplane\ nil) = economy)$

$IntermediateConstraint_3 = ((Cost\ AllService\ Sumation) < 150,000)$ .

Next, the planner generates an abstract workflow as follows.

$AbstractTask_1 = (train\ (departureStation\ stationA)(line\ lineAB)$   
 $(arrivalStation\ stationB)))$

$AbstractTask_2 = (airplane\ ((departureAirport\ airportB)\ (arrivalAirport\ airportC)))$

$AbstractTask_3 = (hotel\ (place\ C))$

Two interim intermediate constraints for temporal connection are generated by the planner.

$IntermediateConstraint_4 = ((TimeTo\ 1\ nil) > (TimeFrom\ 2\ nil))$

$IntermediateConstraint_5 = ((TimeTo\ 2\ nil) > (TimeFrom\ 3\ nil))$

The transformation algorithm creates concrete constraints from intermediate constraints according to the ontology.

$ConcreteConstraint_1 = (((1\ TimeFrom)) nil) > 2009/11/23\ 8:00)$

$ConcreteConstraint_2 = (((2\ Seat)) nil) = economy)$

$ConcreteConstraint_3 = (((1\ Cost)(2\ Cost)(3\ Cost)) sum) \leq 150,000)$

$ConcreteConstraint_4 = (((1\ TimeTo)) nil) < ((2\ TimeFrom)) nil))$

$ConcreteConstraint_5 = (((2\ TimeTo)) nil) < ((3\ TimeFrom)) nil))$

Let us illustrate the transformation of the intermediate constraint  $IntermediateConstraint_2$  to a concrete one. The meaning of the constraint is that the user wants an economy class seat on the airplane. The relation ‘=’ and constant *economy* are used in the concrete constraint without change.  $(Seat\ Airplane\ nil)$  is an intermediate term that will be transformed. *Airplane* is domain *sDomain*. The only abstract task pointed to by the domain is  $AbstractTask_2$  because  $AbstractTask_2.sDomain$  belongs to *Airplane*. The index of the new concrete term is therefore set to 2. *Seat* is the class name of the *variable domain ontology*. It will be used in the concrete term without change. In this case, only one abstract task has the service domain *Airplane*.

Finally, service candidates described in the tables 1, 2 and 3 and concrete constraints are solved by CSP solver.  $Service_2$  is selected to an  $AbstractTask_1$  by CSP solver because  $Service_1$  does not satisfy  $ConcreteConstraint_4$  and  $Service_3$  does not satisfy  $ConcreteConstraint_1$ .  $Service_4$  is selected to an  $AbstractTask_2$  because  $Service_5$  is not economy class and if  $Service_6$  is selected, summation of cost is over.  $Service_7$  is selected to an  $AbstractTask_3$  because if  $Service_8$  is selected, summation of cost is over. Therefore, the concrete process is  $(Service_2, Service_4, Service_7)$  If user allow this plan, the composer books to use these concrete services.

## 6 Conclusions and Future Work

Planning, discovery, and selection, and also translation and transformation of NFPs, are all important phases for ASC. We defined the level of NFPs, a data model of the abstract-level, intermediate-level, and concrete-level NFPs. We also presented a novel transformation algorithm to generate concrete constraints from intermediate constraints. The algorithm works well for intermediate constraints that are defined on the service domain ontology. The accurate translation of constraints that have highly abstract terms to intermediate constraints is important issue to solve abstractness for ASC too.

Future work will address the following issues. First, algorithm to translate abstract-level NFPs to intermediate-level ones will be studied. Second, more semantically complex FPs and NFPs including inference and rules on domain ontology will be defined in the intermediate level for more intelligent transformation.

## References

1. Claro, D.B., Albers, P., Hao, J.K.: *Web Services Composition in Semantic Web Service, Processes and Application*, pp. 195–225. Springer, New York (2006)
2. Hassine, A.B., Matsubara, S., Ishida, T.: *A Constraint-based Approach to Horizontal Web Service Composition*. In: *Proc. of International Semantic Web Conference*, Athens, U.S.A., pp. 130–143 (2006)
3. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: *Dynamic Web Service Composition in METEORS*. In: *Proc. IEEE Int. Conf. on Services Computing*, Shanghai, China, pp. 23–30 (2004)
4. De Paoli, F., Palmonari, M., Comerio, M., Maurino, A.: *A Meta-model for Non-functional Property Descriptions of Web Services*. In: *Proc. IEEE Int. Conf. on Web Services*, Beijing, China, pp. 393–400 (2008)
5. *OWL-S: Semantic Markup for Web Services* (2004), <http://www.w3.org/Submission/OWL-S/>
6. *WSML. The Web Service Modeling Language (WSML). Final Draft* (2008), <http://www.wsmo.org/TR/d16/d16.1/v1.0/>
7. Paik, I., Takada, H.: *Modeling and Transforming Abstract Constraints for Automatic Service Composition*. In: *Proc. of IEEE International Conference on Computer Information Technology*, Xiamen, China, pp. 136–141 (2009)
8. Traverso, P., Pistore, M.: *Automated Composition of Semantic Web Services into Executable Process*. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, pp. 380–394. Springer, Heidelberg (2004)

# Dealing with Fixable and Non-fixable Properties in Service Matchmaking\*

Octavio Martín-Díaz, Antonio Ruiz-Cortés, José M<sup>a</sup> García, and Miguel Toro

Dpto. Lenguajes y Sistemas Informáticos  
ETS. Ingeniería Informática – Universidad de Sevilla  
41012 Sevilla, España – Spain  
octavio@lsi.us.es, {aruiz,josemgarcia,migueltoro}@us.es

**Abstract.** In the context of service discovery, matchmakers check the compliance of service-level objectives from providers and consumers. The problem of bounded uncertainty arises if some property is non-fixable. In this case, the provider is not able to control the value it takes at runtime, so the eventual consumer must not have the choice to select a value and fix it, but only knowing the guaranteed range of values it may take. To the best of our knowledge, there does not exist any approach which deals with this scenario. Most matchmakers work as if all properties were fixable, and a few have assumed the contrary. In either case, the accuracy of their results is likely to be in question since there may be involved both fixable and non-fixable properties at the same time, and there may also exist dependencies between them. In order to improve the accuracy, we present a holistic approach to matchmaking under bounded uncertainty and propose constraint programming as our choice to deal with it, so that matchmaking is transformed into a quantified constraint satisfaction problem.

## 1 Introduction

Quality-aware enhancement of automated service discovery is one of the main challenges of service-oriented computing [13]. In a common scenario, administrators allow providers and consumers to publish their agreement offers (AO) in a repository. AOs specify both the service-level objectives (SLO) guaranteed by the provider to consumers, and the SLOs required by the consumer to providers. SLOs state assertions over service properties, or properties for short. In this context, matchmakers check the compliance, i.e. the existence of commonalities between the AOs from a provider and a consumer to determine if coming to a service-level agreement (SLA) between them is feasible. In general, SLAs regulate the execution of the services and provide guarantees to that end [2].

---

\* This work has been co-supported by the European Commission FEDER and Spanish Govt. under CICYT projects Web-Factories (TIN2006-00472) and SETI (TIN2009-07366), and the Andalusian Administration under project Isabel (TIC-2533).

## 1.1 Fixability and Dependency

The fixability of a property is related to the fact that the provider is able to control the value it takes at runtime. If so, it may be declared as fixable, otherwise it must be declared as non-fixable. In turn, the controllability is related to the ability of the service to adapt itself in order to maintain the objectives which are being guaranteed, adjusting dynamically the use of common resources which are shared with other services [1, 11, 18].

As an example, consider a video streaming service whose provider guarantees a SLO such as  $\text{IMAGE} \in \{320 \times 200, 640 \times 480\}$  where  $\text{IMAGE}$  stands for *the size of the screen*. This property takes a value from a limited set of options, and the provider is usually able to control the requested value regardless of the environment of the service execution. Thus, the property may be declared as fixable, and the consumer has the choice to fix the value, say  $\text{IMAGE} = 320 \times 200$ .

Consider the provider guarantees another SLO as  $\text{FRAMERATE} \in [22..25]$  where  $\text{FRAMERATE}$  stands for *the frame ratio in images/sec to show a video*. In this case, the provider is not usually able to control the value it takes at runtime, being under the influence of the heterogeneous environment of the underlying network. Quality-of-service properties, such as bandwidth, performance, reliability, and availability, are usually dynamic and have an unpredictable nature [13]. This is said to be a scenario of *bounded uncertainty*. At best, the provider is only able to guarantee the range of values the property may take. Thus, the property must be declared as non-fixable, so that the consumer does not have the choice to fix it, but only knowing the guaranteed range of values it may take, e.g.  $\text{FRAMERATE}$  takes any value between 22 and 25 at runtime.

There may be also eventual dependencies between properties. Hereafter, we focus on whether non-fixable properties are dependent on fixable properties, so that the quality-of-service may vary according to the values the fixable properties take. As an example, consider the provider guarantees several SLOs as:

$$\begin{aligned} \text{IMAGE} &\in \{320 \times 200, 640 \times 480\} \\ \text{IMAGE} = 320 \times 200 &\implies \text{FRAMERATE} \in [22..25] \\ \text{IMAGE} = 640 \times 480 &\implies \text{FRAMERATE} \in [20..23] \end{aligned}$$

where the  $\text{IMAGE}$  property is declared as fixable, and  $\text{FRAMERATE}$  is non-fixable. On the one hand, the consumer is allowed to fix a value for  $\text{IMAGE}$  by choosing between  $320 \times 200$  and  $640 \times 480$ . On the other hand, the value taken by  $\text{FRAMERATE}$  is not controllable at runtime, but it depends on the value taken by  $\text{IMAGE}$ , so that if  $\text{IMAGE} = 320 \times 200$  then  $\text{FRAMERATE}$  ranges between 22 and 25 at runtime, else if  $\text{IMAGE} = 640 \times 480$  then  $\text{FRAMERATE}$  ranges between 20 and 23 instead.

## 1.2 Inaccuracy of Matchmaking

Matchmakers may not work properly, mainly due to mis-assumptions taken regarding with fixability. Most approaches treat all properties as if they were fixable, and only a few assume the contrary, i.e. that all properties are non-fixable.

However, the accuracy of their results is likely to be called into question since there may be involved both fixable and non-fixable properties at the same time. This issue is related to the problem of bounded uncertainty, which arises if a non-fixable property is involved in guaranteed SLOs. According to [5], if bounded uncertainty is not properly dealt with, matchmaking may be inaccurate.

Thus, two scenarios are distinguished. Most usually, matchmakers have not taken fixability into account at all. As a matter of fact, they have optimistically defined matchmaking as if all properties were fixable, so that it is simply based on searching for common values between guaranteed and required SLOs. However, once a matching service is found, the consumer may be possibly using the service under non-satisfying values: if some of the properties are actually non-fixable, the provider can not control their values at runtime in order to avoid the values which do not satisfy the required SLOs. This is a matter of false positives.

As an example, consider a provider guarantees a SLO as the following `FRAME-RATE`  $\in [22..25]$ , being the property known as non-fixable, and a consumer states a required SLO as `FRAMERATE`  $\geq 25$ . Under the optimistic scenario, the matching is possible because there exists a common value, say `FRAMERATE` = 25. But this is a false positive because there are also some values which do not satisfy the required SLOs, any value in  $[22..24]$ , that may occur at runtime because the provider is not able to control them.

On the contrary, there are only a few matchmakers which pessimistically assume all properties to be non-fixable [7, 14]. As the provider does not control the values properties take at runtime, matchmaking is based on checking that every value satisfying the guaranteed SLOs also satisfies the required SLOs. However, the matchmaker may be turning down a service which can be used under satisfying values: if some of the properties are actually fixable, the provider will control their values at runtime, hence the values which do not satisfy the required SLOs will be completely avoidable. This a matter of false negatives.

As an example, consider the provider guarantees a SLO as the following `IMAGE`  $\in \{320 \times 200, 640 \times 480\}$ , being the property known as fixable, and a consumer states a required SLO as `IMAGE` = 320x200. Under the pessimistic scenario, the matching is surprisingly not possible (!) because there exists a value, say `IMAGE` = 640x480, which does not satisfy the required SLO. This is a false negative because the other value, `IMAGE` = 320x200, does satisfy the required SLO. As the provider does control it at runtime, the non-satisfying value is avoidable, and thus the provider's SLO should be considered as a match.

### 1.3 Holistic Approach to Matchmaking under Bounded Uncertainty

To avoid inaccurate results, matchmakers must take into account the fact that (1) there may be both fixable and non-fixable properties at the same time, and (2) there may be also eventual dependencies between fixable and non-fixable properties. To the best of our knowledge, there are quite a few approaches which have tackled uncertainty but from different perspectives: non-probabilistic set theory [5], historical records [6], fuzzy ranking [10], predictive and probabilistic models based on workflows [4, 17] or trustworthiness notions [12, 16].

In this paper, we present a holistic approach to matchmaking under bounded uncertainty, incorporating both fixability and dependency of properties, which improves the accuracy of the results. Our contribution is twofold:

- *Problem space.* We provide an abstract, rigorous definition of matchmaking by means of a geometrical interpretation. This makes easier the understanding of matchmaking and the comparison between the basic scenarios, and it justifies the necessity of a more complex, holistic vision of matchmaking.
- *Solution space.* We propose constraint programming as a feasible technique to solve the matchmaking under different scenarios. In particular, we emphasize the use of quantified constraint satisfaction problems [8] as a means to solve the matchmaking under bounded uncertainty.

The rest of the paper is structured as follows. First, Sec. 2 makes an overview of basic scenarios of matchmaking by means of a geometrical interpretation. Then, Sec. 3 presents our holistic approach to matchmaking under bounded uncertainty. Finally, Sec. 4 exposes our conclusions and future work.

## 2 Preliminaries

*Geometrical interpretation.* Because of the multiple ways by which AOs/SLOs can be specified, a geometrical interpretation based on set theory is given in order to abstract from the language to describe them [14]. As illustrated in Fig. 1(a), the AO property domains configure a space of points-of-agreement, or points for short, so that the AO is represented by means of a region in such space. In this region, each point is determined by assigning a value to every property, so that all SLOs in the AO are satisfied as a whole. Thus, each point refers to a possibility, either from the provider or the consumer, to achieve a SLA with the other party. These regions may adopt different shapes, which act as indicators of the degree of expressiveness to specify SLOs.

Concerning expressiveness, symmetry refers to the fact that both guaranteed and required SLOs can be specified with the same expressiveness. Fig. 1(b) illustrates both cases, namely, asymmetric and symmetric SLOs. In the former case, it is usual for guaranteed SLOs to be defined by means of property/value pairs, so that their regions are given by single points in the agreement space.

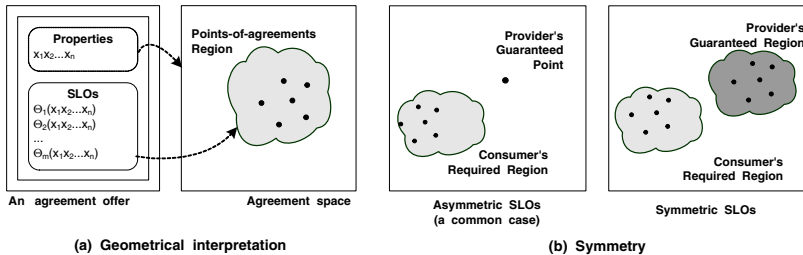


Fig. 1. Geometrical interpretation of agreement offers

Both expressiveness and symmetry have severe influences on matchmaking performance, as shown in the following.

*Optimistic scenario.* Matchmaking is defined as if all properties were fixable. Let  $G$  be the provider's guaranteed region, and  $R$  the consumer's required region. In case of symmetric SLOs, matching is possible provided that the intersection between these regions is non-empty:

$$\text{matches}(G, R) \Leftrightarrow G \cap R \neq \emptyset$$

That is to say, matching refers to searching for a common point between both regions. This scenario is computationally less costlier from an operational perspective, and also much easier to address.

*Pessimistic scenario.* Matchmaking is defined as if all properties were non-fixable instead. Let  $G$  be the provider's guaranteed region, and  $R$  the consumer's required region. In case of symmetric SLOs, matching is possible provided that the former region is a subset of the latter:

$$\text{matches}(G, R) \Leftrightarrow G \subseteq R$$

In other words, matching refers to checking that all points in the guaranteed region belong to the required region. This scenario is much more difficult to address, depending on the shapes which may have the involved regions. The more expressive is the specification of SLOs, the more complex are the shapes which their regions take. To sum up, the comparison of regions with complex shapes is a problem which is computationally costly, so matchmaking is usually reduced to less expressive specifications, leading towards optimistic scenarios. Thus, most approaches restrict regions to be either punctual or having limits parallel to axis.

### 3 Holistic Matchmaking under Bounded Uncertainty

In this section, we present our approach to matchmaking under bounded uncertainty, incorporating both fixability and dependency of properties, as well as taking high expressiveness and symmetry of SLOs into account. As these features can not be separately dealt with, because of the existing relationships among them, a holistic approach is needed.

#### 3.1 The Problem Space

*Fixable points & regions of bounded uncertainty.* Every property is declared either as fixable or non-fixable. Thus, any point-of-agreement may be split into its fixable and non-fixable parts. The former part is specifically called fixable point (FP) since the provider controls the actual values the fixable properties take during the service execution, so that the consumer is allowed to fix them.

There may be also eventual dependencies among properties, so that non-fixable properties take values which are dependent on values which fixable properties take. As a result, a FP is associated with a region of bounded uncertainty (RBU) whose dimensions are given by the non-fixable properties. Thus, a RBU is composed of the non-fixable part of all points which share the same FP.

As an example, consider the AO in Section 1 whose provider declares that IMAGE as fixable, and FRAMERATE as non-fixable. The points are given by the tuples (IMAGE, FRAMERATE) which satisfy the guaranteed SLOs, so that the fixable part is given by (IMAGE) and the non-fixable part by (FRAMERATE). Thus, the FPs are IMAGE = 320x200 and IMAGE = 640x480, so that all values of FRAMERATE between 22 and 25 constitute the RBU associated with IMAGE = 320x200, and all values of FRAMERATE between 20 and 23 constitute the RBU associated with IMAGE = 640x480, as shown in Fig. 3.

*Filtering and projection.* These auxiliary operations are needed to define both FPs and their RBUs, which are illustrated in Fig. 2.

Let  $Z$  and  $X$  stand for properties which are fixable and non-fixable, respectively. Let  $Q$  be a region of points, the region of FPs  $C_Q$  is obtained by projecting  $Q$  on the  $Z$ -axis. In turn, if  $z^*$  stands for a FP in  $C_Q$ , its associated RBU  $N_Q(z^*)$  is obtained by filtering  $Q$  through  $z^*$  and then projecting on the  $X$ -axis.

*Holistic matchmaking under bounded uncertainty.* Matching is interpreted as the search for a FP common to the provider’s guaranteed and consumer’s required regions, so that their associated RBUs also match. Note it adopts the optimistic scenario regarding with FPs, and the pessimistic one regarding with RBUs.

Let  $G$  be the provider’s guaranteed region, and  $R$  the consumer’s required region. Matching is possible provided that the following holds:

$$\text{matches}(G, R) \Leftrightarrow \{z^* \in C_G \cap C_R \mid N_G(z^*) \subseteq N_R(z^*)\} \neq \emptyset$$

where: (i)  $z^*$  is a FP which belongs to the intersection between the provider’s and consumer’s regions of FPs,  $C_G$  and  $C_R$ , respectively, and (ii) the RBUs associated with  $z^*$  match if  $N_G(z^*)$  (the RBU associated with  $z^*$  in the guaranteed region) is a subset of  $N_R(z^*)$  (the RBU associated with  $z^*$  in the required region).

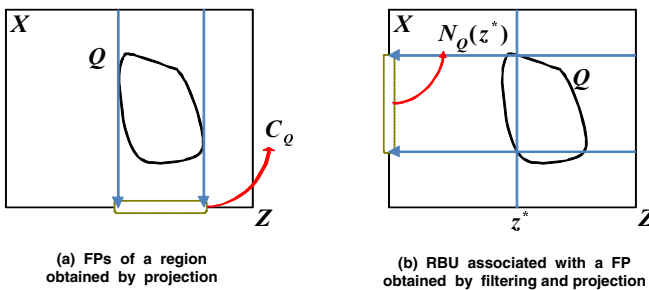
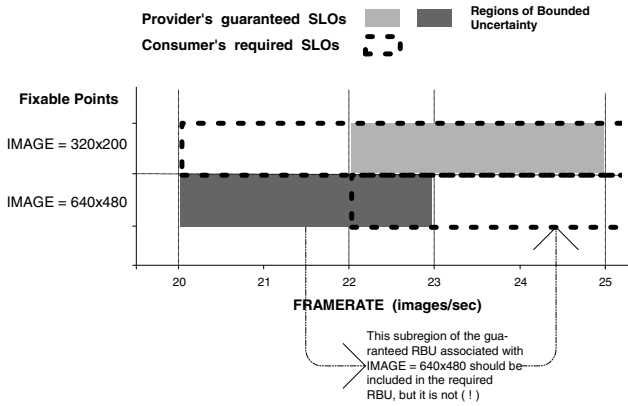


Fig. 2. Filtering and projection





**Fig. 3.** Matching of FPs associated with RBUs

As an example, consider the AO from a consumer whose required SLOs are:

$$\begin{aligned} \text{IMAGE} = 320 \times 200 &\implies \text{FRAMERATE} \geq 20 \\ \text{IMAGE} = 640 \times 480 &\implies \text{FRAMERATE} \geq 22 \end{aligned}$$

It matches the provider’s AO (see Fig. 3) because there exists a FP,  $\text{IMAGE} = 320 \times 200$ , so that each value in the associated guaranteed RBU ( $\text{FRAMERATE} \in [22..25]$ ) belongs to the associated required RBU ( $\text{FRAMERATE} \geq 20$ ). However, the other FP,  $\text{IMAGE} = 640 \times 480$ , is not a match because of the values in the associated guaranteed RBU ( $\text{FRAMERATE} \in [20..23]$ ) which do not belong to the associated required RBU ( $\text{FRAMERATE} \geq 22$ ).

### 3.2 The Solution Space

Constraint programming (CP) [9, 15] is our choice to deal with matchmaking. Most CP solvers are able to process highly-expressive constraints, so that the matchmaking can be usually carried out even if SLOs are not restricted to property/value or property/range pairs. This approach was demonstrated to be feasible in [14] where we presented a pessimistic scenario of matchmaking, which we extend in this paper by considering the new scenario.

*CP in a nutshell.* A constraint satisfaction problem (CSP) is defined by means of a set of variables and their domains, together with a set of constraints specifying which combinations of variables and values, or *solutions*, are acceptable.

Let  $\psi$  be a CSP, it is satisfiable iff its solution space is not empty:

$$\text{sat}(\psi) \Leftrightarrow \text{sol}(\psi) \neq \emptyset$$

Due to the declarative nature of CP, computational procedures to enforce CSPs need not to be programmed. Given a problem, the idea is to solve it by stating a CSP which models the problem itself, and then trying to check its satisfiability by means of a CP solver.

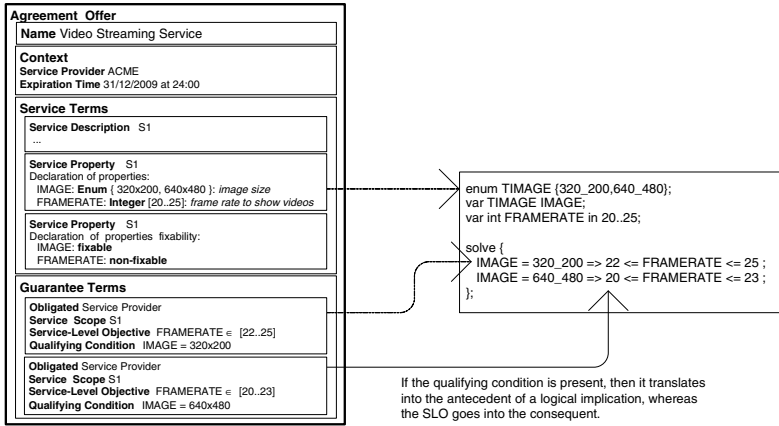


Fig. 4. Translation from an AO to the corresponding CSP

*Translating from AOs to CSPs.* The key point of the approach is the translation from an AO to a CSP. In short, the CSP is obtained by translating AO properties to CSP variables and SLOs to CSP constraints, so that the CSP solution space is equivalent to the AO region. As an example, Fig. 4 illustrates it. On the one hand, the AO is written in a schematic representation of WS-Agreement, the state-of-the-art recommendation by the Open Grid Forum [2]<sup>1</sup>. On the other hand, the CSP is written in OPL, a language designed to specify CSPs [9].

*Transformations for basic scenarios.* Let  $G$  be the guaranteed region by a provider, and  $R$  the required region by a consumer [2], whose respective CSPs are denoted with  $\psi_G$  and  $\psi_R$ . The matching formulae is transformed to a CSP whose satisfiability is checked for a result, as shown in Fig. 5(a).

*Transformations for holistic matchmaking under bounded uncertainty.* Let  $z$  and  $x$  stand for variables corresponding to fixable and non-fixable properties, respectively, so that the expression  $\psi(z, x)$  denotes a CSP based on such variables. The matching formulae is transformed to a CSP which consists of two parts, as shown in Fig. 5(b): (i) the former corresponds to the transformation for the optimistic scenario in order to get the common FPs; (ii) the latter corresponds to the transformation for the pessimistic scenario in order to check whether the guaranteed and required RBUs (associated with such FPs) match.

Unfortunately, checking the satisfiability of this CSP does not yield the expected results. Its parts can not be solved separately since both parts are bound by the  $z$  variables, which stand for the common FPs. The solution lies in transforming the latter part to an equivalent CSP based on universal quantification, in order to properly process the projection on  $z$  variables.

<sup>1</sup> Note a new service term to declare the fixability of properties has been added.

<sup>2</sup> It is assumed both guaranteed and required SLOs are based on the same properties.

	Problem space	Solution space
Optimistic scenario	$matches(G, R) \Leftrightarrow G \cap R \neq \emptyset$	$matches(G, R) \Leftrightarrow sat(\psi_G \wedge \psi_R)$
Pessimistic scenario	$matches(G, R) \Leftrightarrow G \subseteq R$	$matches(G, R) \Leftrightarrow \neg sat(\psi_G \wedge \neg \psi_R)$

(a) Transformations for basic scenarios of matchmaking

	Problem space	Solution space
Scenario of holistic matchmaking under bounded uncertainty	$matches(G, R)$ $\Leftrightarrow \{z^* \in C_G \cap C_R \mid$ $N_G(z^*) \subseteq N_R(z^*)\} \neq \emptyset$	$matches(G, R)$ $\Leftrightarrow sat(\psi_G(z, x) \wedge \psi_R(z, x)$ $\wedge \neg(\psi_G(z, x') \wedge \neg \psi_R(z, x')))$

(b) Transformations for holistic matchmaking under bounded uncertainty

Fig. 5. Transformations for scenarios of matchmaking under study

Consequently, the CSP turns out to be as follows:

$$\begin{aligned}
 matches(G, R) \Leftrightarrow & sat(\psi_G(z, x) \wedge \psi_R(z, x) \\
 & \wedge \forall x' \cdot \psi_G(z, x') \Rightarrow \psi_R(z, x'))
 \end{aligned}$$

Solving a quantified-CSP is a classic problem in CP. Most solutions are based on particular instances of it. As an example, quantified variables can be restricted to be denumerable. In doing so, an iterative solving of CSPs for holistic matchmaking is made easier, and feasible provided that domains of quantified variables (which correspond to FPs) remain as short as possible [3, 8].

## 4 Conclusions and Future Work

In this paper, we have introduced a holistic scenario of matchmaking which deals with bounded uncertainty, incorporating both the fixability and dependency of properties. Our approach improves the accuracy of matchmaking, avoiding both false positives and false negatives which are due to mis-assumptions regarding with the fixability of properties. A CP-based solution has been also proposed to perform the service matchmaking in that scenario, whose feasibility was demonstrated regarding the pessimistic matchmaking in [14], though the empirical evaluation have to be repeated to properly find out about the influences of quantified variables in CSP solving.

Finally, there are some untouched questions. On the one hand, the optimal selection needs also to be studied under this scenario. On the other hand, we are considering other technologies such as soft constraints and fuzzy constraints to be used, whose feasibility to solve problems under uncertainty is well-known.

## Acknowledgements

The authors would like to thank Mr. Carlos Müller, Mr. José Antonio Parejo, and Dr. Manuel Resinas for their helpful discussions, whose comments and suggestions improved the paper substantially.

## References

- [1] Afandi, R., Zhang, J., Gunter, C.A.: AMPol-Q: Adaptive Middleware Policy to Support QoS. In: Dan, A., Lamersdorf, W. (eds.) ICSSOC 2006. LNCS, vol. 4294, pp. 165–178. Springer, Heidelberg (2006)
- [2] Andrieux, A., Czakowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Services Agreement Specification (WS-Agreement) Version 1.1 draft 20 (September 2006)
- [3] Bordeaux, L., Monfroy, E.: Beyond NP: Arc-Consistency for Quantified Constraints. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 371–386. Springer, Heidelberg (2002)
- [4] Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of Service for Workflows and Web Service Processes. *Journal of Web Semantics* 1(3), 281–308 (2004)
- [5] Cheng, W., Wang, H.: Uncertainty-Aware QoS Description and Selection Model for Web Services. In: 4th IEEE Services Computing Conf., pp. 154–161 (2007)
- [6] Cheng, W., Wang, H.: Web Service Decision-Making Model Based on Uncertain-but-Bounded Attributes. In: 4th IEEE SCC Workshop on Semantic Web for Web Services and Processes, pp. 81–86. IEEE CS Press, Los Alamitos (2007)
- [7] Frølund, S., Koistinen, J.: Quality-of-Service Specification in Distributed Object Systems. *Distributed Systems Engineering Journal* 5(4) (1998)
- [8] Gent, I.P., Nightingale, P., Rowley, A., Stergiou, K.: Solving Quantified Constraint Satisfaction Problems. *Artificial Intelligence* 172(6-7), 738–771 (2008)
- [9] Hentenryck, P.: Constraint and Integer Programming in OPL. *Informations Journal on Computing* 14(4), 345–372 (2002)
- [10] Hwang, S.Y., Wang, H., Tang, J., Srivastava, J.: A Probabilistic Approach to Modeling and Estimating the QoS of Web-Services-based Workflows. *Information Sciences* 177(23), 5484–5503 (2007)
- [11] Li, B., Nahrstedt, K.: A Control-Based Middleware Framework for Quality of Service Adaptations. *Journal on Selected Areas in Communications* 17(9), 1632–1650 (1999)
- [12] Mandaric, A., Oberweis, A., Perc, P.: Web Services-based Architecture for Reducing Behaviour and Quality Uncertainties. In: 1st IEEE Conf. on e-Science and Grid Computing, Melbourne, Australia, pp. 320–327 (December 2005)
- [13] Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.: Service-Oriented Computing: A Research Roadmap. In: Dagstuhl Seminar on Service Oriented Computing (2006)
- [14] Ruiz-Cortés, A., Martín-Díaz, O., Durán, A., Toro, M.: Improving the Automatic Procurement of Web Services using Constraint Programming. *International Journal of Cooperative Information Systems* 14(4), 439–467 (2005)
- [15] Tsang, E.: *Foundations of Constraint Satisfaction*. Academic Press, London (1995)
- [16] Vu, L., Aberer, K.: A Probabilistic Framework for Decentralized Management of Trust and Quality. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS (LNAI), vol. 4676, pp. 328–342. Springer, Heidelberg (2007)
- [17] Wang, P., Chao, K.M., Lo, C.C., Huang, C.L., Li, T.: A Fuzzy Model for Selection of QoS-Aware Web Services. In: 2nd Intl. Conf. on e-Business Engineering, Shanghai, China, pp. 585–593. IEEE Computer Society, Los Alamitos (October 2006)
- [18] Wohlstadtter, E., Tai, S., Mikalsen, T., Rouvellou, I., Davanbu, P.: GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions. In: 26th Intl. Conf. on Software Engineering, Edinburgh, Scotland, pp. 189–199. IEEE CS Press, Los Alamitos (2004)

# Using SLA Mapping to Increase Market Liquidity

Marcel Risch<sup>1</sup>, Ivona Brandic<sup>2</sup>, and Jörn Altmann<sup>1</sup>

<sup>1</sup> TEMEP, Department of Industrial Engineering, College of Engineering  
Seoul National University, 599 Gwanak-Ro, Gwanak-Gu, 151-742 Seoul, South-Korea

marcel.risch@temep.snu.ac.kr, jorn.altmann@acm.org

<sup>2</sup> Information Systems Institute, Vienna University of Technology

Argentinierstr. 8/184-1, 1040 Vienna, Austria

ivona@infosys.tuwien.ac.at

**Abstract.** Research into computing resource markets has mainly considered the allocative fairness of market mechanisms. It has not been discussed how a large variety of resource types influences the market liquidity. Markets containing large numbers of buyers and sellers for heterogeneous resources suffer from a low likelihood of matching offers and requests. Traders therefore have the high risk of not being able to trade resources. We suggest a solution that derives SLA templates from a large number of heterogeneous SLAs in the market and, by using these templates instead of the original SLAs, facilitates SLA mapping. The usefulness of this approach is demonstrated through simulation results and a comparison with an alternative approach, in which SLAs are predefined.

**Keywords:** Service level agreements, Cloud computing, commodity goods, SLA matching, market liquidity, market mechanisms, SLA matching.

## 1 Introduction

When developing markets for Cloud resources, it should be considered that the Cloud resources are designed to be liquid. Illiquid goods (which include rare goods and differentiated goods) have a higher risk that they cannot be sold or purchased when needed, driving market participants away [19]. Therefore, markets can only function efficiently with a sufficient number of market participants.

Creating such a market with a large number of Cloud resource traders is far from trivial. Firstly, consumers will only join an open Cloud market, if they are able to find what they need quickly. Secondly, providers will only join the market if they can be fairly certain that the resources they are trying to sell will be sold for the right price. Should either of these conditions not be met, providers and consumers will not participate in the market.

Open Cloud markets face an unusual challenge in this instance. The widespread use of virtualization enables resource providers to create a wide range of tradable resource types. At the same time, resource consumers can also specify their needs very precisely. In such a case, this lack of liquidity will make the market unattractive to consumers and providers.

In this paper, we demonstrate the problem (low liquidity for each resource type) caused by a large number of resource definitions. To counteract this problem, we

introduce an approach, based on SLA mapping, which ensures sufficient liquidity in the market. These SLA mapping techniques not only simplify the search for similar offers but also allow us to derive public SLA templates from all existing offerings. These SLA mappings map parts of a consumer-defined SLA document to a public SLA template. The purpose of SLA mappings is twofold: Firstly, users may discover services with less effort and secondly, based on predefined learning functions and accumulated SLA mappings; the proposed SLA mapping approach facilitates user-driven definitions of public SLA templates, increasing the chances of matching of offers in the future.

Summarizing, the contributions of this paper are: (1) the demonstration of the problem caused by a large number of resource types; (2) an approach to overcome the shortcomings of an open market by introducing an SLA mapping approach; and (3) a first evaluation of the proposed SLA mapping approach considering different strengths and weaknesses.

## 2 Computing Resource Markets

The research into resource markets can be divided into two groups, when looking at their attempts of describing the tradable good. The first group does not define goods at all, while the second group focuses on one aspect of a computing resource only. However, neither group discusses the liquidity of goods in Cloud computing markets.

The first group consists to a large extent of early Grid market designs [1-3, 8]. One such example can be seen in [1], where Buyya et al. describe the entities of a Grid market. However, the question of how the market mechanism can handle a multi-dimensional good has not been answered. Similarly, GRACE [2] is a market architecture for Grid markets and outlined a market mechanism without defining the good “computing resource” or considering that consumers and providers have to agree to the resource specifications.

The second group of Grid market research has simplified the computing resource good. When developing the MACE exchange [8], the authors recognize the importance of developing a definition for the tradable good and abstract computing resources into services which can be traded. However, a detailed specification of a computing resource service has not been given and hence its effects on market liquidity cannot be assessed. The Spawn market was envisioned to work with CPU time slices [9]. While matching demand and supply is trivial, the fact that the resources have been reduced to a single component is not realistic, as the CPU slice requirements depend on the CPU vendor due to different instruction sets. Lastly, the Tycoon market was developed before virtualization tools (e.g. Xen [11]) became widely used [10]. Initially, it worked with basic computing cycles and it was planned to extend this market by making use of virtualization. However, it seems that the effort has been discontinued. The SORMA project focused on fairness and efficient resource allocation [3-6]. The project identified several requirements for Grid markets [7]. However, the requirement analysis has not considered that a market can only function efficiently, if there is sufficient liquidity of goods.

Overall, much of the computing resource market research has worked with either simplified definitions of the tradable good or without defining the good at all. This

means that the issue of maintaining a sufficiently high liquidity (i.e. the likelihood of matching bids and asks) in such a market has not been addressed.

Despite some lacking research, a large number of commercial Cloud providers have entered the utility computing market, offering various types of services. On the one hand, there are resource providers, such as Amazon (e.g. EC2 [12]) and Tsunamic Technologies [13], who provide computing resources. Next, there are providers, who not only sell their own resources but also their own software services, such as Google Apps and Salesforce.com [14-15]. Furthermore, there are companies that attempt to run a mixed approach, i.e. they allow users to create their own services but also offer services themselves., e.g. Sun N1 Grid [16] , Salesforce.com, and Microsoft Azure [17]. Most of these providers have in common that they only sell a single type of resources (or, in the case of Amazon, resource types which are based on a basic instance [18]). This limited number of different resource types enables a market creation, since all demand is channeled towards very few resource types.

### 3 The Liquidity Problem in Markets

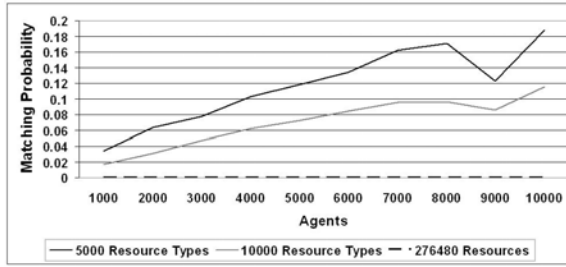
Virtualization makes the use of Cloud computing environments more interesting, since virtualization allows for portability of machine images. For a Cloud market, however, the introduction of virtualization introduces the problem that providers can define their own resource types to differentiate themselves from other providers. The impact of this behavior on liquidity is shown with an example of a double auction market mechanism. This liquidity issue also exists for other market mechanisms.

#### 3.1 Double Auction

There have been a number of proposals for using double auctions for Grid markets [21-23], since double auctions have shown their suitability for non-storable commodities, such as electricity [24]. Contracts (i.e. service level agreements) in the double auction market, that we sketch here, are traded as follows: First, traders send their bids and asks to the market. Second, using the continuous double auction matching procedure, the market attempts to match bids and asks. Once a match is made, both parties receive a contract (i.e. signed SLA) with each other's information. The advantages of using such a contract lie in the fact that simple bids and asks suffice for trading resources.

#### 3.2 Matching Performance for a Double Auction

We have developed a simulation for a double auction market, for which we defined a range of tradable resource types. Any trader could decide at the start of the day, whether resources had to be purchased or could be sold. The demand generator created a normally distributed demand across all traders. The simulation was repeated with various numbers of traders and various numbers of resources types. Each simulation allowed traders to interact over a period of 500 simulated trading days. Using these parameters, we measured the percentage of matched bids to the total number of bids. This measure was chosen, since the absolute trading volume only has meaning, if it can be compared to historical data. Our results are shown in Figure 1.



**Fig. 1.** Matching Probability for Bids

The top line shows the likelihood of matching a bid in a market with 5000 resource types. Initially, the matching probability is low, since the supply and the demand are spread out over many resource types and, hence, it is less likely that a buyer and seller will meet. However, the likelihood increases, as the number of traders in the market grows. To determine how many traders would be needed to achieve a match probability of 75%, we performed a linear regression on the available data and found that about 39,800 traders would be needed. The grey line indicates the likelihood of matching a bid in a market with 10,000 resource types. Again, we can see that the probability is quite low but increases slowly with the number of traders. Using linear regression, we found that about 46,400 traders are needed to achieve a 75% match probability in this market. The dashed line indicates the likelihood of matching a bid in a market where each parameter of a resource can be set by traders, leading to 276,480 possible resource permutations. It can be seen that this probability is quite low and rises very slowly. A linear regression analysis indicated that about 33 million traders would be required to achieve a 75% matching probability for bids.

In general, these results show that, for a market to function properly, a sufficiently large base of traders must exist. Since large numbers of continuously acting traders are uncommon in current resource markets, our results indicate that a market, which sells fewer resource types, has higher liquidity. Next, we will show how resource types can be homogenized through SLA mapping, to get less resource types.

## 4 SLA Mapping

In this section, we explain the SLA mapping approach used to obtain public SLA templates, focusing on the lifecycle of SLA templates, management of SLA mappings, and SLA transformations used for the realization of SLA mappings. Finally, we introduce a marketplace, in which SLA mappings can be used to limit the number of resource types and, consequently, increase liquidity in the computing resource market. Note that automated trading is not required, purchases can be made manually.

### 4.1 The Importance of SLAs in Markets

Based on the outcome of the analysis of the liquidity problems in markets, we are faced with an interesting research challenge. On the one hand, to exploit the potential of open markets, a large number of traders is necessary. On the other hand, the large



number of traders inflates the variety of resources available. Without some form of SLA matching or some way of limiting resource diversity, the set of available computing resources in a market would grow, spreading supply and demand across a wide range of resources, thereby reducing the liquidity of each resource type.

Some current adaptive SLA matching mechanisms are based on OWL, DAML-S [25-26]. Another semantic technology work (by Oldham et al.) describe a framework for semantic matching of SLAs based on WSDL-S and OWL [16]. Dobson et al. present a unified quality of service (QoS) ontology, which is applicable to specific scenarios such as QoS-based Web services selection [24]. Ardagana et al. present an autonomic Grid architecture with mechanisms to dynamically reconfigure service center infrastructures to fulfill varying QoS requirements [27]. Koller et al. discuss autonomous QoS management, using a proxy-like approach for exploiting SLAs to define certain QoS parameters that a service has to maintain during its interaction with a specific customer [28].

None of the presented approaches address the issues of the open market, where traders meet on demand. In most existing approaches, traders have to agree either on specific ontologies [26-27] or have to belong to a specific portal [28]. None of the discussed approaches handle semi-automatic definitions of SLA mappings, allowing negotiations between inconsistent SLA templates. Also, none of the presented approaches allow user-driven definitions of publicly available SLA templates.

### 4.2 The SLA Template Lifecycle

The following figure illustrates the lifecycle of SLA templates. In particular, it shows how public SLA templates are generated and managed.

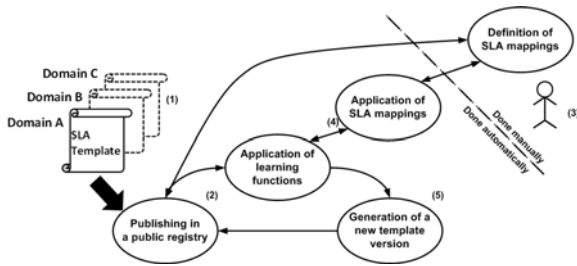


Fig. 2. SLA Template Lifecycle

Figure 2 shows how an initially generated SLA template (step 1), which may not necessarily reflect the needs of users from a specific domain, can be used to derive realistic templates for a new specific domain. As indicated through step 1, we assume that for specific domains, specific SLA templates are initially generated. These generated SLA templates are published in the public registry (step 2). Thereafter, SLA mappings are defined manually by users (step 3). At the same time, learning functions for the adaptation of these public SLA templates are defined. A learning function determines how often a specific SLA mapping had to be used during a predefined time period before the modified SLA template becomes a template stored in the repository (step 5). In such an environment, the application of SLA mappings can also be done automatically, as described in Section 4.3 (step 4).

### 4.3 Managing SLAs

Figure 3 shows the architecture for managing SLA mappings. The registry comprises different SLA templates where each of them represents a specific application domain, such as medicine or telecommunication.

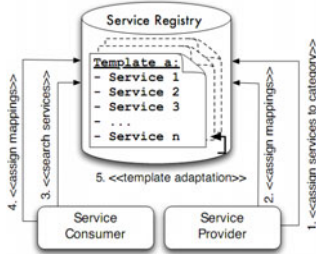


Fig. 3. Template Registry

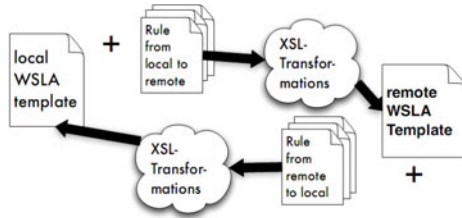


Fig. 4. SLA Transformation

Service providers may assign their service to a particular public SLA template (step 1 in Figure 3) and, if necessary, assign SLA mappings afterwards making modifications to the original template (step 2). Service consumers may search for the services using meta-data and search terms (step 3). After finding appropriate services, each service consumer may define mappings (i.e. modifications) to a posted template (step 4). Besides, public SLA templates should be updated frequently according to predefined adaptation rules to reflect the actual SLAs used by traders (step 5). To reduce the mapping cost for traders though, SLA mappings should also be executed automatically and public SLA templates should be defined and deleted dynamically.

### 4.4 SLA Mapping

To explain the SLA mapping, we describe how the transformation from modified public SLAs to the final SLA template is conducted. Figure 4 depicts a scenario for defining XSL transformations of SLAs expressed in the Web Service Level Agreements (WSLAs) specification language.

Public SLA templates are published in a searchable registry, from which traders may download and compare them with their local SLA template. If any differences are discovered, traders may write rules for transforming (XSL transformation) the local WSLA template to the remote template. The rules are stored in the database and can be applied at runtime to the remote WSLA template.

Figure 4 shows how negotiations can be performed using SLA transformation. Each of the two parties generates a WSLA. The locally generated WSLA plus the rules defining transformations from the local WSLA to the remote WSLA deliver a WSLA, which is compliant with the remote WSLA. Reversing this process, the remote WSLA plus the rules defining transformations from the remote to the local WSLA deliver a WSLA, which is compliant to the local WSLA. Thus, the negotiation may be done between non-matching WSLAs in both directions: from service consumer to service provider and vice versa. Thus, both parties may match on a publicly available WSLA template. To facilitate matching, SLAs are transferred into a canonical form as described in [29].

For example, in a double auction market, traders generate their own WSLA plus the rules for transforming their local WSLA to the market WSLA (i.e. the WSLA of the auction marketplace). The market has a learning function which can determine which of the WSLAs best matches the requirements of the traders. Based on this result, new public WSLAs can be created or old ones can be removed.

### 4.5 Application of the Public SLA Template in Double Auction Markets

In a double auction market environment, the market participants trade using public SLA templates, which describe the traded computing resources, the software running on these resources, the Terms of Use, and the price [30]. In Figure 5, a graphical representation of such a tradable public SLA template is shown.

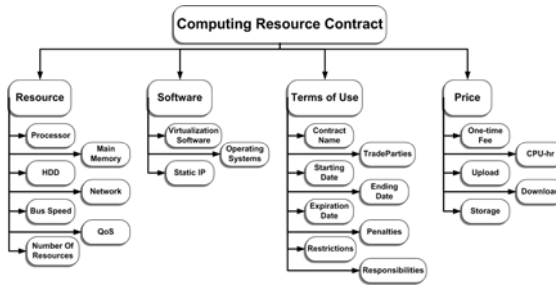


Fig. 5. A Sample SLA Template Obtained from a Learning Function

A market trading only a small number of public SLA templates encourages consumers to map their demand to the existing SLA templates, and providers to map their supply to the public SLA templates. However, once the market is operating, provisions can be made to add additional SLA templates (as new resources become available) and to remove SLA templates. Two rules are suggested to address this issue: First, a new tradable SLA is to be added, if the learning function determines that there is sufficient level of demand for this type of SLA template. Second, the removal of SLA templates depends on the trading volume: If the traded volume falls beneath a threshold for a certain time period, the SLA template will be removed.

Trading of SLA templates has to take into account that specific starting times and ending times are parameters of the templates. Standardization of starting times and usage durations provides one solution. Standardization for usage durations is fairly trivial, since it will become apparent that certain resources are commonly used for certain durations and less popular for other durations. Focusing on these popular periods will be sufficient for successful standardization. The learning function described previously can be used to determine popular starting and ending times.

## 5 Discussion and Validation

The following table gives an overview of the advantages and disadvantages of the SLA Mapping approach compared to an approach, in which the SLAs are predefined.

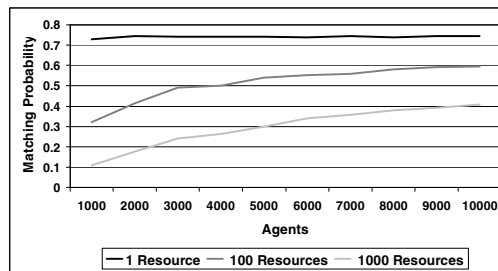
**Table 1.** Strengths and Weaknesses of SLA Mappings and Standardized SLAs

Approach	Strengths	Weaknesses
SLA Mapping	<ul style="list-style-type: none"> <li>- No explicit resource limitations</li> <li>- Flexible SLA usage</li> <li>- Increased liquidity</li> <li>- User-driven approach</li> </ul>	<ul style="list-style-type: none"> <li>- SLA translations are time intensive</li> <li>- System abuse by submitting fake SLA mappings</li> </ul>
Standardized SLA Templates	<ul style="list-style-type: none"> <li>- Clearly defined resources</li> <li>- Adaptability of Standardized SLAs</li> <li>- Increased liquidity</li> </ul>	<ul style="list-style-type: none"> <li>- Standardized SLAs may prevent users with unusual resource profiles to join the market</li> </ul>

The advantage of the SLA mapping approach is that users face few limitations if they determine their resource needs. They can define their SLA templates flexibly. The SLA transformation then determines those public SLA templates that minimize the differences to all user-defined SLAs. However, the SLA translations are time-intensive, which can be a problem if the SLAs are large. Furthermore, users can submit fake SLA mappings which have to be detected by a learning function.

### 5.1 Consequences of Few Resource Types

Figure 7 shows the gains that can be achieved by transforming many heterogeneous SLAs into a few public SLA templates, which can be traded in a double auction. This figure was obtained using the same simulation environment, which was used to demonstrate the liquidity problems earlier.

**Fig. 7.** Matching Probability for Bids with Fewer SLA Templates

The top line shows the likelihood of matching a bid for a market with one tradable SLA template. The likelihood is high (75%), since supply and demand is channeled to a single resource type. The next lower line shows the likelihood of a bid being matched for a market with 100 tradable SLA templates. Using linear regression, we calculated that a market with about 12,000 traders would have a matching probability of 75%. The lowest line shows the likelihood of a bid being matched in a market with 1000 tradable SLA templates. Using linear regression, we have calculated that, to reach a matching probability of 75%, 17,000 traders would be needed.

The matching probability of about 75% for a market with a single resource type seems quite low. The main cause of a lower matching probability was that bids had to

be satisfied by a single ask, since multi-tier applications usually must be closely co-located. The impact of duration and number of resources has been outlined in [20].

Concluding, double auction markets for computing resources require lots of standardization. Having many similar resource types distributes demand and supply, thereby threatening the liquidity. If resources are standardized, supply and demand can be channeled to fewer resources. Despite the reduction in overall utility caused by standardizing, standardization will increase liquidity. Our proposed SLA mapping approach creates standardized goods that can be traded in computing markets.

## 6 Conclusion

We have demonstrated that a market trading diverse computing resources can suffer from a reduced market liquidity. Since this problem makes a market extremely unattractive to traders, we have developed an approach to ensure that the market liquidity remains high. Our approach allows finding public SLA templates that are closest to the SLA templates defined by the user, therefore meeting the needs of users best. This approach is very flexible, since it adapts to the changing needs of the users. We also described the rules established within our approach to ensure that new SLA templates can be added and old SLA templates can be removed. We believe, based on the simulation results and the comparison with an alternative approach, that our approach will prove to be valuable in the development of an open Cloud market and will ensure sufficient liquidity to attract traders.

## References

1. Buyya, R., Vazhkudai, S.: Compute Power Market: Towards a Market-Oriented Grid. In: First IEEE International Symposium on Cluster Computing and the Grid, CCGrid, p. 574 (2001)
2. Buyya, R., Abramson, D., Giddy, J.: An Economy Grid Architecture for Service-Oriented Grid Computing. In: 10th IEEE International Heterogeneous Computing Workshop, HCW 2001. IEEE Computer Society Press, Los Alamitos (2001)
3. The SORMA project (2009), <http://sorma-project.org/>
4. Nou, R., Julia, F., Guitart, J., Torres, J.: Dynamic Resource Provisioning for Self-adaptive Dynamic Heterogeneous workloads in SMP Hosting Platforms. In: International Conference on e-Business (2008)
5. Amar, L., Muallem, A., Stoesser, J.: On the Importance of Migration for Fairness in Online Grid Markets. In: International Conference on Autonomous Agents and Multiagent Systems (2008)
6. Amar, L., Barak, A., Levy, E., Okun, M.: An On-line Algorithm for Fair-Share Node Allocations in a Cluster. In: IEEE International Symposium on Cluster Computing and the Grid, CCGRID (2007)
7. Neumann, D., Stoesser, J., Weinhardt, C.: Bridging the Grid Adoption Gap – Developing a Roadmap for Trading Grids. In: Bled eConference, Merging and Emerging Technologies, Processes, and Institutions (2007)
8. Schnizler, B., Neumann, D., Veit, D., Weinhardt, C.: Trading Grid Services - A Multi-Attribute Combinatorial Approach. European Journal of Operational Research 187(3), 943–961 (2008)

9. Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., Stornetta, W.S.: Spawn: A Distributed Computational Economy. *IEEE Transactions on Software Engineering* 18(2), 103–117 (1992)
10. Lai, K., Rasmusson, L., Adar, E., Zhang, L., Huberman, B.A.: Tycoon: An Implementation of a Distributed, Market-Based Resource Allocation System. *Multiagent Grid Systems* 1(3), 169–182 (2005)
11. XenSource, Inc. (2009), <http://xen.org/>
12. Amazon Elastic Compute Cloud (Amazon EC2) (2009), <http://aws.amazon.com/ec2/>
13. Tsunamic Technologies Inc. (2008), <http://www.clusterondemand.com/>
14. Google Apps (March 2009), <http://www.google.com/apps/>
15. Salesforce.com (March 2009), <http://www.salesforce.com>
16. Sun Grid (2009), <http://www.sun.com/service/sungrid/index.jsp>
17. Microsoft Azure (2009), <http://www.microsoft.com/windowsazure/>
18. Amazon EC2 Instance Types (2008), <http://aws.amazon.com/ec2/instance-types/>
19. Samuelson, P.A., Nordhaus, W.D.: *Economics*, 18th edn. McGraw-Hill/Irwin (July 2004), ISBN 0072872055
20. Altmann, J., Courcoubetis, C., Stamoulis, G.D., Dramitinos, M., Rayna, T., Risch, M., Bannink, C.: GridEcon - A Market Place for Computing Resources. In: Altmann, J., Neumann, D., Fahringer, T. (eds.) *GECON 2008*. LNCS, vol. 5206, pp. 185–196. Springer, Heidelberg (2008)
21. Weng, C., Lu, X., Xue, G., Deng, Q., Li, M.: A Double Auction Mechanism for Resource Allocation on Grid Computing Systems. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) *GCC 2004*. LNCS, vol. 3251, p. 269. Springer, Heidelberg (2004)
22. Kant, U., Grosu, D.: Double Auction Protocols for Resource Allocation in Grids. In: *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005)*, vol. 1(1), pp. 366–371. IEEE Computer Society, Washington (April 2005)
23. Weng, C., Li, M., Lu, X.: Grid Resource Management Based on Economic Mechanisms. *Journal of Supercomputing* 42(2), 181–199 (2007)
24. European Energy Exchange (2008), <http://www.eex.com/en/>
25. Dobson, G., Sanchez-Macian, A.: Towards Unified QoS/SLA Ontologies. In: *Proceedings of the IEEE Services Computing Workshops, SCW 2006*, Chicago, Illinois, USA, pp. 18–22 (September 2006)
26. Oldham, N., Verma, K., Sheth, A.P., Hakimpour, F.: Semantic WS-Agreement Partner Selection. In: *Proceedings of the 15th International Conference on World Wide Web, WWW 2006*, Edinburgh, Scotland, UK (May 2006)
27. Ardagna, D., Giunta, G., Ingraa, N., Mirandola, R., Pernici, B.: QoS-Driven Web Services Selection in Autonomic Grid Environments. In: *International Conference Grid Computing, High Performance and Distributed Applications, GADA 2006*, France (November 2006)
28. Koller, B., Schubert, L.: Towards Autonomous SLA Management Using a Proxy-Like Approach. In: *Multiagent Grid Systems*, vol. 3(3). IOS Press, The Netherlands (2007)
29. Brandic, I., Music, D., Dustdar, S.: Service Mediation and Negotiation Bootstrapping as First Achievements Towards Self-Adaptable Grid and Cloud Services. In: *Grids Meet Autonomic Computing Workshop, GMAC 2009*. In conjunction with the 6th International Conference on Autonomic Computing and Communications, Barcelona, Spain (June 2009)
30. Risch, M., Altmann, J.: Enabling Open Cloud Markets Through WS-Agreement Extensions. In: *Service Level Agreements in Grids Workshop*, in conjunction with *GRID 2009*, Banff, Canada. CoreGRID Springer Series (October 2009)

# Translation of Service Level Agreements: A Generic Problem Definition

Constantinos Kotsokalis<sup>1</sup> and Ulrich Winkler<sup>2</sup>

<sup>1</sup> Dortmund University of Technology, Germany  
constantinos.kotsokalis@udo.edu

<sup>2</sup> SAP Research, Belfast, UK  
ulrich.winkler@sap.com

**Abstract.** A Service Level Agreement (SLA) is the electronic equivalent of a real-life contract, which describes expectations from a service and governs its consumption. Ideally, a SLA provides certainty as regards customer experience and Quality of Service (QoS) received. For self-contained, isolated services this type of certainty is relatively straightforward to achieve. However, for services that are composed by others, or that rely on others to execute, such functional dependencies imply similar non-functional ones. Therefore, SLAs offered by a service to its customers are in turn depending on other SLAs, which the same service establishes in its role as a customer of the services it relies upon. In this paper we explore this dependency between different SLAs, and formalize the problem of converting an agreement for a composed service into individual agreements for the services from which it is composed.

## 1 Introduction

Service composition and re-use is the cornerstone of Service-Oriented Computing. Through service composition, it becomes possible to take advantage of the expertise that each provider offers through its products. At the same time, the provisioning, execution and access of a service may presume the existence of infrastructural services, which are used explicitly or implicitly. In both cases, there are dependencies between the composite (or higher-level) services, and the composed (or infrastructural) services. In the text that follows, we will refer to the former as *dependents*, and the latter as *antecedents*. Such dependencies *may* directly affect the operation of a dependent if non-functional expectations of its antecedents are not met. For instance, a company's web server will not be accessible to the outside world if the company's network fails. If, however, such a company is connecting to the internet through two different links and one of them fails, then the web server will still be accessible to the rest of the world through the second connection. Clearly, finding the dependencies and documenting them in detail is not always straight-forward, especially in complex systems with a multitude of inter-dependent services.

Nevertheless, when a provider needs to augment its services with guarantees towards its clients, this automatically means that guarantees must also be provided by the respective antecedents. The exact way that such guarantees are

related is use-case specific and cannot be universally defined for an application domain, as is also evident from the example of the previous paragraph. However, a formalization of the problem is much needed as a first step for translating the guarantees that a dependent service offers to guarantees that it requires from its antecedents. This paper provides a means to address this open scientific problem, using *Service Level Agreements* (SLAs) as an instrument. SLAs are electronic contracts, that specify what a customer should expect from a service he/she consumes. These expectations may concern both functional and non-functional service properties, which are grouped in a single reference document (the SLA) as it results from negotiation between the parties involved.

The scientific contribution of the present work is *a generic definition of the SLA translation problem*, and *an abstract description of how it needs to be addressed during negotiation time*.

The paper is organized as follows: Section 2 provides information on prior related work. Section 3 discusses the topic of service dependencies, followed by Section 4 that elaborates on the dependencies of service properties and therefore SLAs. Next, Section 5 discusses SLA translation. Finally, Section 6 concludes the paper with a summary and outlook on future work.

## 2 Related Work

Although not directly related to SLAs, there is prior art that concerns relationships between different services and/or resources. In [1], the authors describe a *Functional* and a *Structural* dependency model, on which they base their work. They provide a specific classification of dependencies, into a number of dimensions such as locality, time, type, dependency strength and criticality, etc. Then, building on the two models, they discuss a method for dependency analysis. In [2] the same authors append the two models with an *Operational* one, and further refine a proposed architecture for managing dependencies. The main differences with our work is that they constrain theirs in the ICT domain where they also provide an architecture to tackle the dependency discovery problem. Additionally, they do not consider SLA management.

In [3] the author discusses the problem in a generic manner, much like our paper. However, the work is tied to software services; more specifically, it proposes a scheme where services declare their dependencies on other services based on access patterns (how often are services accessed, in what sequence, etc) with dynamic service selection in mind. Additionally, it looks specifically at functional dependencies. Contrary to that, dependency management in the context of SLAs requires a generic view that handles non-functional dependencies as well.

[4] is looking at the same problem as regards network services. The authors describe a model called *Inference Graph* that represents the dependencies. Based on that, they present an algorithm for inferring probabilistically the malfunctioning components of the network, given real-world observations. The main difference with our work is that the Inference Graph describes dependencies based on service states, and probabilities that the state of one depends on the state



of another; additionally, the referenced work is explicitly addressing network services.

In [5] the authors introduce an agent-based automated SLA negotiation architecture and discuss efficient search-based algorithms to determine an acceptable Service Level Agreement in a multi-agent environment. The main difference to our work is that we assume a complex and layered multi-tier services landscape that requires additional analysis steps and related data-structures for SLA negotiation. In addition, we consider SLA translation as an essential part of the SLA negotiation process. Although some of the search-based algorithms analyzed in [5] would be able to cope with multiple objectives that negotiation agents may have, it is not the primary concern of the authors' work.

In a purely IT context, there have been efforts to address similar problems for modeling, analyzing and converting metrics and properties (e.g. response time and throughput) into concrete system configuration descriptors (e.g. number of servers or system policies). Performance engineering, capacity planning and configuration management are IT areas concerned with this specific problem. Layered queuing network (LQN) models [6,7] are, for example, an important modeling and analysis concept in performance engineering. LQNs are used to model layered client-server architectures, where a server can become a client to other servers and servers are mapped onto resources, such as CPUs or storage. Analytic solvers or simulations are used to estimate resource utilization. Other approaches are probabilistic models, reinforcement learning [8] and Bayesian networks classifiers [9] to map system properties and metrics to system configurations. The main difference to our work is that we address a formal specification of SLA metric-to-metric translation problem in service oriented architectures, and we do not target capacity planning or performance engineering specifically for IT systems.

### 3 Service Dependencies

As already mentioned in Section [1], service dependencies may be *explicit*, or *implicit*. Explicit dependencies play a central role in Service-Oriented Computing. The assumption here is that a service's logic (the functionality it delivers) depends on some other service by means of composition. If we assume that a service implements an algorithm, and that an algorithm is represented by a series of atomic operations (*instructions*) as shown in Figure [1], then one or more of those instructions are carried out by the services on which this former service depends on. A typical example are business processes, usually modeled as workflows, that bind to external services and invoke them during execution. Complete outsourcing of one's work is also an example; in this case, a service provider who cannot satisfy the demand of its customers, acts as a customer to another service provider and delegates some of its work to this latter provider. A service modeled as part of a composition may or may not be actually used, depending on the composition's logic. It is use-case-specific details which define if a service is invoked or not. Still, dependencies exist in the form of *candidate invocations*.

Figure 2 illustrates a composed service (“*Composition*”) that relies on four different services (“*A*”, “*B*”, “*C*” and “*D*”). The instructions that correspond to the composite service are representing the glue code for the composition, which performs invocations, data staging and similar tasks.



Fig. 1. A series of abstract instructions implementing an algorithm

Implicit dependencies are, typically, dependencies related to infrastructure services (without which a higher-layer service cannot operate at all), and services used for reasons of redundancy. For instance, any web server presumes that a working DNS system is available so that clients can access it. Any software relies on some (physical or virtual) hardware on which it can execute. When it comes to redundancy, the standby services set up for this reason are normally not used, but their existence affects the service that depends on them, in the case of a failure. Figure 2 illustrates the concept, with two infrastructure services used from the higher-level software services.

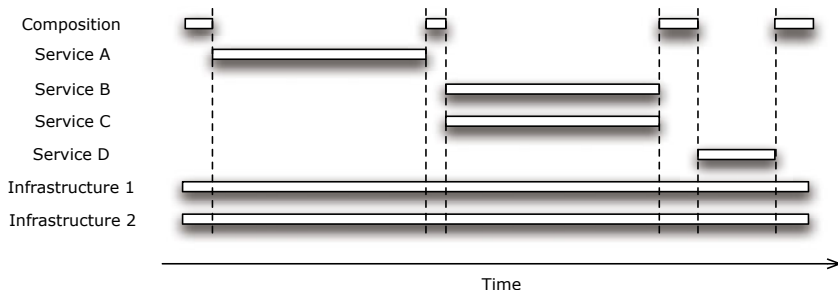
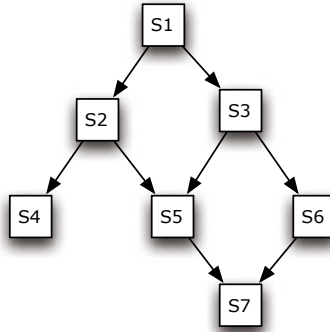


Fig. 2. Example service dependencies: Compositions and infrastructure

Although we can perform this rough classification of explicit and implicit dependencies, it is not possible to refine it further and try to identify in greater granularity the relations between dependent services. Rather, the notion of *dependency* itself is enough to construct a *dependency graph* [10,11] as a first step. This dependency graph can then be elaborated in the context of SLAs, as we will see in the upcoming sections, and function as a basis for the translation process. Figure 3 illustrates an example service dependency graph, where each vertex represents a dependency (direction is from dependent to antecedent).



**Fig. 3.** Example service dependency graph

## 4 Service Properties Dependencies

We define a SLA as a set of guarantees over the consumption of a service. Often, these guarantees may be some facts universal to the SLA, or may refer to properties of the service that the SLA concerns. For instance, if we assume a software service, its availability is one of its properties. A threshold for this availability in the form of a guarantee towards customers, may constitute part of a SLA.

Dependency of a service upon others, naturally means that its properties are depending on properties of these other services. Assuming the availability example for a depending service, it will be affected by the availability of its antecedents. However, there might be a case where a dependent service's property may be related to a completely different property of its antecedents. For instance, the cost of a service is typically affected by the quality of the services that it uses for completing its tasks. Additionally, it is certainly possible to have a SLA which does not refer to all properties of a service, and therefore we only need to take into account a limited number of dependencies: only those which affect the properties mentioned in the SLA.

It is becoming clear that there cannot be a universal classification of service properties, and their dependencies. Certainly, facts that set the context of a SLA cannot apply to all SLAs either, but we should rather assume distinct facts for distinct SLAs. We can only define the problem generically based on the abstraction of conditions that need to hold true, when aggregated according to the SLA structure. Use-case-specific knowledge needs to be applied by domain experts to instantiate these conditions, and associate them in the context of different, dependent services.

Starting from the service dependency graph, we can make a next step towards a service *Properties Dependency Graph* (PDG). Let us assume a service  $S_i$ , with properties  $P^{S_i} = (p_1^{S_i}, p_2^{S_i}, \dots, p_m^{S_i})$ , and a service  $S_j$ , with properties  $P^{S_j} = (p_1^{S_j}, p_2^{S_j}, \dots, p_n^{S_j})$ . If  $S_i$  depends on  $S_j$ , then we can formulate a dependency of  $S_i$ 's  $r$ -th property as a function  $f_r$  of properties of  $S_j$ :

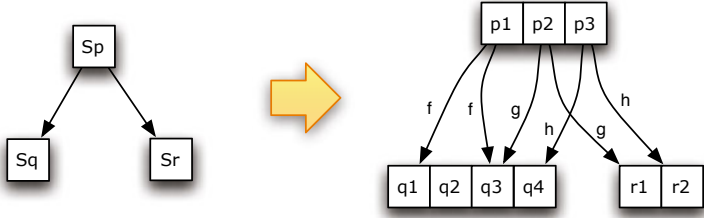


Fig. 4. Example service Properties Dependency Graph

$$p_r^{S_i} = f_r(p_1^{S_j}, p_2^{S_j}, \dots, p_q^{S_j}) \tag{1}$$

where  $1 \geq r \geq m$ , and  $1 \geq q \geq n$ .

This definition can be extended for a service  $S$  that depends on more than one services. Let us assume that  $S$  has properties  $p_1^S, p_2^S, \dots, p_N^S$ . Let us also call the antecedents  $S_1, S_2, \dots, S_m$ , with property sets  $P^{S_i} = (p_1^{S_i}, p_2^{S_i}, \dots, p_{N_i}^{S_i}), 1 \leq i \leq m$ . Then, each property  $p_r^S$  of service  $S$  can be generically expressed as follows:

$$p_r^S = f_r(p_1^{S_1}, \dots, p_{N_1}^{S_1}, p_1^{S_2}, \dots, p_{N_2}^{S_2}, \dots, p_1^{S_m}, \dots, p_{N_m}^{S_m}) \tag{2}$$

Figure 4 illustrates such an example. Here, service  $Sp$  depends on services  $Sq$  and  $Sr$ . Domain experts can then create rules about the dependencies of  $Sp$ 's properties  $p1-p3$  on those of  $Sq$  ( $q1-q4$ ) and  $Sr$  ( $r1-r2$ ), based on modeling techniques, simulation, or real-world observation. These rules are represented by functions  $f, g$  and  $h$  that implement a mapping of each property on the properties it depends.

## 5 SLA Translation

Operating under the abstractions from Sections 3 and 4, we can discuss about the concept of *SLA translation*. In essence, SLA translation is the process of analyzing a SLA in relation to the PDG, applying heuristics and pre-existing knowledge, and coming up with one or more subsequent SLAs for antecedents. These subsequent SLAs provide reasonable (but not complete) certainty that the top-level SLA will not be violated, unless some of them are violated too. We assume all SLAs to be negotiated using some well-specified language and protocol, such as WS-Agreement [12] – probably with an extension that supports counter-offers.

Starting from Equation 2, one can first define a complete set of such equations, one for each property that the customer of the dependent service requires in the negotiated SLA. Properties that the customer does not require guarantees for, can be excluded. Then, this system must be solved, taking into account existing constraints. For that, we presume that services expose *SLA templates*, which are used exactly to guide the negotiation process with initial values, and indicate

the constraints of negotiable parameters. For instance, a provider may be unable (or unwilling, for business reasons) to accept a SLA request about a service with the “*availability*” property set to 100%, although availability level is up for negotiation. In this case, the provider would publish a template, where the upper limit would be clearly stated, e.g. 99.5%. The limits as regards what can be negotiated help customers to reduce the search space for their offers. This search space may otherwise be so large that the problem becomes practically infeasible from a computational point of view.

Optimization of subsequent SLAs is one more requirement for the translation process. As a matter of fact, optimization is part of translation, as it affects the final output. The system mentioned above will typically accept plenty of different solutions, and it is up to optimization to select the best of all those candidates. As we deal with many properties simultaneously, this becomes a multi-criteria optimization problem [13,14].

Using the example from Figure 4, we have the following relations:

$$\begin{aligned} p1 &= f(q1, q3) \\ p2 &= g(q3, r1) \\ p3 &= h(q4, r2) \end{aligned}$$

If an incoming SLA for service  $Sp$  refers to all three properties, a SLA negotiation mechanism which includes translation functions should first find out the dependencies of these three properties based on the PDG. We assume, as also mentioned earlier, that these dependencies are known as domain-specific expertise encoded into our system. For instance, if the property is “*availability*”, it will typically depend on the availability of all antecedents; this is fairly straightforward to assume. However, if we examine cost, it is not necessary that the cost of invoking service  $Sp$  is related specifically to cost properties of services  $Sq$  and  $Sr$ . On the contrary, it may be the case that there are no cost properties for these two services, but rather that their providers only apply flat-rate pricing. In this case, the cost of  $Sp$  may rely on properties such as Quality of Service characteristics of  $Sq$  and  $Sr$ .

The next action should be to find acceptable value spaces for all of  $(q1, q3, q4, r1, r2)$ . By “acceptable”, we refer to values which remain within constraints set inside the templates of the two lower-level services, and on the same time satisfy the requested values in the offer for the higher-level SLA.

The last step during the translation process, would be to come up with an optimal solution to the problem, according to a multi-criteria optimization algorithm, which perhaps takes into account business objectives such as profit maximization. It may well be the case that there exists no single optimal solution. In this case, one of the solutions on the *Pareto front* should be chosen. The Pareto front is a set of all those solutions that are considered to be optimal in multi-criteria optimization. More formally, they are solutions where none of the included criteria can be improved (accept a better value), without some other criteria in the same solution receiving a worse value.

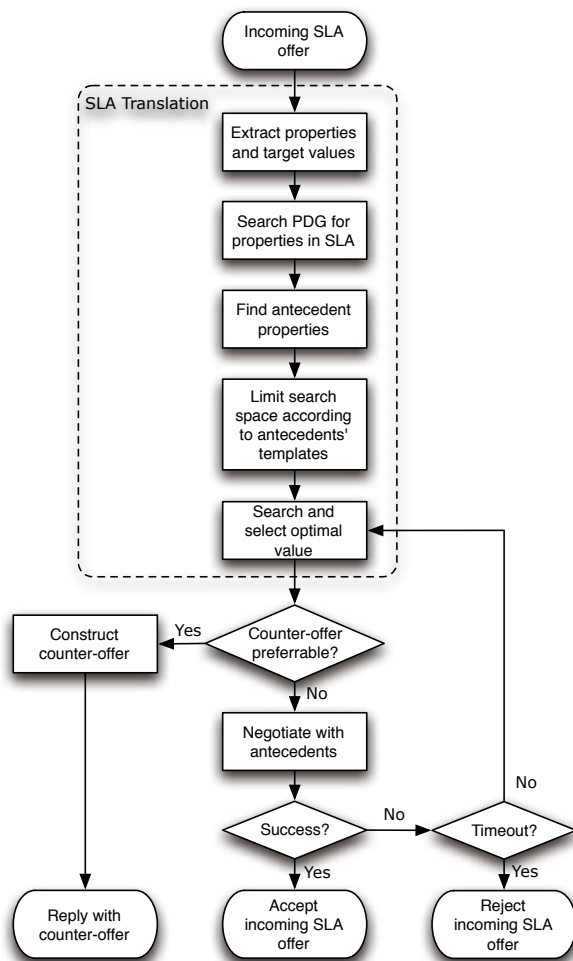


Fig. 5. Generic flowchart for translation/negotiation of SLAs

Figure 5 shows how this translation process can be implemented according to the previous discussion. We are assuming that counter-offers are possible, as part of the negotiation protocol being used. The reason that we are interleaving translation and negotiation in this flowchart, is exactly that they take place in parallel and depend on each other. Negotiation feeds translation with confirmations whether the translation’s results are acceptable; and translation provides negotiation with input according to existing domain-specific knowledge, and any further intelligence implemented within the system. It should be clarified, at this point, that we are separating between the negotiation process and all kinds of decision making; that is to say, we assume negotiation concerns only the exchange of messages in such a way that implements correctly a specific protocol, while all

planning functionality lives elsewhere and is, in this case, part of the translation process. Regarding planning itself, we anticipate that coping with multiple service level attributes and  $n$ -dimensional Pareto fronts might become difficult. We assume that –in order to be able to implement an efficient and automated SLA translation and negotiation framework– use-case dependent heuristic functions will be required to narrow the search space down to a feasible size.

One last thing to mention here, is that in this case we are tackling the problem of producing lower-level SLAs starting from a higher-level one, in a top-down approach. A different scenario is that of a bottom-up approach, where the lower-level SLAs are already established and used to decide whether an incoming higher-level offer can be satisfied. Although this is a valid and realistic scenario in the business world, in this paper we have set to define the process for a completely dynamic setup of SLA hierarchies, which may help to avoid under- and over-provisioning. Nevertheless, the essentials of our previous discussion on producing and using PDGs for SLA translation can be applied just as well on such a bottom-up process, as they are modeling service relationships without being tied to a specific usage scenario.

## 6 Conclusions

In this paper we discussed the topic of service dependencies, and how they reflect onto dependencies of respective SLAs. More specifically, we described service dependencies in terms of Dependency Graphs, and extended this data structure into Properties Dependency Graphs. Based on the latter, we defined what SLA translation means, and illustrated its relation to SLA negotiation.

The process of translating SLAs for dependent services into SLAs for antecedents requires domain-specific and use-case specific expertise, for producing the conversion rules and for fine-tuning them. In some occasions, historical data may provide insight, while in others this will be a modeling task.

As part of our continuing work within the SLA@SOI EU project, we are researching structures and algorithms that can express these concepts and ease the computational task of producing optimal SLA offers towards antecedents, during negotiation. An architecture that takes into account these requirements has already been produced, and is being currently implemented.

## Acknowledgements

The research leading to these results is supported by the European Community's Seventh Framework Programme (FP7/2007-2013) and the SLA@SOI Integrated Project under grant agreement no.216556.

## References

1. Keller, A., Blumenthal, U., Kar, G.: Classification and computation of dependencies for distributed management. In: IEEE Symposium on Computers and Communications, p. 78 (2000)

2. Keller, A., Kar, G.: Determining service dependencies in distributed systems. In: IEEE International Conference on Communications (ICC 2001), vol. 7, pp. 2084–2088 (2001)
3. Hasselmeyer, P.: Managing dynamic service dependencies. In: 12th International Workshop on Distributed Systems: Operations & Management (DSOM 2001), pp. 141–150 (2001)
4. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D.A., Zhang, M.: Towards highly reliable enterprise network services via inference of multi-level dependencies. In: SIGCOMM 2007: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 13–24 (2007)
5. Di Nitto, E., Di Penta, M., Gambi, A., Ripa, G., Villani, M.: Negotiation of service level agreements: An architecture and a search-based approach. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, p. 295. Springer, Heidelberg (2007)
6. Woodside, C.M., Neilson, J.E., Petriu, D.C., Majumdar, S.: The stochastic rendezvous network model for performance of synchronous client-server-like distributed software. *IEEE Trans. Computers* 44, 20–34 (1995)
7. Petriu, D., Woodside, M.: Analysing Software Requirements Specifications for Performance. In: Third Int. Workshop on Software and Performance, pp. 1–9 (2002)
8. Tesauro, G., Jong, N.K., Das, R., Bennani, M.N.: On the use of hybrid reinforcement learning for autonomic resource allocation. *Cluster Computing* 10(3), 287–299 (2007)
9. Kumar, V., Schwann, K., Iyer, S., Chen, Y., Sahai, A.: A State Space Approach to SLA based Management. In: IEEE/IFIP NOMS (2008)
10. Katzela, I., Schwartz, M.: Schemes for fault identification in communication networks. *IEEE/ACM Transactions on Networking* 3(6), 753–764 (1995)
11. Gruschke, B.: Integrated event management: Event correlation using dependency graphs. In: Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 1998), pp. 130–141 (1998)
12. Open Grid Forum: Web Services Agreement Specification, WS-Agreement (2007)
13. Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of Multiobjective Optimization. *Mathematics in Science and Engineering*, vol. 176. Academic Press, London (1985)
14. Ehrgott, M.: *Multicriteria Optimization*. Springer, New York (2005)



# Ontology-Based Feature Aggregation for Multi-valued Ranking

Nathalie Steinmetz<sup>1,2</sup> and Holger Lausen<sup>2</sup>

<sup>1</sup> Semantic Technology Institute (STI) Innsbruck, University of Innsbruck,  
Technikerstrasse 21, A-6020 Innsbruck, Austria

`nathalie.steinmetz@sti2.at`

<sup>2</sup> seekda GmbH, Grabenweg 68, A-6020 Innsbruck, Austria

`Holger.Lausen@seekda.com`

**Abstract.** In the last years we see a clear trend in the Computer Science area to a move towards Service-Oriented Architectures (SOAs). Research in the service domain encompasses its whole life-cycle, including topics as creation, discovery, selection, ranking and composition. This paper focuses on the ranking of discovered Web Services, proposing a novel approach based on non-functional properties of services: information that is available about services by analyzing their description that is available on the Web, their hyperlink relations, monitoring information, etc. The approach is making use of semantic technologies, aggregating the various real-world service aspects as described above in a unified model and providing different rank values based on those aspects.

## 1 Introduction

Within the last years we could observe a clear trend both on the Web and in Computer Science in general to a move towards Service-Oriented Architectures (SOAs). While the Web of documents as we used to know it is changing to a Web of services, the traditional software domain is altering as well: software starts being seen more and more as resource that itself is a service in a cloud (following the software model 'Software as a Service', SaaS). Using Web Service technologies all possible functionalities can be exposed and flexibly integrated in all kinds of applications (e.g., traditional software as well as Web pages). This way Web Services provide new means for interoperability of business logics.

Research in the Web Service domain encompasses the whole service life-cycle, including topics as creation, discovery, selection, ranking and composition. This paper focuses on the ranking aspect, proposing a novel approach based on non-functional properties of services: information that is available about services by analyzing their description that is available on the Web, their hyperlink relations, and similar aspects. After discovery of services, ranking is one of the most important steps to support them in selecting the most best-fitting service fulfilling their needs. Regarding the large size of publicly available services on the Web (e.g., more than 28.000 Web Services found by the seekda Web Service search

engine<sup>[1]</sup> and more than 5.000 Web APIs and Mashups published on the specific Web API portal ProgrammableWeb<sup>[2]</sup>, it would be a tedious task for a human to filter out the services that are relevant to him.

Many current service ranking approaches work with the assumption that services are either semantically well-described or that we have detailed Quality of Service (QoS) information about them available (e.g., availability and response time of services). [5] addresses the ranking of services based on the semantic description of their non-functional properties, including aspects like locative, temporal, availability, obligation, price, payment, etc. This approach has the downside that the required, rather complex semantic descriptions are rather hard for non-experts to provide. [6] follows an approach for QoS-based ranking with trust and reputation management. This approach assumes that QoS properties such as availability, acceptable response time, through-put, etc. are provided by the service provider.

In contrast to these approaches, that require the active participation of the service providers to describe their services in one way or another, we base our ranking approach on the 'real-world' information that is available about them, like their descriptions, their hyperlink relations, some monitoring information, etc. We do not assume unrealistically - that we know how a service behaves on execution, what functionality he delivers, etc. (as would be the case in a man-in-the-middle approach, where we would assume to have such knowledge before doing the ranking).

Most publicly available Web Services are published using either the WSDL (Web Service Description Language)<sup>[3]</sup> standard or following a RESTful (Representational State Transfer)<sup>[1]</sup> approach. Our ranking approach supports both types of services and is targeted to work over a large number of services crawled from the Web and lightweight annotations automatically aggregated from them. [3] provides more information on the service crawling and information aggregation approach.

The rest of this paper is structured as follows: in Section [2] we describe the aggregation of features, based on ontologies, that is underlying our ranking approach. Section [3] provides our ranking algorithms, Section [4] gives a short overview of how the ranking approach is implemented and Section [5] finally concludes the paper with an overview of future work.

## 2 Feature Aggregation Based on Ontologies

This section describes the data aggregation for our ranking approach. We use semantic technologies to aggregate various aspects related to Web Services in a unified model, aspects that encompass information that is available about services by analyzing their description and their hyperlink relations, by talking to their

<sup>1</sup> <http://webservices.seekda.com/>

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

<sup>3</sup> <http://www.w3.org/TR/wsdl>

hosting server, etc. We do not rely on handcrafted, manually added, information, but only take into account real-world information that is anyway available.

As already mentioned in the introduction, the services are gathered by crawling the Web. Together with the services the Web is fostered for related documents, e.g. service descriptions, help pages, etc. The data that is resulting from the crawler comes together with RDF metadata that describes amongst others the relation from services and their related documents (in the case of WSDL services) or that tells us to what extent we believe that a certain Web resource is a Web API (in the case of RESTful services). More details on the crawl data and the corresponding RDF metadata can be found in [4]. For our multi-value ranking approach we take into account aspects like the number and the quality of related documents, classification scores of Web APIs, live monitoring data and metrics from the WSDL descriptions, as e.g. how much documentation is provided for a service. Based on our crawling experience and on our work on the seekda Web Service search engine (<http://webservices.seekda.com>) we see that the aspects upon which we base the ranking approach are realistically available: from more than 28.000 publicly available services approximately 20.000 contain relations to other Web resources; around one fourth of all WSDL descriptions contain some documentation on the service or operation level; all available services are monitored on a daily basis by the seekda search engine.

In the following we will first outline what RDF metadata we use for ranking; next we will describe the WSDL metrics and the monitoring data that we build upon and will in a last step provide an overview on existing and new ontologies that we use for modeling the ranking. The data described in this section is used for the Multi-valued ranking approach as described in Section 3, where we will describe in detail the calculation of the ranking.

We use the following namespaces and prefixes in the above mentioned subsections:

- Service-Finder Service Ontology - `sf`:  
`http://www.service-finder.eu/ontologies/ServiceOntology#`
- seekda Crawl Ontology - `sco`:  
`http://seekda.com/ontologies/CrawlOntology#`
- seekda Ranking Ontology - `sro`:  
`http://seekda.com/ontologies/RankingOntology#`
- XML Schema - `xsd`: `http://www.w3.org/2001/XMLSchema#`

**Crawl Meta-data.** In the case of WSDL services (and related resources) the meta-data delivered by the crawler consists mainly of annotations to the single Web documents, tying them on the one side to a service and describing on the other side of what kind the relation to the service is. As information for ranking we will use (a) the number of related documents per service, and (b) the kind of relation from document to service. The meta-data is stored using elements of the Service-Finder Service Ontology, as shown in Listing 1 as RDF triples.

```

<sf:DirectInLink > <sf:isAboutEntity> <sf:Service>
<sf:DirectInLink > <sf:belongsToDocument> <sf:Document>
<sf:DirectOutLink> <sf:isAboutEntity> <sf:Service>
<sf:DirectInLink > <sf:belongsToDocument> <sf:Document>
<sf:TermVectorSimilarityAssociation> <sf:isAboutEntity>
  <sf:Service>
<sf:TermVectorSimilarityAssociation> <sf:belongsToDocument>
  <sf:Document>

```

**Listing 1.** WSDL service and related documents meta-data used for ranking

In the case of Web APIs the crawl metadata describes some specific features of the Web document (e.g. number of external links, number of camel-case tokens) and provides (a) single scores that specify to what extent the two crawl classifiers (see [3]) believe that a given resource is a Web API and (b) a confidence score that is built from the single scores for convenience reasons. For ranking we will use (a) the Web API Confidence score of a document and (b) which classifier has classified the document as Web API. The meta-data is stored using elements of the Service-Finder Service Ontology and of the seekda Crawl Ontology, as shown in Listing 2 as RDF triples.

```

<sf:DocumentAnnotation> <sf:hasScore> <xsd:number>
<sf:Document> <sco:hasWebAPIConfidenceScore> <xsd:number>
<sf:Annotation> <sf:source> <sf:Agent>
<sf:Annotation> <sf:isAboutEntity> <sf:AnnotatableEntity>
<sf:DocumentAnnotation> <sf:belongsToDocument> <sf:Document>

```

**Listing 2.** Web API meta-data used for ranking

**WSDL Metrics.** A WSDL describes a Web Service from an operational point of view: services, their operations, messages, message formats, endpoints, network bindings, etc. Here the documentation of the single elements is worth being taken into account for ranking: a well documented WSDL improves the ranking of the corresponding service. We take into account the documentation of the service and of the operations. The data is stored using elements of the Service-Finder Service Ontology, as shown in Listing 3 as RDF triples.

```

<sf:Service> <sf:hasDescription> <xsd:string>
<sf:Operation> <sf:hasDescription> <xsd:string>
<sf:Service> <sf:implementsInterface> <sf:Interface>
<sf:Interface> <sf:hasOperation> <sf:Operation>

```

**Listing 3.** WSDL meta-data used for ranking

**Monitoring Information.** Interesting criteria for service ranking are related to Quality of Service information. One such information is the availability of services, i.e. their liveness. This data is monitored and stored by seekda (Web Service search engine at <http://webservices.seekda.com/>) on a daily basis. The availability is based upon the endpoint of a service and is only available for WSDL services. Monitoring the liveness of a service does not mean that the functionality of the service is tested in any kind; it expresses whether the server

where the service is hosted is reachable or not, checks at the same time whether the server is correctly implementing the SOAP protocol, whether the page needs an authentication, and more, based on the HTTP response codes.

We work with the average (percentage) availability of a service over the last 6 months, the last month and the last week (if possible). Listing 4 shows the elements that we use from the Service-Finder Service Ontology to store this data.

```
<sf:Endpoint> <sf:availabilityLast6Months> <xsd:number>
<sf:Endpoint> <sf:availabilityLastMonth> <xsd:number>
<sf:Endpoint> <sf:availabilityLastWeek> <xsd:number>
<sf:Service> <sf:hasEndpoint> <sf:Endpoint>
```

**Listing 4.** Monitoring meta-data used for ranking

**Ranking Ontologies.** To structure and store the data as described above we rely, as already mentioned, on ontologies. Where possible, we reuse the Service-Finder Service Ontology and the seekda Crawl Ontology. Furthermore we have developed a new seekda Ranking Ontology that allows us to express the new ranking specific information that is not yet expressible within the other two ontologies (shown in Listing 5).

```
<sf:Service> <sro:numberOfRelatedDocuments> <xsd:number>
<sf:Service> <sro:numberOfRelatedDocsRank> <xsd:number>
<sf:Service> <sro:numberOfWebAPIScoreRank> <xsd:number>
<sf:Service> <sro:numberOfWSDLMetricRank> <xsd:number>
<sf:Service> <sro:numberOfMonitoringRank> <xsd:number>
<sf:Service> <sro:numberOfGlobalRank> <xsd:number>
```

**Listing 5.** seekda Ranking Ontology

All ranking values are expressed by numbers between 0 and 1, 1 being the best-possible rank. We calculate a rank for each of the criteria that we take into account from the crawl data and the monitoring. Then we calculate a final rank for each service. Details on how the ranks are calculated will be provided in the following section [3](#).

### 3 Multi-valued Ranking

In Section [2](#) we have outlined what aspects of services we aggregate to build our ontology-based feature aggregation for multi-valued ranking approach and what ontologies we use. The fact that we have all the service meta-data that we need for our ranking available as semantic data allows each semantic-aware client to build its own ranking based on the same service meta-data creating individual rules (e.g. using SPARQL). In the following we will present the way we combine the data gathered as described in Section [2](#) to get a global ranking value.

#### 3.1 Rules for a Global Multi-valued Rank

The ontology-based feature aggregation for multi-valued ranking approach differs for the two types of services that we support: WSDL services and Web APIs.

For WSDL services we first calculate three independent ranking values (based on crawl meta-data, WSDL metrics and monitoring data) that are then combined to one global rank. For Web APIs we so far only take into account the Web API confidence score. The following subsections present the rules that we apply to calculate the single rank values, described in a procedural pseudo-code. It is nevertheless as well possible to describe and implement the rules in a declarative language, or, e.g., using SPARQL.

**Related Documents Rank.** This rank is based on the crawl meta-data that is delivered by the crawler, as shown in Listing 1 and will be calculated based on the following information and assumptions:

- How many related documents does a service have? We need to check the document annotations that belong to a service and then count the unique documents that are tied to the annotations. We improve the ranking of a service that is related to other Web resources as in that case the probability is higher that the service contains some relevant documentation, pricing information, etc.
- How is the document related to a specific service? After manual analysis of a number of services and their related documents we got to the conclusion that important information bits are mostly contained in the services' inlinks (i.e., pages that link to the WSDL document), as well as in pages that are related via term vector similarity (e.g., pages that speak about the service but are not related to it via a direct link).

In a first step we thus need to calculate the number of related documents per service. To do so it is not enough to just take the number of document annotations as one document might have several annotations (e.g. a document that has a `DirectOutLink` annotation and a `TermVectorSimilarityAssociation`). We need to first extract all annotations to get the identifiers of all documents that correspond to them. Now we count the documents, counting multiple occurrences of the same document only once. This value is then stored using the `hasNumberOfRelatedDocuments` relation of the seekda Ranking Ontology (see Section 2).

Now the related documents rank is calculated as follows (described in pseudo-code) in Listing 6:

```
// get the average number of related docs per service
average = totalNumberOfRelatedDocs / numberOfServices;
// get the root mean square deviation of the distribution of
// related docs per service
sumDeviationFromAverage = 0;
for (Service s : allServices) {
    sumDeviationFromAverage += s.numberofRelatedDocs - average;
}
variance = power(sumDeviationFromAverage, 2) /
    numberOfServices - 1;
rootMeanSquareDeviation = positiveSquareRoot(variance);
// get max outliers values
maxOutlier = average + (2.5 * rootMeanSquareDeviation);
```

```

// take into account the kind of relation from document to
// service. If a service has a number of related documents
// that is outside of the max outlier value we set the number
// to the average of related documents per service in order
// to not allow spam to influence the ranking value
for (Service s : allServices) {
    temporaryRank = 0;
    if (s.numberofRelatedDocs > maxOutlier) {
        s.numberofRelatedDocs = maxOutlier;}
    if (s.hasInlink) {
        temporaryRank = s.numberofRelatedDocs * 5;}
    if (s.hasTermVectorAssociatedDoc) {
        temporaryRank += s.numberofRelatedDocs * 4;}
    if (s.hasOutlink) {
        temporaryRank += s.numberofRelatedDocs * 2;}
    s.finalRank = temporaryRank / maxOutlier / 11;
}

```

**Listing 6.** Calculation of the Related Documents Rank

The single values that are used for the single kinds of related documents to calculate the temporary rank are currently experimental. These might be changed on a frequent basis until we discover the values that seem optimal for our needs. The final rank is stored for each service using the `hasRelatedDocsRank` relation of the `seekda Ranking Ontology`.

**WSDL Metrics Rank.** This rank is based on metrics that we extract from the WSDL descriptions. We currently take into account the documentation of (a) the service element, and (b) the operations. As we mentioned already in Section 2, approximately a fourth of all service descriptions contain some documentation on the service or operation level. The rank is calculated as follows in Listing 7:

```

for (Service s : allServices) {
    finalRank = 0;
    if (s.hasServiceDocumentation) {s.finalRank = 1;}
    if (s.hasOperationDocumentation) {s.finalRank += 3;}
    s.finalRank = finalRank/4;
}

```

**Listing 7.** Calculation of the WSDL Metrics Rank

We put more importance on the documentation of the single operations than of the service documentation, as we think that the operation might contain useful information regarding the functionality provided by the operation and regarding its invocation. We currently do not differentiate between whether all operations of a service are documented or only one or some. The final rank is stored for each service using the `hasWSDLMetricRank` relation of the `seekda Ranking Ontology`.

**Monitoring Rank.** This rank is based on the liveliness information of a service, e.g., is the server reachable, does it correctly implement the SOAP protocol, etc. This liveliness information is delivered by `seekda` on a weekly basis as shown in Listing 4. The availability score is a number between 0 and 1 that is set depending on the endpoint check result. The score is, e.g., 0 for read time-outs or errors and 1 if, based on the resulting payload (e.g., XML fault), we are rather

sure to be talking to a WSDL over SOAP. In-between different scores are set to express pages that are not found, pages that require a login or an authentication, etc., mostly based on the HTTP response code.

We get the average service availability score for different time periods: last week, last month and last 6 months. We assume that the long-time availability of a service is more relevant than only the short-time availability over one week. It is important to note that this rank does not state anything about whether the functionality that the service announces is correctly implemented or not. The rank is calculated as follows in Listing 8 and is stored for each service using the `hasMonitoringRank` relation of the `seekda` Ranking Ontology:

```
for (Service s : allServices) {
    finalRank = ((s.availabilityLastWeek * 1.5) +
                (s.availabilityLastMonth * 2.5) +
                (s.availabilityLast6Months * 6)) / 10;
}
```

**Listing 8.** Calculation of the Monitoring Rank

**Web API Rank.** For ranking Web APIs we currently only take into account the Web API confidence score. This score is calculated based on two classifiers within the crawler that check whether a Web resource might be a Web API or not. The rank is based on the crawl meta-data that is delivered by the crawler, as shown in Listing 1 and will be calculated based on the following information and assumptions:

- What is the Web API Confidence score of a document? This score is a final confidence score that is calculated from single scores provided by two Web API classifiers.
- Which crawler classifier has classified the document as Web API? As described in [3], one automatic classifier (SVM Classifier) has been trained on a set of data, while the other classifier (Web API Evaluator) performs structural and term vector analyses of the resources and assigns scores for specific indicators.

To calculate the rank we thus need to extract both the score and the component that has assigned the score. Based on first evaluations of the classifiers, we deem the score of the SVM classifier more important than the one of the Web API Evaluator. Listing 9 shows how the rank is calculated:

```
for (Service s : allRESTServices) {
    finalRank = 0;
    if (s.hasSVMClassifierAnnotation) {
        finalRank = s.hasWebAPIConfidenceScore * 3;}
    if (s.hasWebAPIEvaluatorAnnotation) {
        finalRank += s.hasWebAPIConfidenceScore * 1;}
    s.finalRank = finalRank/4;
}
```

**Listing 9.** Calculation of the Web API Rank



**Global Rank.** As already mentioned above, the calculation of the global rank differs depending on whether the ranked service is a WSDL-based service or a Web API. For WSDL services we calculate the global rank based on the Related Documents Rank, the WSDL Metrics Rank and the Monitoring Rank. The single ranks are numbers between 0 and 1, and from these we calculate the global rank as follows in Listing 10, putting equal relevance on the availability of documentation (related documents being estimated more important than the documentation within the WSDL) and on the liveliness of a service. This way services that are on the side reliable and on the other side (assumably) well documented are ranked best. The global rank is stored for each service using the `hasGlobalRank` relation of the `seekda Ranking Ontology`.

```
for (Service s : allServices) {
    s.globalRank = (s.hasRelatedDocsRank * 0.35) +
        (s.hasWSDLMetricRank * 0.15) +
        (s.hasMonitoringRank * 0.5);
}
```

**Listing 10.** Global Rank Calculation for WSDL-based services

For Web APIs, we currently only dispose of one kind of rank: the `WebAPI` rank, that is thus at the same time the global rank of the service.

## 4 Evaluation

The ranking approach that we present in this paper is especially valuable in service search in an open, large scale environment, i.e. search over a large number of services whose degree of documentation is unknown in advance and where no agreements with the service providers have been made beforehand concerning the service documentation. The rank does not take into account specific user requirements, as the features we include in our rank calculation are of such nature that we assume they are always relevant for users (e.g., we assume that it is always important for a user to know whether a service is available or not) and are thus adequate for a generic ranking. Although our final global rank does not take into account specific user requirements, this rank can be easily modified and composed in a different way by any RDF aware client as we return all single rank values as simple RDF triples.

As our ranking approach is still very new, especially concerning the inclusion of related Web resources for WSDL-based services and Web APIs in general, we cannot yet provide a fully-fledged evaluation for it. Parts of the ranking approach have though been evaluated in the scope of the `seekda` Web Service search engine: the WSDL-based services are ranked as presented in Section 3, basing upon the WSDL metrics and the monitoring information as described in the sections 2 and 2. The search engine ranking has been evaluated by experts in the Web Service domain; they have manually assessed the relevance of the discovered services with different ranking aspects (such as availability or documentation) taken into account and different combinations of the weight of these aspects being applied.

Providing a complete evaluation of our ranking approach is part of our future work. We plan to conduct an empirical evaluation, employing techniques as, e.g., user interviews and questionnaires to assess the importance and relevance of our ranking value(s). Also we would like to evaluate our approach with regard to other ranking approaches (such as provided in [2], [6] or [7]).

## 5 Conclusion and Future Work

In the previous sections we described our approach for multi-valued ranking of Web Services - both WSDL-based and Web APIs - based on ontology-based feature aggregation. We outlined what real-world data our ranking values rely upon and provided an overview on how the different rank values are calculated.

The approach we follow with our multi-valued ranking in general and the export of the ranking related data in a semantic format has major advantages (and novelties). These include (a) the fact that each RDF aware client can understand the rationale of a ranking, i.e. it can work with the final ranks, but can as well analyze the data on which the ranking is based. Also it can perform, if desired, its own ranking calculation with the service meta-data that is provided. Another innovative aspect is (b) the fact that our ranking approach covers and combines many aspects of a service, like its documentation, the existence of related information that is available on the Web, the liveliness data (which is a strong QoS criteria), and, in the case of Web APIs ranking, the classification score of services.

As future work we see first the evaluation of our ranking approach and second the further enhancement of it. We plan, e.g., to enlarge the number of criteria that we take into account for the ranking. We expect that especially for the Web APIs we will take into account more metrics; we will base our improvements on evaluations of the current approach. Also we can imagine to actively taking user feedback and requests into account, i.e. providing the user with a possibility to select what features are most important to him.

## Acknowledgements

The work is funded by the European Commission under the FP7 project SOA4All and by the FFG (Österreichische Forschungsförderungsgesellschaft mbH) under the FIT-IT project Service-Detective.

## References

1. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
2. Pajntar, B., Grobelsnik, M.: Searchpoint - a new paradigm of web search. In: 17th World Wide Web (WWW) Conference - Developers Track (2008)

3. Steinmetz, N., Lausen, H., Brunner, M.: Web service search on large scale. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) *ICSOC-ServiceWave 2009*. LNCS, vol. 5900, pp. 437–444. Springer, Heidelberg (2009)
4. Steinmetz, N., Lausen, H., Brunner, M., Martinez, I., Simov, A.: D5.1.3 - second crawling prototype (2009)
5. Toma, I., Roman, D., Fensel, D., Sapkota, B., Gomez, J.M.: A multi-criteria service ranking approach based on non-functional properties rules evaluation. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) *ICSOC 2007*. LNCS, vol. 4749, pp. 435–441. Springer, Heidelberg (2007)
6. Vu, L.-H., Hauswirth, M., Aberer, K.: QoS-based Service Selection and Ranking with Trust and Reputation Management. Technical report, EPFL (2005)
7. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering* 30(5), 311–327 (2004)

# Multi-level Monitoring and Analysis of Web-Scale Service Based Applications

Adrian Mos<sup>1</sup>, Carlos Pedrinaci<sup>2</sup>, Guillermo Alvaro Rey<sup>3</sup>, Jose Manuel Gomez<sup>3</sup>,  
Dong Liu<sup>2</sup>, Guillaume Vaudaux-Ruth<sup>1</sup>, and Samuel Quaireau<sup>1</sup>

<sup>1</sup> INRIA, 655 avenue de l'Europe, 38 334 Saint Ismier Cedex, France

<sup>2</sup> Knowledge Media Institute - The Open University Walton Hall, Milton Keynes, MK7 6AA,  
United Kingdom

<sup>3</sup> iSOCO, Pedro de Valdivia, 10, 28006 - Madrid

{Adrian.Mos, Guillaume.Vaudaux-Ruth, Samuel.Quaireau}@inria.fr,  
{C.Pedrinaci, D.Liu}@open.ac.uk,  
{Galvaro, Jmgomez}@isoco.com

**Abstract.** This paper presents a platform that aims at monitoring and analyzing large service-oriented applications executing on a very large scale. This is part of a vision of web-scale service utilization and management that is proposed by the SOA4All EU project. The paper shows how the platform obtains data from distributed runtimes and how it presents monitoring information at different levels of abstraction. They range from low-level infrastructure-related event details to high-level service and process analysis. Each level uses appropriate visualization techniques and widgets in order to convey the relevant information to the users in an efficient manner. The platform is under development and an advanced prototype is already available and described in the paper.

**Keywords:** monitoring, service-based applications, business processes, knowledge extraction, user interfaces for monitoring.

## 1 Introduction

SOA is largely still an enterprise specific solution exploited by and located within large corporations used mainly for integration. For instance, the current web only exposes around 28,000 traditional WS-based web services<sup>1</sup>. Nevertheless, as mobile devices and more efficient wireless communications facilitate ubiquitous computing, and as optical and broadband communication infrastructures expand, we expect the number of Web services to grow exponentially in the next few years.

The main objective of the EU project SOA4All is to provide a framework and an infrastructure that help to realize a world where billions of parties are exposing and consuming services via advanced Web technology. An important part in realizing this vision is the capability to provide users with a good understanding of how services and processes perform at functional and non-functional levels. This is achieved

---

<sup>1</sup> According to seekda.com the number of WSDL services available online on March 04, 2009 was 27.813.

through the SOA4All Analysis Platform that aims to provide the SOA4All users with information that would help them understand the performance characteristics and usage patterns of the services and processes they share. This platform is part of the SOA4All Studio that groups all the user-facing functionality of SOA4All.

There are important scalability requirements on the overall SOA4All infrastructure and in particular on the Analysis Platform. For example, the monitoring and provenance tools should cope with the exponential growth of the number of message interchanges and the size of log files. The monitoring and management infrastructure should be either able to handle growing amounts of work in a graceful manner or to be readily enlarged to cope with new workload on the fly (i.e. should be elastic).

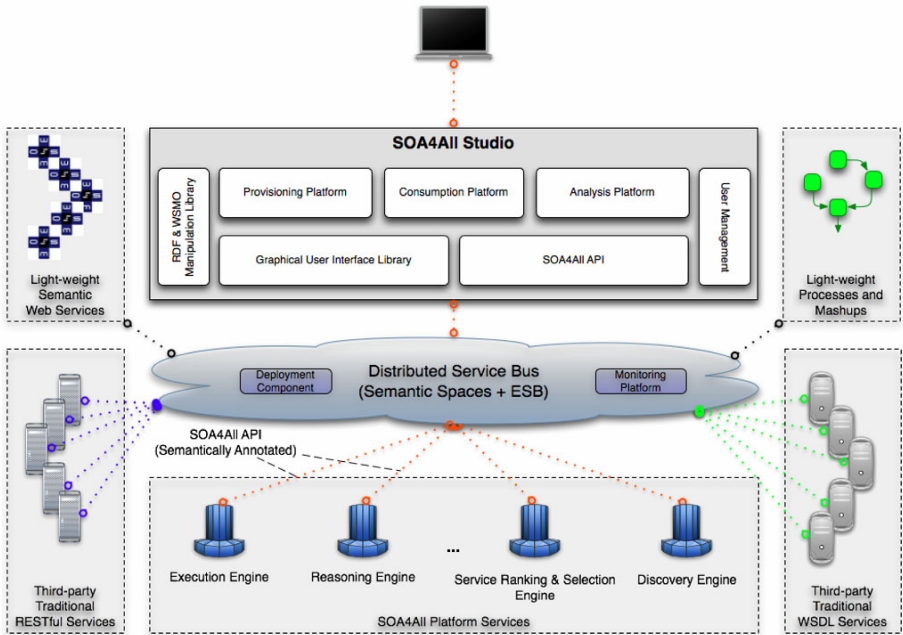


Fig. 1. Overall SOA4All Architecture

Fig. 1 provides a high-level view on the overall SOA4All architecture centered on the infrastructural components, the main artifacts manipulated and how the SOA4All Studio is integrated. Integration in SOA4All takes place through the so-called Distributed Service Bus (DSB) which integrates notions from Enterprise Service Buses and Semantic Spaces into a unified distributed infrastructure providing message-based and event-based communication, as well as a distributed RDF repository. From a client perspective, the DSB provides access to the platform services as well as the myriad of services provide by third parties. The services offered through the bus include, traditional third-party WSDL and RESTful services, light-weight Semantic Web Services based on annotations over traditional services, and infrastructural services supporting actions such that the discovery, ranking & selection and execution of services.

In a nutshell, the SOA4All Studio is the gateway for the user. It therefore provides a Web-based interface for creating or enhancing annotations, browsing them, discovering suitable services, invoking services, and finally analyzing their execution. These different functionalities are provided by three different platforms composing the Studio. In particular, the Service Provisioning Platform supports the user in providing annotations may them be WSMO-Lite, MicroWSMO, tags or ratings. The Service Consumption Platform allows the user to browse, discover and invoke existing services. Finally the Service Analysis Platform provides the means for users to analyze the execution of services either at runtime or post-execution.

The SOA4All Studio, like every other component is integrated into the overall architecture through the Distributed Service Bus. It is worth noting however that as opposed to platform services that provide infrastructural services, the Studio is mainly a client for these different components, allowing the user to interact with them in a seamless and transparent way. To support this, there is an internal component within the Studio in charge of supporting the interaction with the bus by making use of the appropriate messages and protocols based on Web and WS-\* standards.

The SOA4All Analysis Platform (AP) provides a wide range of widgets in its graphical views, organized according to their potential use and corresponding to the different levels of abstraction [1] that are useful for a variety of stakeholders. Furthermore, the AP offers a completely customizable approach to data visualization so as to correspond precisely to the expectations and needs that more advanced users have. These capabilities are presented in detail in the paper.

## 2 Related Work

Analysis of Provenance information in environments where high volumes of linked-data is combined and consumed is of particular interest in order to evaluate and assess the quality of the data [11].

Methods for representing and analyzing Provenance information have been widely addressed for a long time now [13]. In [14], the authors provide a taxonomy of provenance characteristics, differentiating between data-oriented approaches, focused on data items, and process-oriented approaches, where the emphasis is placed on the processes that generate and consume the data.

We also highlight the Open Provenance Model (OPM, [12]), a community-driven data model for Provenance designed to support interoperability of provenance technology. In OPM, three different pieces of provenance information are identified: artifacts, processes, and agents; while Provenance is represented by graphs, in which the nodes are those elements.

Another approach regarding data provenance is the one by Bunemann et al. [2] where they raise questions such as how the provenance information is obtained, how to cite components of a digital library such as a document in another context, or how to ensure the integrity of citations under the assumption that the cited databases evolve.

Finally, provenance models such as the one proposed by Harth et al. [12] include a “social dimension” which associates provenance information with the originator – which typically will be a person– of such a piece of information. In these models, the social context of the users can be combined with the analysis of provenance information to perform an extra assessment of the quality of the data.

Major efforts have been devoted so far to the monitoring and analysis of processes and services. In [13] the author provides a comprehensive analysis on current techniques in process monitoring and control within organisations. The author covers the topic by addressing four relevant perspectives to process controlling: the data perspective, the usage perspective, the tool perspective and the method perspective. The data perspective is concerned with collecting, storing and representing audit trail information. The author describes existing techniques and proposes an audit trail format that we took into account while devising our conceptualisations. Our approach, although similar in many respects, provides a more formal conceptualisation that is amenable to automated reasoning. This conceptualisation, which underlies the system, provides us with the capacity to apply advanced knowledge-based techniques in a domain-independent manner, as opposed to current practices.

The usage perspective concerns how process controlling management is approached. Some of the state of the art solutions focus on exception handling [14], whereas others focus on the global management of processes [15]. Our work so far has focussed mostly on gathering knowledge about processes in a way that can support automated reasoning as well as it can help business analysts in the management of processes. Although, automated process control is so far not supported by the tool, the very goal of our extensive conceptualisation work has precisely been carried out in order to better support machines in automatically controlling processes [16]. The work carried out so far represents substantial steps in this direction.

The tool perspective is concerned with the architecture and tools that have been developed for process monitoring and control so far. Among these the most relevant to us are for instance the work on PISA (Process Information System based on Access) which precedes the work presented in [13], and the work on the Business Process Intelligence tool suite [17-19]. While the former does not make any use of semantic technologies to enhance the monitoring capabilities the latter uses lightweight taxonomies. Their work is however more focussed on the integration of mining techniques and support for explanations. In this respect we believe both approaches are complementary and consider that a more extensive use of knowledge-based techniques could indeed enhance their results. In a similar way, future work on SENTINEL will indeed be inspired by this research.

On a broader sense, BAM functionality as part of the BPM system is already supported in several products, e.g. [20,21]. Typically the BAM solution is tightly integrated with the BPEL engine which is part of the BPM system. Metrics are defined and calculated based on the events which are published by that BPEL engine. The event publishing mechanism is thereby proprietary. Another approach is to extend the BPEL process with event publishing activities, which invoke operations on a monitoring tool. This approach is utilized in [22] and [23]. The benefit of this approach is that event publishing does not depend on a proprietary mechanism of a BPEL engine. The main disadvantage however is that the BPEL process is more difficult to read and maintain, as it contains new activities, which deal with technical issues, and not just business logic. Our work relies on event publishing mechanism by the execution engines (e.g., BPEL engine). The difference to the existing tools is that in our approach ontologies are used for the description of events and the data they contain. In this way we better support the integration of proprietary formats via ontological mappings, and still allow inferring implicit knowledge at runtime.

### 3 Overall Architecture

This section gives a high-level overview of the Analysis Platform architecture. It is illustrated in Fig. 2., which presents the main components of the platform and their connections to the external components. The elements within the light grey rectangle correspond to the Analysis Platform components and their Studio counterparts, while the elements drawn outside the rectangle represent external components (the Analysis Warehouse and the DSB). The thick arrows illustrate interactions between the Analysis Platform and the external components. The figure also illustrates that the Analysis Platform provides RESTful services that wrap parts of its functionality for easy external use.

The main source of data for the Analysis Platform is the *SOA4All Distributed Service Bus* (DSB), which is the backbone of the SOA4All runtime infrastructure. Several data collectors (bus, grid and engine collectors) available through the bus will feed relevant monitoring data to the AP, which in turn will be able to control and filter data collection through appropriate management operations. In addition, the AP will be able to send management commands to the DSB to instruct its components to perform a variety of infrastructural operations related to the customization or control of DSB components.

We distinguish between two types of monitoring events that can be received from the DSB: *infrastructure events* and *application events*. The former correspond to low-level details of the execution platform while the latter correspond to data about the actual execution of the application services and processes.

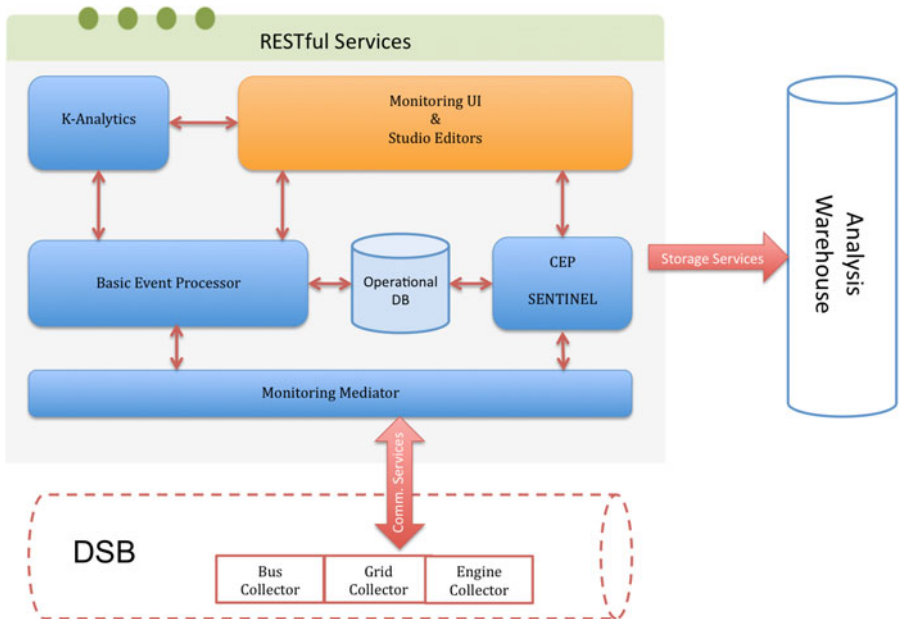


Fig. 2. Overview of the Analysis Platform Architecture



The following are the components of the Analysis Platform.

- **Monitoring Mediator (MM):** Obtains Data from the DSB and the execution engines using the Studio APIs. It is both a listener and a proactive entity: its main role is to interface with the infrastructure (it receives events and can also filter and control the event sources); it can also communicate to the management/monitoring APIs to coordinate event production.
- **Basic Event Processor (BEP):** performs pre-processing of monitoring events from the MM, including computing basic averages and statistics; provides data to Knowledge Analytics and SENTINEL for knowledge extraction; feeds data to basic UI widgets; uses the analysis warehouse to store derived information and basic computation results; the BEP acts as a single point of entry to the analysis warehouse thus interfacing with the other internal components of the Analysis Platform as well as with the Studio editors requesting analysis information to display on diagrams.
- **Knowledge Analytics (K-Analytics) component:** this component will be able to interpret the environment provenance information in order to produce domain-oriented interpretations of process executions to increase user understanding of such executions, which will most likely be potentially very large and complex. Since K-Analytics performs post-mortem analysis, it will query the BEP, and other data sources, for monitoring information when necessary.
- **UI Widgets:** basic graphical representation (as seen in mockups, such as average response time for a service, availability etc.);
- **SENTINEL:** this environment provides knowledge-based techniques in order to detect and diagnose process deviations based on monitoring information and informed by context data.

## 4 Mediation and Basic Event Processing

This section briefly presents the mechanisms used by the Analysis Platform to integrate with the Studio editors and the runtime part of SOA4All as well as the event processing functionalities that are going to be offered by the BEP module.

The **Monitoring Mediator (MM)** is the interface of the Analysis module to the runtime platforms. It connects to the aggregated runtime data collectors and transforms the events received into API calls to the Basic Event Processor. In addition, the Monitoring Mediator is used to relay management operations originating in the Analysis Platform through to the runtime environments.

The Monitoring Mediator leverages mechanisms complying with the WSDM standard to communicate to the data collectors.

The **Basic Event Processor (BEP)** is responsible for

- Parsing the monitoring events received from the MM and to perform data processing in order to extract derived information (such as computing averages and basic aggregated events)

- Communicating raw and derived to upper-level processing entities, the K-Analytics and SENTINEL engines.
- Updating the graphical data structures used by the UI widgets.
- Storing derived events into the analysis warehouse using the Studio Storage Services. These derived events can originate in the BEP itself or indeed can arrive from K-Analytics or SENTINEL.

## 5 SENTINEL

Semantic Business Process Analysis, that is, the extension of Business Process Analysis with Semantic Web and Semantic Web Services technologies has been proposed as a means for increasing the automation of tasks through the provisioning of semantic descriptions of the artefacts involved in the life-cycle of business processes [16]. Research in this direction has already produced an extensive set of ontologies and tools within an overall framework that spans from low-level technical details to high-level methods and tools for analyzing business processes and services to achieve the desired business goals [6, 7, 16, 24]. The tool presented in this paper is part of this overall research thread and as such is strongly based on the use of semantic technologies to enhance the state of the art in Business Process Analysis.

SENTINEL [10] makes use of semantic technologies to facilitate the monitoring of processes and activities. In a nutshell, it uses ontological models for keeping track of execution of processes, and it uses the overall information gathered over time in order to derive business-level knowledge from the low-level trails. In particular, it uses the COBRA and EVO ontologies.

COBRA[9] provides a core terminology for supporting Business Process Analysis (BPA) where analysts can map knowledge about some particular domain of interest in order to carry out their analyses. It is worth noting that COBRA does not aim to provide a fully-comprehensive conceptualisation for supporting each and every kind of analysis since the scope would simply be too big to be tackled appropriately in one ontology. Instead COBRA, depicted in [9] provides a pluggable framework based on the core conceptualisations required for supporting BPA and defines the appropriate hooks for further extensions in order to cope with the wide-range of aspects involved in analysing business processes.

COBRA builds upon Time Ontology that provides a temporal reference by means of which one can determine temporal relations between elements. COBRA provides a lightweight foundational basis in order to ensure a coherent view among additional extensions. It defines concepts such as Process, Activity, Process Instance, Role, Resource, Organisation or Person which are to be refined within specific ontologies as defined within SUPER, or other approaches like the

Enterprise Ontology [7] or TOVE [8]. COBRA has been extended with a reference Events Ontology (EVO) [9] that provides a set of definitions suitable to capture monitoring logs from a large variety of systems and ready to be integrated within our core ontology for analysing business processes.

In order to carry out these activities, SENTINEL makes use of an event processing engine, namely Drools Fusion<sup>2</sup>. Drools provides support for Complex Event Processing over the stream of events generated by the monitoring infrastructure during the execution of processes and services. Based on this stream, the engine is able to identify meaningful events within the cloud, correlate them, reason about the temporal relationships, etc.

The current version of the tool for instance contains rules that process the events collected through the BEP in order to:

- Update the information about lifecycle of business processes and activities therefore tracking all the states through which they have been (e.g., suspended, executing, etc) and the time spent on each of these states;
- Track the involvement of agents (i.e., humans and computers) in processes and activities;
- Compute metrics based on user-defined formulae and whenever certain criteria are met, e.g. compute the execution time of business processes when they finish;
- Compare the results obtained with thresholds determining the limit between “normal” situations and extreme ones;
- Support near real-time visualization of computations performed.

Therefore, in a nutshell, the tool performs continuous data warehousing whereby information coming from the monitoring infrastructure is collected, correlated, processed and stored in a convenient high-level format that better supports its subsequent analysis. On the basis of this higher-level model, additional tools perform near real-time computations when certain criteria are met (e.g., process finished), and periodical computations (e.g., daily batch processing) for less time critical activities. The results are all stored within an integrated analysis warehouse which supports querying and reasoning and therefore it allows the seamless retrieval, interpretation, correlation and combination of all analysis results in order to better support the analysis of processes and services execution by both humans and machines.

The current version of the tool performs the majority of these analyses in a somewhat disconnected way from other results. For instance, thresholds are currently defined manually by users. In the future, however, the idea is to benefit from the integrated analysis warehouse that is populated during execution and analysis, may it be post-execution or real-time, in order to benefit from previously calculated results (e.g., average execution time, deviation of the execution time) to directly establish a reasonable threshold. Similarly, more advanced techniques such for the diagnosis of processes, such as Heuristic Classification [16], have been tested but require additional work in order to find an appropriate trade-off between performance, accuracy, sensibility and the level of prediction.

## 6 K-Analytics: Knowledge Analytics Component

The Knowledge Analytics component (K-Analytics) derives information from several sources in order to deal with it more suitably at an upper-level which end-users can

---

<sup>2</sup> <http://jboss.org/drools/drools-fusion.html>

understand more easily. Hence, the motivation of this component comes from the need to abstract from the many sources of data which include a huge number of logs to a conceptual level that aggregates that information into a more condensed and specific version, in order to facilitate end-users the understanding of all this information.

In SOA4All, we do not rely on the BEP mentioned in the previous section as the only source from which we will get raw information, but we will be making use of further data which is semantically linked, in the form of feedback information and logs which do not come from the execution of the services itself, but from the interactions of the users within the platform, such as the services they open or use.

Therefore, even though the component is open enough to cater for the needs of different situations where the data we are interested in comes from other sources, we will focus in this particular case on the following three ones:

- The **BEP**, which will be queried for the necessary information about executions of the services.
- The **Auditing Framework**, which takes care in SOA4All of semantically logging the actions of the users within the platform (e.g., opening a service, invoking a service), which are stored in the Semantic Spaces via the Storage Services of the SOA4All Studio.
- The **Feedback Framework**, which deals with information of ratings, tags and comments made by the users, also stored in RDFs in the Semantic Spaces via the Storage Services, following the Review Schema<sup>3</sup> and the Tag Ontology<sup>4</sup>.

The aforementioned three sources will obviously yield such a quantity of logs from the project that it will be unfeasible to analyze them at run-time retrieving each piece of information we are interested in. On the contrary, this component performs a post-mortem analysis of those sources in order to make the access to the relevant information easier. Therefore, the purpose of our component is to extract and compute the relevant information and store it again in a more suitable and condensed format at a conceptual level, in such a way that we are able to easily access the results of the computations we are interested in, when desired. The knowledge-level information we aim to infer is supported by an ontological layer, mainly based on the following concepts, which, in turn, come from several useful subconcepts, also addressed:

- **Frequency:** How much is a service used. We derive this concept from Visibility frequency (how many times a service is opened) and Invocation frequency (how many times it is actually consumed)
- **Performance:** How well does the service behave. We derive it from Time performance (how long does the execution take) and Reliability performance (if the execution finishes well)
- **User Perception:** What do the users think about the service. Derived from Ratings perception (average of the ratings) and Reviews perception (number of ratings, comments and tags)

The BEP source is obviously important in order to extract the information about invocation frequency and performance, while the Auditing Framework is the source for

---

<sup>3</sup> <http://purl.org/stuff/rev#>

<sup>4</sup> <http://www.holygoat.co.uk/owl/redwood/0.1/tags/>

the visibility frequency, and the Feedback Framework is the source for the derivation of the User Perception. In turn, the results of the analysis are supported by a generic ontology, based on the aforementioned concepts, which may be combined with suitable domain ontologies for establishing better explanations of the characteristics of each service. Regarding the way K-Analytics operates, in order to extract the upper-level information that we are interested in, our component permits selecting the services we are interested in “tracking”. Hence, when a *track* has been set on a service, our component begins to perform its calculations at batch-time nightly, condensing the great amount of logs that come from different sources into a more appropriate format that the GUI will be able to access more efficiently.

K-Analytics places a special emphasis on the analysis of provenance information, which is a topic of particular interest in a Web of Data where so much interlinked information is used [11], as it actually happens in SOA4All, where a huge number of services are foreseen to be invoked. In particular, we are interested in the so-called *why-provenance* (the origins that were involved in calculating a single entry of a query result), which Buneman et al. [2] distinguished from *where-provenance* (the exact location from where an element of a query has been extracted from), and which is also different from *how-provenance* (the way the origins were involved in the calculation) addressed by Green et al. [3]. Hence, K-Analytics will permit to go from the computed concepts to the actual logs that have generated them, passing through the intermediate sub-concepts that are used for these calculations, and in this way be able to understand the characteristics of the results more deeply when desired.

### 7 Scalability Aspects

This section briefly discusses scalability aspects that are being addressed in the presented Analysis Platform. The discussion is based on a simple example illustrated in Fig 3.

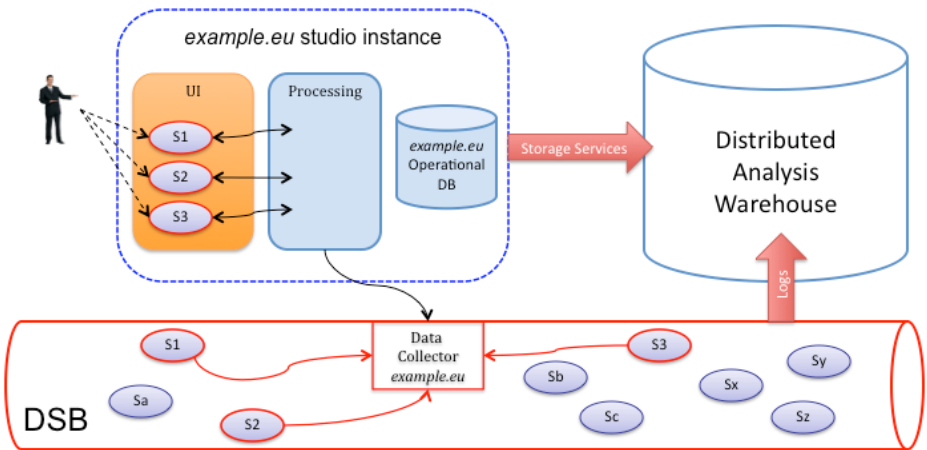


Fig. 3. Illustration of Scalability Aspects

As suggested in the Introduction, in the scenarios envisaged by SOA4All, the proposed Analysis Platform, together with other Studio components, will potentially be deployed on many nodes, each administered by an organisation that wants to provide SOA4All functionality to its users. However, the invisible SOA4All infrastructure based on the distributed service bus can be seen as transcending individual providers and offering a pervasive infrastructure base that is accessible from any SOA4All-enabled node. This is also true for the RDF storage support available as a pervasive service for SOA4All nodes. This pervasive aspect is realised through different federation techniques that are out of the scope of this paper.

This architecture however has important implications with regard to scalability of the Analysis Platform. In order to cope with the extremely large numbers of users envisaged by SOA4All as a whole, as well as potentially massive amounts of data, we leverage the natural distribution provided by the infrastructure and optimize data processing and storage to achieve a good compromise in terms of performance and scalability. Fig 3 shows a sample instance of the Studio called *example.eu*. This Studio instance runs its own individual Analysis Platform instance, separated from other instances available with other providers. Users connecting to *example.eu*'s Studio will access this version of the Analysis Platform and will be interested, as a group, in a relatively limited number of services and processes (that are relevant to *example.eu*'s business). This implies that all the processing that this instance of the Analysis Platform will perform will relate to these services (illustrated in the image as S1, S2 and S3). Its internal operational database will contain detailed analysis data for these items. This prevents this instance from processing data for services that are not of interest to its users (in effect, only services that at least one user of the domain is interested in, will be analyzed).

Naturally, *example.eu*, as all the other SOA4All-enabled nodes running the Studio, will connect to the DSB and will leverage the highly-distributed RDF storage. This enables the collection of data for any service or process executing anywhere in the world, when such data is required. We envisage that even for services/processes that no user has expressed interest in yet (in any domain instance), the DSB will collect basic data such as moving averages for execution times and availability information, and store it in the RDF storage, in order to have minimum bootstrap information ready when users become interested in the particular service/process. As soon as they do become interested, data collection becomes much more significant as it is driven by the Analysis Platform instance, and detailed analysis data can be stored in the individual operational databases. In short, detailed analysis for selected entities is performed "locally" in the same domain as the user, and basic analysis and long-term storage is performed on the distributed infrastructure for all entities.

## 8 Information Presentation

The widgets, consoles and screens used by the Analysis Platform are accessible in a unified and integrated manner through the SOA4All Studio.

Monitoring information is often structured around three different views [4]: (i) the Process View which is concerned with key performance indicators of processes and services; (ii) the Resource View centered around the resources, human or mechanized,

required for executing processes; and (iii) the Object View which focuses on business objects such as inquiries, orders or claims. These three views are populated with statistical information such as the minimum, the maximum, the average or the deviation of some parameter of interest.

These views are of major relevance to analysis and management, and as a consequence they are typically supported by monitoring tools such as Business Activity Monitoring solutions. However, different users have different roles, interests, access rights, and preferences and these vary depending on the specific scenario, the focus of their analysis, etc. The user interface of a fully-fledged general purpose solution must therefore be characterized by its flexibility [4]. This includes for instance support querying, filtering and ordering the information according to user defined specifications [5]. Indeed, given the kinds of users addressed, the specification of these queries and filters should be supported in a simple way so that humans can browse the existing execution information effectively. Similarly, different domains exhibit particular characteristics, which impede a “one size fits all” approach. The monitoring tool should therefore support users in defining their own visualization templates to be populated with relevant monitoring information. The visualization framework should be supported by a wide range of graphical representations such bar charts, line charts, pie charts, time series charts, etc. Additionally, the visualisation framework should support the presentation of user-defined information combining diverse statistical information about processes, etc.

The use of knowledge-based technologies will play a major role in bringing flexibility to the Analysis Platform to be able to adapt to a myriad of users and services in a seamless way. On the one hand the use of a formal conceptual model closer to human understanding than low-level syntactic representations will bring the body of knowledge to a higher level of abstraction more suitable for human interpretation. On the basis of this conceptual model we shall support humans in defining queries or navigating through the data by simply following the conceptual schema and generating the appropriate ontological queries transparently. This will allow, among other things, to seamlessly navigate across different abstraction layers. Additionally, we envisage the use high-level strategic models such as the one presented in [6] in order to guide the presentation of analysis information driven by the importance and impact data can have on underlying services and their related interdependencies.

## 9 Conclusion and Future Work

Large-scale web-based service platforms require the support of comprehensive monitoring and analysis tools if they are to become widely used. This paper presented our approach for an Analysis Platform that we believe is suitable in such contexts. A detailed description of the current prototype, which is in an advanced stage, can be found in [28].

There are however important remaining developments for the AP. Different editors that will likely be used by end-users to create processes and services need to use information generated by the AP to augment their graphical displays. The APIs that are currently available may need to be refined to correspond to this need.

The AP needs to better leverage semantic information about services and processes and update semantic repositories with more analysis and monitoring information. This

information could in future be used by ranking and selection mechanisms that aid users in finding the best services for their needs.

One of the main remaining challenges however refers to the scalability ambitions of the envisaged context, which may involve millions of services used by numerous users. The current prototype already supports a distributed environment and it will need to be tested and potentially refactored to correspond to the wide distribution of its data sources both from functional as well as non-functional points of view (i.e. dealing with large amounts of data in information presentation as well as ensuring that the AP infrastructure can cope with the demands). We are confident that the decoupled, distributed approach we have taken will enable us to reach this goal and we look forward to the validation tests that will be carried out through SOA4All.

## Acknowledgements

The authors wish to gratefully acknowledge the support for this work provided by the European Commission through the SOA4All project.

## References

1. Mos, A., Boulze, A., Quaireau, S., Meynier, C.: Multi-layer perspectives and spaces in SOA. In: Proceedings of the 2nd international Workshop on Systems Development in SOA Environments (SDSOA 2008), Leipzig, Germany (May 2008)
2. Buneman, P., Khanna, S., Tan, W.C.: Why and Where: A Characterization of Data Provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, p. 316. Springer, Heidelberg (2000)
3. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance Semirings. In: Proceedings of the 26th Symposium on Principles of Database Systems (PODS). ACM, New York (June 2007)
4. Muhlen, M.z.: Workflow-based Process Controlling. In: Foundation, Design, and Implementation of Workflow-driven Process Information Systems, Logos, Berlin, vol. 6 (2004)
5. Hur, W., Bae, H., Kang, S.-H.: Customizable Workflow Monitoring. *Concurrent Engineering* 11(4), 313–325 (2003)
6. Pedrinaci, C., Markovic, I., Hasibether, F., Domingue, J.: Strategy-driven business process analysis. In: 12th Conference on Business Information Systems, BIS (2009)
7. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *Knowledge Engineering Review* 13(1), 31–89 (1998)
8. Fox, M.S.: The TOVE Project Towards a Common-Sense Model of the Enterprise. In: Belli, F., Radermacher, F.J. (eds.) IEA/AIE 1992. LNCS, vol. 604, pp. 25–34. Springer, Heidelberg (1992)
9. Pedrinaci, C., Domingue, J., Alves de Medeiros, A.K.: A Core Ontology for Business Process Analysis. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 49–64. Springer, Heidelberg (2008)
10. Pedrinaci, C., Lambert, D., Wetzstein, B., van Lessen, T., Cekov, L., Dimitrov, M.: SENTINEL: A Semantic Business Process Monitoring Tool. In: Workshop: Ontology-supported Business Intelligence (OBI 2008) at 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany (2008)
11. Hartig, O.: Provenance Information in the Web of Data. In: Proc. of the Linked Data on the Web Workshop at WWW (2009)



12. Moreau, L., Plale, B., Miles, S., Goble, C., Missier, P., Barga, R., Simmhan, Y., Futrelle, J., McGrath, R., Myers, J., Paulson, P., Bowers, S., Ludaescher, B., Kwasnikowska, N., Van den Bussche, J., Ellkvist, T., Freire, J., Groth, P.: The Open Provenance Model. Technical report, Electronics and Computer Science. University of Southampton (2008)
13. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys* 37(1), 1–28 (2005)
14. Simmhan, Y., Plale, B., Gannon, D.: A Survey of Data Provenance in e-Science. *SIGMOD Record* 34(3), 31–36 (2005)
15. Harth, A., Polleres, A., Decker, S.: Towards a Social Provenance Model for the Web. In: *Proceedings of the Workshop on Principles of Provenance* (November 2007)
16. Mühlen, M.: Workflow-based Process Controlling. In: *Foundation, Design, and Implementation of Workflow-driven Process Information Systems. Advances in Information Systems and Management Science*, vol. 6. Logos, Berlin (2004)
17. Casati, F., Castano, S., Fugini, M., Mirbel, I., Pernici, B.: Using patterns to design rules in workflows. *IEEE Trans. Softw. Eng.* 26(8), 760–785 (2000)
18. Kueng, P., Kueng, P., Meier, A., Meier, A., Wettstein, T.: Computer-based Performance Measurement in SME's: Is there any option. In: *Proceedings of the International Conference on Systems Thinking in Management*, pp. 8–10 (2000)
19. Alves de Medeiros, A.K., Pedrinaci, C., van der Aalst, W., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In: *Proceedings of International IFIP Workshop On Semantic Web & Web Semantics, SWWS 2007* (2007)
20. Sayal, M., Casati, F., Dayal, U., Shan, M.-C.: Business process cockpit. In: *VLDB 2002: Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment*, pp. 880–883 (2002)
21. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.-C.: Business Process Intelligence. *Computers in Industry* 53(3), 321–343 (2004)
22. Castellanos, M., Casati, F., Dayal, U., Shan, M.-C.: A comprehensive and automated approach to intelligent business processes execution analysis. *Distributed and Parallel Databases* 16(3), 239–273 (2004)
23. Wahli, U., Avula, V., Macleod, H., Saeed, M., Vinther, A.: *Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products*. Number SG24714801. IBM RedBooks (2007)
24. Oracle Corporation. Oracle Business Activity Monitoring (BAM) (2008), <http://www.oracle.com/appserver/business-activity-monitoring.html>
25. Baresi, L., Ghezzi, C., Guinea, S.: Smart monitors for composed services. In: *ICSOC 2004: Proceedings of the 2nd international conference on Service oriented computing*, pp. 193–202. ACM Press, New York (2004)
26. Roth, H., Schiefer, J., Schatten, A.: Probing and Monitoring of WSBPEL Processes with Web Services. In: *CEC-EEE 2006: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, Washington, DC, USA, p. 30. IEEE Computer Society, Los Alamitos (2006)
27. Alves de Medeiros, A.K., Van der Aalst, W., Pedrinaci, C.: Semantic Process Mining Tools: Core Building Blocks. In: *16th European Conference on Information Systems*, Galway, Ireland (June 2008)
28. Mos, A., Pedrinaci, C., Rey, G.A., Martinez, I., Hamerling, C., Vaudaux-Ruth, G., Liu, D., Quaireau, S.: D2.3.2 Service Monitoring and Management Tool Suite First Prototype, SOA4All Deliverables, <http://www.soa4all.eu/file-upload.html?func=startdown&id=129>

# Calculating Service Fitness in Service Networks\*

Martin Treiber<sup>1</sup>, Vasilios Andrikopoulos<sup>2</sup>, and Schahram Dustdar<sup>1</sup>

<sup>1</sup> Distributed Systems Group, Vienna University of Technology

<sup>2</sup> ERISS, Dept. of Information Systems and Management, Tilburg University

treiber@infosys.tuwien.ac.at, v.andrikopoulos@uvt.nl,

dustdar@infosys.tuwien.ac.at

**Abstract.** Inspired by the biological perspective of service ecosystems, we propose to define the fitness of services in service networks. In our work, we show how to calculate the service fitness from the provider perspective using locally available information as a reflection of the position of the service in the service network. For that purpose we define a fitness corridor with upper and lower bounds that confine the service fitness area. After establishing a fitness corridor, we show how to calibrate the fitness calculation parameters to better reflect the service market and how to use the calculated fitness trends for making decisions about the provisioning of a service.

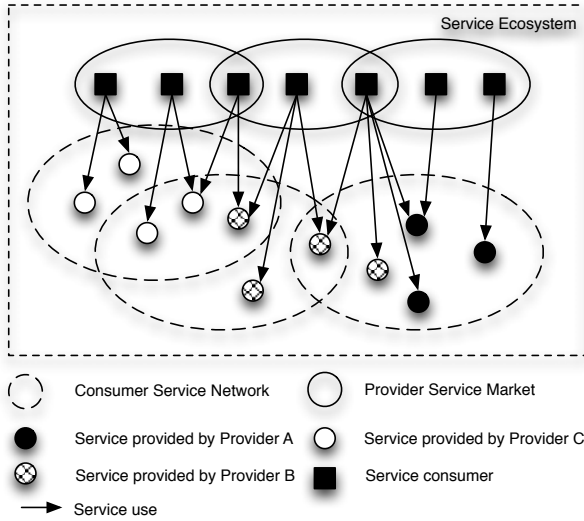
## 1 Introduction

The proliferation of the SOA paradigm into the highly competitive and volatile world of business has naturally led to the necessity for quickly evolving services that aim to fulfill the changing requirements set by the market. The evolution of services [1,2] and the evolution of the context within which the services operate - the *service ecosystem* [3] - create a number of challenges that need to be addressed.

In this context, we consider a Web service ecosystem a set of potentially overlapping service networks [4] of service consumers. Each service network represents a pool of services that is available to a service consumer. A service consumer selects services from this pool for later use in service compositions or for single service invocations (see Figure 1). From the perspective of the service provider, service networks have fundamentally different characteristics. To illustrate the perspective of the service provider, consider the following example: a company which is providing software services is interested that their offerings reach as many customers as possible. For that purpose the company has a number of disparate services in its portfolio targeted at different customers with different characteristics and needs and may therefore participate in more than one service networks. In that sense, the services offered by a service provider may be part of different service networks.

---

\* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).



**Fig. 1.** Service Ecosystem

Consequently, we will refer to the provider’s *service market* (Figure 1) as a particular subset of the service ecosystem in which the provider offers his services to a set of consumers. More specifically, a provider’s service market can be regarded as a segment of a service network where other (competing) service providers offer similar or equivalent services to a specific consumer set. In contrast to the service consumer, a service provider may have limited knowledge about the service network and can only estimate the size of service markets [5].

As a solution to this problem, we propose the use of a utility function that uses available information at the provider side to calculate a *fitness corridor* for services with regard to service networks and service markets respectively. The fitness corridor defines an upper bound for the maximum service network share that can be achieved with a service with the given information. The calculation of the lower bound requires stochastic or marketing methods that are capable of dealing with uncertain information. The observation of a given service fitness corridor over time serves as input for provisioning decisions for the service provider. For example, (negative) fitness development of a service might lead to the decision to decommission the service. Apart from provisioning, we envision the use of service fitness for prediction of performance trends of the service in terms of market share.

The rest of the paper is organized as follows. We introduce our approach in section 2 and illustrate our assumptions concerning the behavior and structure of service networks. In section 3 we define service fitness and provide the means to calculate it based on the assumptions of the previous section and show its usefulness with a concrete example from an existing company. We conclude the paper with related work in Section 4 and an outlook in Section 5.

## 2 Service Networks Characteristics

Service networks emerge around businesses and exhibit by their nature a dynamic behavior. Service providers and consumers may enter and leave the service network as the service network expands and shrinks over time, depending on external factors like consumer demand, profit margins, etc. In addition, a particular service provider that participates in the network does not have in principle up-to-date global knowledge of the service network without the presence of a centralized authority for this purpose. As we will discuss in the following sections in more detail, these two factors (constant change and partial knowledge) limit the calculation of fitness to only locally available information.

### 2.1 Roles and Structures in Service Networks

Traditional Service-Oriented Architectures [6] define three distinct roles, (i) the service provider, (ii) the service consumer and (iii) the service broker (registry). We follow this characterization but focus on the relation between service providers and service consumers. We can distinguish between (i) service providers that offer services to (ii) a set of consumers (or alternatively, *customers* of the service) which are in turn a subset of (iii) the potential customers (see Figure 2).

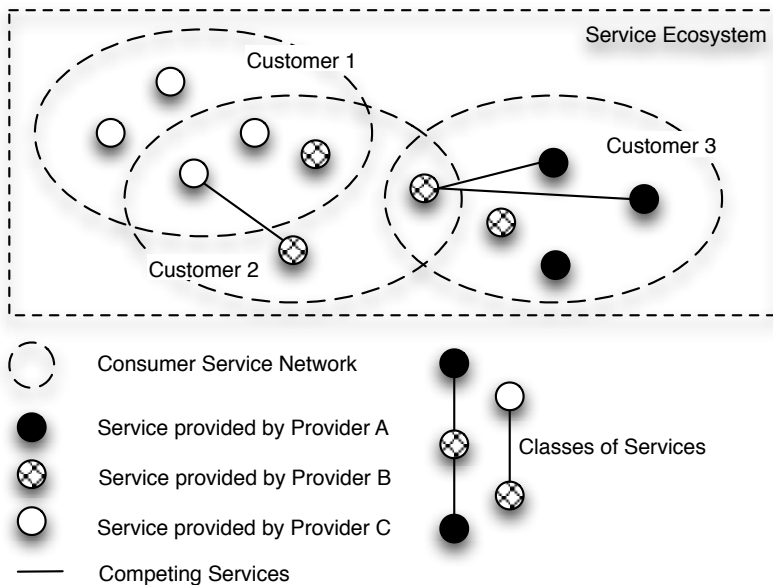


Fig. 2. Emerging classes in service networks.eps

## 2.2 Knowledge in Service Networks

Crucial for the calculation of service fitness is the available information. In our work, we follow a semi-open approach to define service networks, i.e., we assume that there exists a finite number of service network members which may enter and leave the service network at any time. Furthermore, we assume that there is limited information available for each service network member. Lessons learned from the failure of public service registries [7] impose that the existence of a central entity like a public registry is not very likely which leads to the situation as where very service network member has an *observable universe* in which information about the service network can be obtained. During the life time of a service network we can observe various changes to it: the introduction of new services or service versions, the decommissioning of services, or the entering of new service providers may all result in dramatic changes of service networks. We observe these dynamics in our proposed approach by events that occur in service networks and lead to changes of the service network structure. In particular, we consider the following events that are of interest for service networks:

**NS** New service - A new service enters the network.

**NV** New service version - An existing service is modified and a new version of the same service is published. In these cases, the original version can also be available in parallel.

**DS** Decommission of service - A service is removed from the service network.

**DV** Decommission of service version - A (older) service version is removed from the service network.

**NP** New service provider - A new service provider enters the service network and offers new services in the service network.

**LP** Leaving of service provider - A service provider leaves the network and removes its services from the service network.

The evolution of a service network can therefore be seen as a series of modifications to it, in a similar fashion as the evolution of a service itself can be perceived as a series of unambiguous changes to it [1]. It is important to notice that we explicitly consider *time* of central importance in our approach. We measure time in discrete units that are defined by the service provider, enabling each service provider to monitor his services privately without the need for further synchronization with other service providers.

## 3 Service Fitness in Service Networks

In this section, we introduce the notion of *service fitness* as a measure of the success of a service provider in a service network. It is important to notice that the notion of fitness of services depends highly on the context. For instance, an ordering service might be fit in the context of computer part supplier networks, but not in customer service networks since the service might be tailored for the requirements of the part supplier service network.

Changes to the context of the services are reflected on the market share of the service and we can therefore observe them as changes to the service fitness. These changes take place within a certain boundary which we refer as *fitness corridor*. A fitness corridor is defined by an upper bound that denotes the best possible fitness (as calculated using the available data) and a lower bound which is calculated with stochastic methods like the Monte Carlo method [8] or with marketing research methods [5]. We propose to use a utility function that takes local information into account and lets service providers calculate the fitness of their services independently from other service providers. In order to calculate the fitness for a given service, we use Equation 1 which gives the basic definition of service fitness:

$$f^m(\tau) = \text{Actual Use} / \text{Potential Use} \quad (1)$$

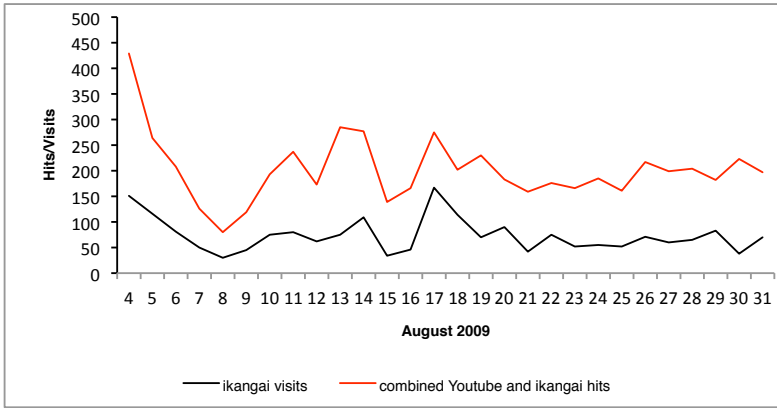
In the formula,  $\tau$  denotes the *time window* i.e. the time interval for which fitness is calculated,  $m$  defines the observation window i.e. how many time windows are included in the calculation of the fitness, the *Potential Use* is the estimated number of possible service customers and *Actual Use* is the number of actual service customers.  $f^2(1h)$  for example denotes that the formula will be calculated for  $m = 2$  windows of  $\tau = 1$  hour each. We always normalize the output of the fitness function to a value of the interval  $[0 \dots 1]$ . Depending on the amount of information that we take into account while estimating *Potential Use* we can come with a theoretical upper and lower bound for fitness at any given time interval.

More specifically, we establish an upper bound for the fitness of a given service by collecting information about the status of the service and the perceived status of the network. We use (exclusively) data from the event logs of each service (version) for a given time period. In particular, we are interested in two types of events, (i) the request for the service description (e.g. the WSDL file itself) and (ii) the invocations of the service by different consumers. Using existing tools like Webalizer [1] we can access this information easily. Figure 3 shows the result of the analysis for the services of ikangai for August 2009. By using these available data, a service provider can aggregate the information from the different services and their versions in his portfolio. In order to establish the lower bound of a fitness corridor, we rely on estimation techniques which can be based on market research. In the case of ikangai solutions for example we derived their market size estimation by simply estimating the hits of their homepage and by investigating relevant forums and social networking outlets for interested users. ikangai solutions estimated an average of 500 hits per day during August, thus being well over the actual average number of 201 hits per day.

### 3.1 Calculating the Fitness of Services

We now show how to apply the basic fitness formula to calculate the fitness of services in service networks. For the upper bound of the fitness function we replace equation 1 with equation 2 that incorporates event log information:

<sup>1</sup> <http://www.mrunix.net/webalizer/>



**Fig. 3.** ikangai web page statistics. The data represents the cumulated youtube hits and ikangai web page visits.

$$f^m(\tau) = \sum \text{Inv}(i) / (\sum_m \text{Req}(i) + \sum_m q(i)) \tag{2}$$

$\text{Inv}(i)$  is a function that returns 1 if a customer  $i$  invokes the service at least once during the given time windows  $m$  and 0 otherwise. Similarly,  $\text{Req}(i)$  is a function that returns 1 if customer  $i$  requests the service specification (at least once) during  $m$  time windows and returns 0 otherwise.  $q(i)$  returns 1 if customer  $i$  invokes the service at least once in the  $m$  time windows without a request for a service specification and returns 0 in all other cases. Notice that if global knowledge about the potential customer class is available, then  $\text{Req}(i) = 1$  and  $q(i) = 0$  for all customers and we can calculate the overall fitness of service in a service network.

### 3.2 Calibration of Service Fitness

To illustrate the effects of different observation windows we’ve depicted the calculation of service fitness for the same scenario and for three different observation windows. Figure 4.  $f^1$  depicts the use of the last two intervals  $\tau_i$  and  $\tau_{i-1}$ ,  $f^{10}$  (Figure 5) uses the last 10 intervals, and  $f^\infty$  (Figure 6) uses all available intervals to calculate the fitness of a service. As shown in the figure, the history of the service has a direct impact on the calculated service fitness by smoothing the fitness function. Notice that we didn’t change the function for establishing the lower bound of the fitness corridor. Depending on the dynamics of the service network different observation windows can be useful. In this context we speak of *re-calibration* of the service fitness function, since the past history is not considered in the calculation of the service fitness. For example, if a service provider enters a network with high fluctuation, the service provider might use a small observation window for its fitness function (e.g.,  $f^2$ ). Consider also the case of a service provider that introduces a new service version into the service network.

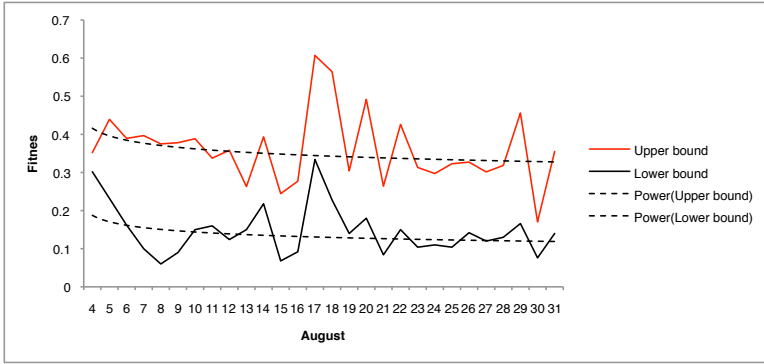


Fig. 4. Fitness corridor using  $f^1$

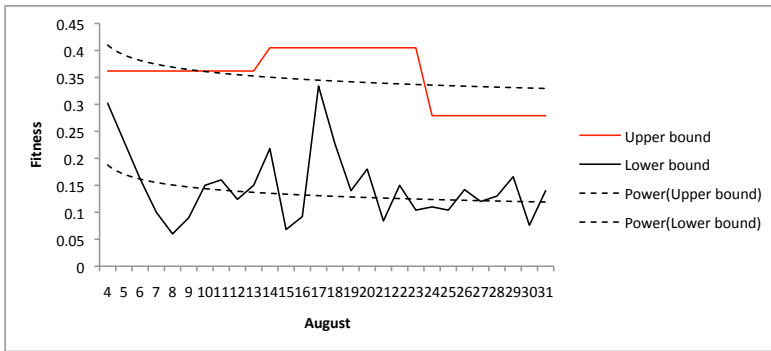


Fig. 5. Fitness corridor using  $f^{10}$

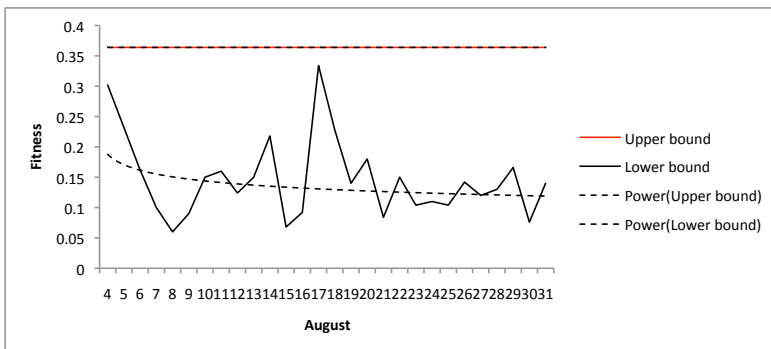


Fig. 6. Fitness corridor using  $f^\infty$



The provider is interested in ranking [9] both service versions and thus needs to re-calibrate the fitness function of the older service accordingly.

Another reason for a re-calibration of the observation window is the *fitness of the (fitness) function* itself. The chosen observation window might not return adequate fitness values for the given network dynamics. For example, a provider might use an observation window of only one time window ( $\tau_0$ ) which can result in good service fitness (see Figure 4) but may not be accurate enough to capture the changing dynamics of the service fitness over time.

The length of the time windows is also set by the service provider and is assumed to be of constant over time (in the example being 24 hours). As there is no global notion of time in the service network, each provider can decide on his own about the length of the time interval. Note that the length of the interval is critical for the calculation of service fitness.

### 3.3 Using Local Fitness for Provisioning Decisions

From the definition of equation 1 it becomes apparent that the closer the service to the upper bound of fitness is, the better the performance of the service with respect to its (estimated) market share is. Additionally, the width of the fitness corridor reflects the actual market share of a service provider. It is worth noticing, that big discrepancies between the upper and lower bounds of fitness require of the service provider either to change the estimation of the service market size or to consider modifying or decommissioning the service.

For prediction purposes we can also select portions of the fitness graph that can serve as decision criteria. For instance, during the observation, we can use functions that predict the time when the service fitness falls below a pre-defined threshold. Depending on the model and on the dynamics of the Web service ecosystem, we can use approximations to calculate the time when a Web service might fall below a critical fitness threshold and requires decommissioning. To illustrate this, recall the example from the previous section. The trend lines obviously show a downward tendency and thus it can be expected that - if the trend continues - the fitness will fall below 10 per cent within the next two months.

Quality of service criteria could also be used for these types of decisions since they offer fine grained metrics for the comparison of services. However, in service networks, such mechanisms may not be applicable due to the lack of information from competitors on the side of the service provider. Even if a service provider constantly monitors the QoS, these might not change. For example, a service may be available with no exception, the response may remain constant, and the service use also may fluctuate within predefined boundaries. However, the service ecosystem may change (e.g., new customers, new service providers, service providers leave) without having impact on QoS attributes. Nevertheless, in business terms, a service that loses market share cannot be considered as competitive (fit) and at some point in time the service might even become obsolete.

## 4 Related Work

Software evolution has been subject to studies for several decades. In this regard, the work of Lehman [10] [11] and Cook [12] is of relevance for our work. In particular, the postulated constant change of software is reflected by our notion of changing service fitness. As noted by Nehaniv et. al. [13] evolutionary concepts have to be carefully transferred into the realm of software engineering. The same is true for our work with regard to service fitness. Thus, we define service fitness as fraction between use and interest of a given service in a service network.

More general, Value Networks [14] are of interest when business aspects of service networks are studied, i.e., the value that can be generated by such networks. Likewise, from a business oriented perspective perspective, the work of Basole et. al. [15] is of relevance for our work. Their conceptual approach models service value and it's exchange in service value networks. In a similar fashion, Caswell et. al. [16] explain how the concept of value be used to study service networks.

Our work is similar to that latter work in spirit, but differs in the approach. We assume that available service network information is limited and the calculation of the importance or value of a service can only be made with local data. Consequently, we consider service fitness as utility function on the service provider side that can be combined with additional functions, like cost or turnover.

Complementary to our work, Bitsaki et. al. [4] present a service network notation which describes the interactions of service network participants. Our work shares similarities in terms of having the goal to provide methods to calculate the value or fitness of given services in service networks. The main difference is that our proposed fitness function can be combined with other utility functions, like cost or generated value. Furthermore, since complete service network information might not be available, our approach is able to calculate local fitness without the requirement of complete network information.

## 5 Conclusion and Future Work

In this paper we showed how to calculate the fitness of services from a provider perspective in service networks using locally available information for that purpose. In particular, we used the calculated service fitness to define a fitness corridor with upper and lower bounds that confine the service fitness area. After having established a fitness corridor, we showed how to calibrate the fitness calculation parameters to better reflect the service market and how to use the calculated fitness trends for making decisions about the provisioning of a service.

In future work we are going to evaluate our approach by using a Web service testbed [17] to provide simulations of service networks and illustrate the consequences of changing service fitness with regard to different service selection policies in service networks. A direct application of this process would also allow to model fitness of composite services in service networks which we are going to

investigate in future work. Furthermore we plan to combine our fitness utility function with other similar functions (e.g., number of hits or generated value) in order to provide additional metrics for estimating the service fitness.

## References

1. Papazoglou, M.P.: The challenges of service evolution. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 1–15. Springer, Heidelberg (2008)
2. Treiber, M., Truong, H.L., Dustdar, S.: Semf - service evolution management framework. In: 34th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2008, pp. 329–336 (2008)
3. Barros, A.P., Dumas, M.: The rise of web service ecosystems. *IT Professional* 8(5), 31–37 (2006)
4. Bitsaki, M., Danylyevych, O., van den Heuvel, W.J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C., Papazoglou, M.: An architecture for managing the lifecycle of business goals for partners in a service network. *Towards a Service-Based Internet*, 196–207 (2008)
5. Macfarlane, P.: Structuring and measuring the size of business markets. *International Journal of Market Research* 44(1), 7–30 (2002)
6. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: *Web Services – Concepts, Architectures and Applications*. Springer, Heidelberg (2004)
7. Microsoft: Uddi shutdown (2006)
8. Metropolis, N., Ulam, S.: The monte carlo method. *Journal of the American Statistical Association* 44(247), 335–341 (1949)
9. Ahmadi, N., Binder, W.: Flexible matching and ranking of web service advertisements. In: MW4SOC 2007: Proceedings of the 2nd workshop on Middleware for service oriented computing, pp. 30–35. ACM, New York (2007)
10. Lehman, M.M., Ramil, J.F.: Software evolution: background, theory, practice. *Inf. Process. Lett.* 88, 33–44 (2003)
11. Lehman, M.M.: Laws of software evolution revisited. In: Montangero, C. (ed.) EWSP 1996. LNCS, vol. 1149, pp. 108–124. Springer, Heidelberg (1996)
12. Cook, S., Harrison, R., Lehman, M.M., Wernick, P.: Evolution in software systems: foundations of the spe classification scheme: Research articles. *J. Softw. Maint. Evol.* 18(1), 1–35 (2006)
13. Nehaniv, C.L., Hewitt, J., Christianson, B., Wernick, P.: What software evolution and biological evolution dont have in common. *software-evolvability*, 58–65 (2006)
14. Allee, V.: Reconfiguring the value network. *Journal of Business Strategy* 21(4) (2000)
15. Basole, R.C., Rouse, W.B.: Complexity of service value networks: conceptualization and empirical investigation. *IBM Syst. J.* 47(1), 53–70 (2008)
16. Caswell, N.S., Nikolaou, C., Sairamesh, J., Bitsaki, M., Koutras, G.D., Iacovidis, G.: Estimating value in service systems: a case study of a repair service system. *IBM Syst. J.* 47(1), 87–100 (2008)
17. Juszczyk, L., Truong, H.L., Dustdar, S.: Genesis - a framework for automatic generation and steering of testbeds of complexweb services. In: ICECCS 2008: Proceedings of the 13th IEEE International Conference on on Engineering of Complex Computer Systems, Washington, DC, USA, pp. 131–140. IEEE Computer Society, Los Alamitos (2008)

# Applying Process Mining in SOA Environments

Ateeq Khan, Azeem Lodhi, Veit Köppen, Gamal Kassem, and Gunter Saake

School of Computer Science, University of Magdeburg, Germany  
ateeq,azeem,veit.koeppen,gamal.kassem,saake@iti.cs.uni-magdeburg.de

**Abstract.** Process mining is an emerging analysis technique, which extracts process knowledge from data and provides various benefits to organizations. In Service Oriented Computing environment, different services collaborate with others to carry out the operations and therefore overall picture of operations and execution is not clear. Process mining extracts the information from log files of systems, as recorded during executions, and depicts the reality. In order to apply process mining, extraction of process trace data from log files is a pre-requisite step. A case study demonstrates the practical applicability of our proposed framework for extraction of the process trace data from application systems and integration portals.

**Keywords:** Business process analysis, Process trace data, Log files, SAP Process Integration, Process mining.

## 1 Introduction

In Service Oriented Computing, different services collaborate with one another to perform tasks. The same situation occurs in enterprises where business operations are carried out by different service interactions. This is due to several characteristics of involved elements, availability of services, resources, employee's experiences. Therefore, the operations or business processes can be executed in different ways as compared to a pre-defined way. This deviation motivates to apply process mining for business process analysis.

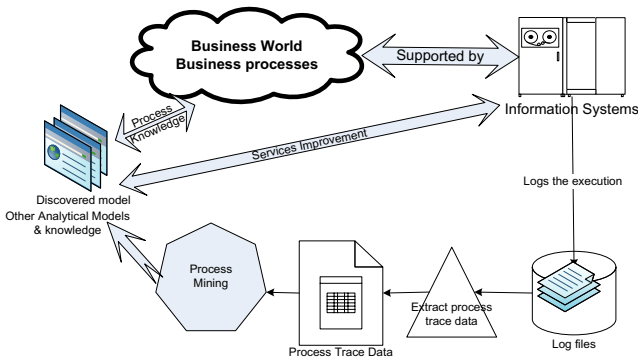
Business process analysis is a basic step for adaption and improvement of systems. Adaptive systems require an actual and consistent picture of the current environment, Process mining is a good candidate for this purpose [1]. Similarly, in case of embedded systems, process mining can be applied for monitoring interactions of devices, performance analysis and execution of tasks. In literature, benefits of process mining with respect to different perspectives are highlighted, e.g. process discovery [2], conformance checking [3] (comparison between designed model and the actual model in execution), social network analysis (who is responsible and collaborating with whom) [4]. These analyses help to improve the future execution of processes and thus to achieve the business goals in an efficient way. To apply process mining, the extraction of process data from log files is a basic and challenging step [5,6]. We address it and provide a framework in Sect. 3 to extract process trace data. Our proposed framework is illustrated with a case study in Sect. 4 SOA-based SAP Netweaver Platform.

## 2 Basics

In this section, we give an overview of the basics required for this paper. First, we introduce process mining. Furthermore, we give some basic insights to the SAP environment to understand the execution environment of our case study.

### 2.1 Process Mining

Process mining aims to extract process knowledge and to discover process models from executions of business processes as recorded by information systems [5]. The overall approach of process mining is shown in Fig. 1.



**Fig. 1.** Overall Approach of Process Mining

**Log Files.** Information systems execute business processes and record the data about elements in the form of log files or as records in database tables. In this work, we refer to log or trace files as interchangeable terms. Log files contain data about occurrences of events, provided inputs, processing information, generated outputs, message exchanges, usage, and condition of resources during execution. These log files are starting point for process mining.

**Process Trace Data.** To apply process mining, it is assumed that the information system records data of events, cases, time stamp of an event, sequence of activities, performer, and originator [4]. These log files also contain unstructured and irrelevant data, e.g. information on hardware components, errors and recovery information, and system internal temporary variables. Therefore, extraction of data from log files is a non-trivial task and a necessary pre-processing step for process mining. Business processes and their executions related data are extracted from these log files. Such data are called process trace data. For example typical process trace data would include process instance id, activity name, activity originator, time stamps, and data about involved elements.

**Conversion, Mining, and Analysis.** Extracted data are converted into the required format, depending on the process mining tools or algorithms. For social

network analysis, information on originator or performer of activity is important, performance based analysis requires for instance time related data. Therefore, the perspective of analysis depends on available data. Several process mining tools and techniques for process knowledge discovery are discussed in [7].

## 2.2 SAP Netweaver Platform and SAP Process Integration

SAP Enterprise resource planning (ERP) systems are used to provide integrated data and processes across all departments. SAP NetWeaver provides a facility for integrating SAP and non SAP systems. SAP NetWeaver uses open integration (providing interoperability) to connect with other systems or existing solution.

SAP Process Integration (SAP PI) is an Enterprise Application Integration product. It resides in process integration layer of SAP NetWeaver to integrate internal and external processes. SAP PI system works as a central hub and all systems communicate with each other through it. It removes inconsistencies between heterogeneous systems. These inconsistencies arise due to different requirements of formats, interfaces, protocols, and connectivity between SAP and non-SAP applications (for A2A and B2B systems).

## 3 Framework for Extracting Process Trace Data

We depicted a framework in Fig. 2 for extracting process trace data from SAP NetWeaver log files. However, this is not only applicable to a SAP system, but it can also be utilized in other platforms for process mining. The integration portal maintains the logs of different system interactions. Firstly, logs are collected from information systems and integration portals. Either the logs are collected manually from each involved system or the integration portal provides the services for log collection.

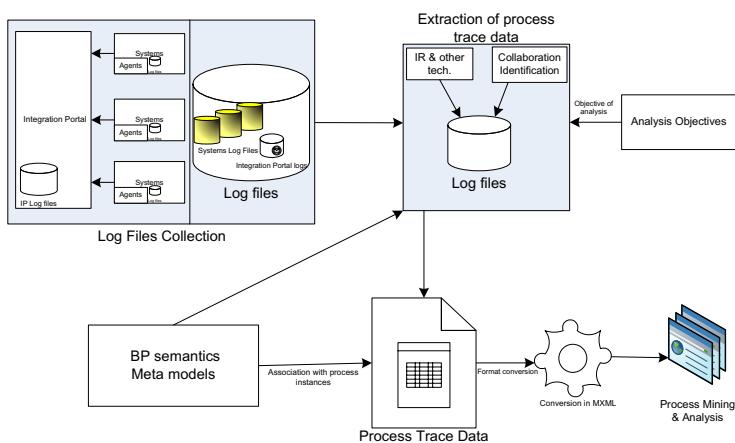


Fig. 2. Extracting process trace data in heterogeneous environment

Different system vendors record and describe the events according to their own standards, internal structures, and languages [8]. Some vendors provide meta models to describe the structure of their log files. These descriptions are used for understanding and extraction of the process trace data from log files. Some meta models of workflow management systems are discussed in [7].

Sometimes, meta models are not available or logging notation (standards, meta models) is not followed completely. This requires the understanding of systems and business semantics. In such situation, logs are manually analyzed to extract the structure. For example data related with events are confounded with system data, and spanned across various log files and tables.

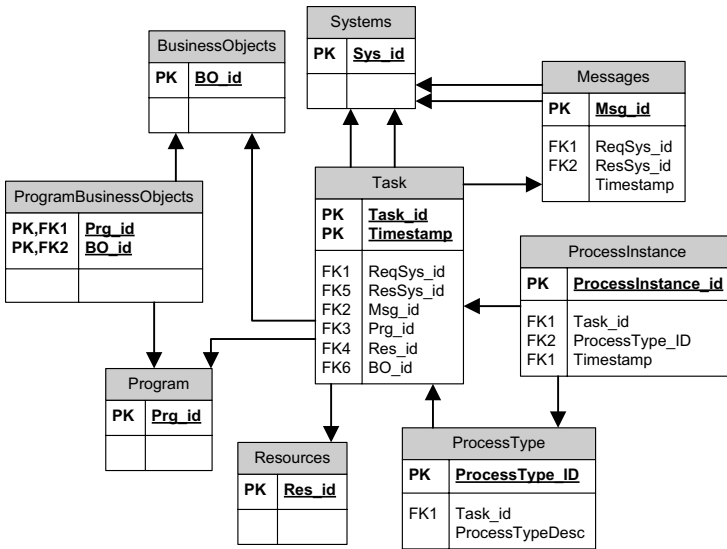


Fig. 3. Generic Meta Model of log files and process trace data

A generic meta model is presented to extract and relate the data of log files in Fig. 3. In the meta model, the entity *ProcessType* describes the business process like receiving an order, manufacturing, and shipping. *Task* is the atomic unit of work, and can belong to more than one *ProcessType* entity (e.g. get-CustomerDetails can be used in receiving order as well as in shipping business process). *Process instance* is the representation of a customer’s request. The request passes through different phases of a process for operations, and therefore, it contains more than one *ProcessType* and *Task*. *Resources* perform different tasks on the process instance and modify/change the state of *Business Objects* attached to them. The *Programs* record the change of state of business objects.

In heterogeneous environments, different systems are involved in the business process execution. These systems communicate with each other to execute business processes within or outside the enterprise. The content of a message describes the nature of operations and involved objects. Integration Portal log files

contain data such as requesting systems, contents of request, response, and performer, time stamps of requests/responses, message ids, and message contents, and resources. Such data can be used as process trace data to identify the collaboration between systems. Organizational business data (operational) is also used for relating/associating the data. For example in customer-order scenario, customer order number can be used to relate the customer name, ordered items, and other data. Data from log file are extracted and used to build the execution instances.

Various elements are not considered, e.g. activities, screens (and GUIs), states of processes, and business objects. We present a quite generic model. It is extendable based on analysis objectives, for application usage mining, the *Program* entity is extended and related with the GUIs presented to users, elements contained, and user actions.

## 4 Case Study and Extraction of Process Trace Data

We explained our framework in a case study. Different SAP systems and web services are integrated by SAP PI to complete the business processes.

The case study is a simple supply chain scenario; a customer visits the website and places an order. Order items received as a file in SAP PI from the website through http-file adapter. This file is transferred, using mail adapter, from SAP PI as formatted email to the distributor. Stock is checked and if the item is not available then a production order, containing item details, is issued from SAP R/3 system using RFC and IDOC adapter to SAP IDES system used at the manufacturing site. To ship, the total weight and customer destination are forwarded to shipping web service, which calculates a freight charge for the order and notifies the customer after shipping.

We use SAP PI for analysis and monitoring of systems (services). User trace loggings are activated in SAP systems and monitoring is done during execution. After execution, tracing is deactivated, and we collect trace files.

We develop a tool(output is depicted in Fig. 5) to extract process trace data from SAP trace files. Development of the tool consists of following phases and steps in 5. We divide the process into four phases:

### 4.1 Contents and Structure Analysis in Log Files

SAP does not provide meta models of log files, so we study log files and different repositories of SAP systems for extraction. In normal cases, SAP transaction data are stored in log files and used for process mining. Analysis of log files enables us to find patterns of characters that occur with events, and these characters help in extraction of process trace data from the systems. For process trace data, diag processor (Dialog work processes deal with requests from an active user to execute Dialog steps) and task handler component are required. Task handler contains user information and time of activities. Detailed analysis of SAP trace files and important structure used for extracting required data from trace file are described in 9.



**Screen Messages:** Screen messages are shown to users during business process execution. These messages indicate flow of the system. Information stored with these messages are message type, message time, process component, associated user, user terminal, server, and executed transactions.

**Extraction of Database Fields:** Elements in trace file are represented as entity types, also called etypes. Etypes have data fields attached with them, which belong to screen elements, e.g. text fields or data displayed from the database. Etypes have a specific structure and database field names are extracted after meeting conditions, e.g. a mark 'X' present on line relative to etypes type line in trace file. The database fields are assigned descriptive names. Thus, the semantics and structure of log files detected in this part help to identify important elements in log files for process trace data.

## 4.2 Structure Analysis in Integration Portal

We identify that trace files from SAP PI do not contain information on executed transactions or any exchanged data, e.g. when RFC module or IDOC adapter data are exchanged. In SAP PI, the logs are maintained but at the user interface level only very limited view is provided. Furthermore, trace data, of exchanged messages between systems, are stored in SAP PI system tables. In Fig. 4 a meta model for extraction of data is presented. For each message exchange, communication systems are involved as message sender or receiver. Message's contents are mapped between communication systems by mapping relations and condition involved for these mappings. Each message has a message id, associated sender, receiver, and message payload. Payload is transferred data and contains attributes, which we use for process mining and different perspectives of analyses. The time stamps of exchanged messages and other data provide connections where processes are collaborating with the system.

## 4.3 Collaboration Data from Integration Portal

Database tables and descriptions identified in previous steps are used to extract the data. With SAP Developer role in SAP PI, ABAP programs can be written to get data from these tables and store them in separate files. This will make the extraction of data for evaluation easier. Depending on the requirements, tables are further investigated and necessary data are extracted.

## 4.4 Extraction of Data from Log Files

In this step, data are extracted from log files. Data may consist of user details, screens displayed to a user, transaction executed, confirmation or error messages, and data objects against database fields found in section 4.1. These elements are extracted and saved sequentially. Extraction of elements follows a specific structure in the trace file. Extracted data may contain redundant information, which we remove by further processing.

Extracted process trace data are used to generate case instances and to use in process mining tools as explained in Section 5.

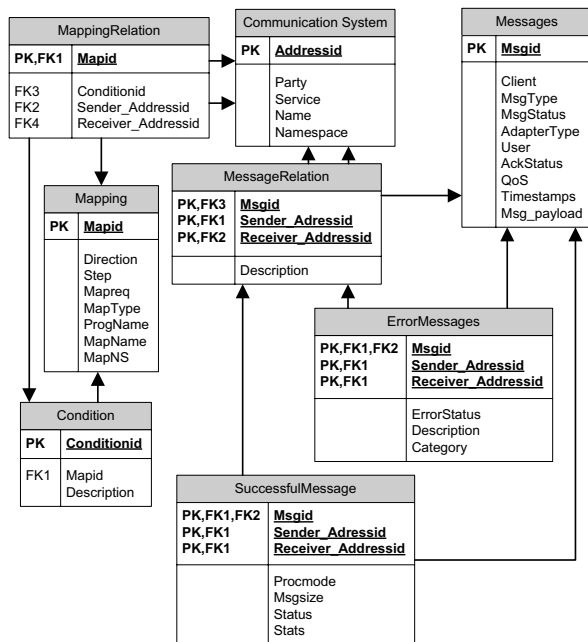


Fig. 4. SAP PI meta model for processed trace data

## 5 Relationship between Data and Process Mining

A primary key attribute is selected from the process trace data, e.g. key attribute may be order id, customer no., application no., or product id. All extracted information is sorted on this key attribute and written as cases. Therefore, the extracted data are correlated based on these and case instances are built. Business semantics and meta models presented in earlier sections can be used for this purpose. The extracted information is converted into a suitable format for mining and analysis tools. We provide the extracted data into two formats. In one format, cases are assigned with specific functions of log files, while in other format case numbers, object values, and functions, in which they are executed are described, see Fig. 5. We add SAP trace data conversion plug-in in ProM Import Framework [11] and use it conversion of our data into Mining XML (MXML) format. ProM [4] tool is used for analysis and mining. The actual process model is discovered. Examples of two different business scenario models are depicted in Fig. 6. Other analyses can also be done as discussed in Section 2.1. This provides the opportunity to analyze the business processes in different perspectives and directions for improvement.

<sup>1</sup> www.processmining.org

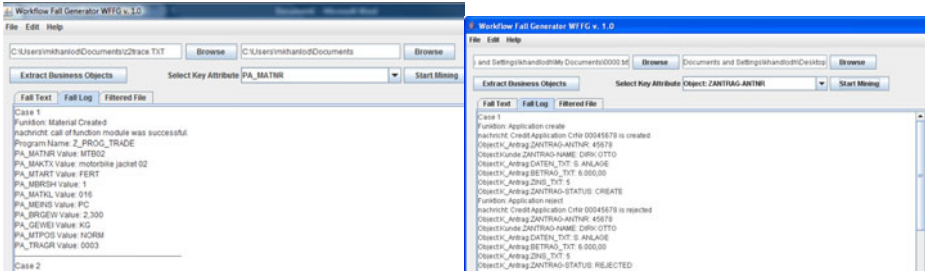


Fig. 5. Case data

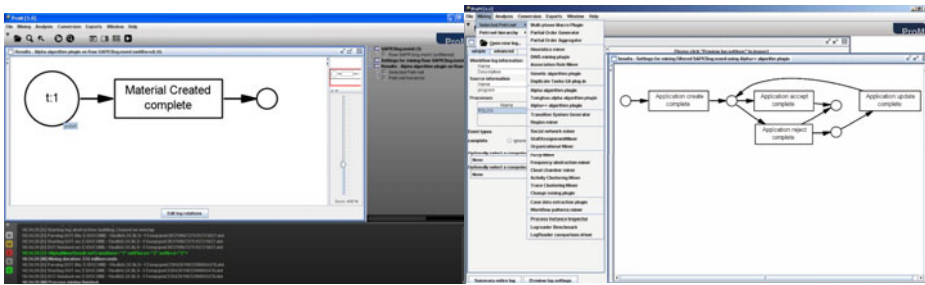


Fig. 6. AS-IS models from Prom tool [10]

## 6 Related Work

Process mining is an active research area in business process analysis and discussed in many papers [5,7,12,13]. Most of the papers focus on different kinds of analyses and benefits that can be achieved. These approaches assume that process trace data are already available. Extracting process trace data is a challenge and important as discussed in [5,6]. Different approaches are proposed to address this challenge. We categorize them in three major approaches:

**Table-based approaches.** Attempts are made to extract process data from ERP systems log files, e.g. SAP, PeopleSoft. In [14], the author tries to extract process trace data from SAP R/3 log files, but it was not fully satisfying because process execution information is too detailed due to the usage at the transaction level and scattered in many tables. In [15], authors built a tool for constructing process instances from business process executions in SAP systems. They use SQL statements to identify used business objects in SAP tables and correlate them with events to construct instances. However, this table-based approach provides less data for process mining analysis as not all data are stored with business objects in tables.

**Data warehouse based approaches.** Data warehouse technique is introduced in [16]. HP Process Manager (HPPM) system logs are used, so this approach is not generic. The work is further extended as architecture for analysis and monitoring in [17]. The architecture is related to our presented framework, but we focus more towards the extraction of process trace data from log files rather than analysis techniques. Information from process logs are extracted by querying log tables and stored in a process data warehouse. Mining algorithms are applied in the data warehouse.

**Log files based approaches.** In [7,8], meta models are proposed to correlate elements of log files. The difference from our approach is that they concentrate only to extract a small part of process trace data from log files, e.g. case id, task, timestamps, focus only for process discovery, and do not address the challenge in heterogeneous environments. We present more generic work and deals with log files as well as log tables in heterogeneous systems. In [18], data from log files are used to analyze the interaction behaviour of users with applications.

## 7 Conclusion and Outlook

We proposed a framework to apply process mining in heterogeneous environments. We develop a meta model to extract more process trace data from logs, so process mining can be applied to those systems, which are not process-driven. SOA based platform (SAP NetWeaver) is used as a case study to show the applicability of our framework and process mining in a SOA environment. Monitoring components of SAP NetWeaver are discussed for the extraction of process trace data. We also developed a tool for the extraction of process trace data from the logs of involved systems. We believe that this work provides a guideline to apply process mining in different SOA based systems and steps toward better analyses and monitoring services.

It would be interesting to investigate cases in which more than one integration portal is involved. Organizations may not be interested to share their data. Privacy preserving methods should be developed for this purpose.

**Acknowledgment.** Ateeq Khan is supported by scholarship from federal state of Saxony-Anhalt, Germany. Veit Köppen is funded by the German Ministry of Education and Science (BMBF), project 01IM08003C. The presented work is part of the ViERforES<sup>2</sup> project.

## References

1. van der Aalst, W.M.P., Günther, C., Recker, J., Reichert, M.: Using process mining to analyze and improve process flexibility. In: Latour, T., Petit, M. (eds.) Proceedings of the CAiSE 2006 Workshops / 7th Int'l Workshop on BPMDS 2006, Namur, June 2006, pp. 168–177. Presses Universitaires de Namur (2006)

---

<sup>2</sup> [www.vierfores.de](http://www.vierfores.de)

2. Weijters, A.J.M.M., Maruster, L.: Workflow mining: Discovering process models from event logs. *IEEE Transactions on KDE* 16, 2004 (2004)
3. van der Aalst, W.M.P.: Business alignment: using process mining as a tool for delta analysis and conformance testing. *Requir. Eng.* 10(3), 198–211 (2005)
4. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. *Comput. Supported Coop. Work* 14(6), 549–593 (2005)
5. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: A research agenda. *Computers in Industry* 53, 231–244 (2004)
6. van der Aalst, W.M.P.: Challenges in business process analysis. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) *Proceedings of the 9th ICEIS. Lecture Notes in Business Information Processing*, vol. 12, pp. 27–42. Springer, Heidelberg (2007)
7. Muehlen, M.Z.: *Workflow-based Process Controlling. Foundation, Design, and Implementation of Workflow-driven Process Information Systems. Advances in Information Systems and Management Science*, vol. 6. Logos, Berlin (2004)
8. van Dongen, B.F., van der Aalst, W.M.P.: A meta model for process mining data. In: *Conference on Advanced Information Systems Engineering (CAiSE) Workshops*, vol. 161, p. 209 (2005)
9. Khan, A., Lodhi, A.: *Analysis of Business Processes in Heterogeneous Environment: SAP as a Use Case*. Master thesis, School of Computer Science, University of Magdeburg (February 2009)
10. Dongen, B., Medeiros, A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The proM framework: A new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) *ICATPN 2005. LNCS*, vol. 3536, pp. 444–454. Springer, Heidelberg (2005)
11. Günther, C., van der Aalst, W.M.P.: A generic import framework for process event logs. In: Eder, J., Dustdar, S. (eds.) *BPM Workshops 2006. LNCS*, vol. 4103, pp. 81–92. Springer, Heidelberg (2006)
12. Greco, G., Guzzo, A., Manco, G.: Mining and reasoning on workflows. *IEEE Transactions on KDE* 17(4), 519–534 (2005)
13. Tiwari, A., Turner, C., Majeed, B.: A review of business process mining: State of the art and future trends. *BPM Journal* 14, 5–22 (2008)
14. van Giessel, M.: *Process mining in sap r/3*. Master's thesis, Eindhoven University of Technology, Eindhoven (2004)
15. Ingvaldsen, J., Gulla, J.: Preprocessing support for large scale process mining of sap transactions. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) *BPM Workshops 2007. LNCS*, vol. 4928, pp. 30–41. Springer, Heidelberg (2008)
16. Casati, F., Shan, M.-C.: Semantic analysis of business process executions. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) *EDBT 2002. LNCS*, vol. 2287, pp. 287–296. Springer, Heidelberg (2002)
17. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.-C.: Business process intelligence. *Computers in Industry* 53, 321–343 (2004)
18. Kassem, G.: *Application Usage Mining: Grundlagen und Verfahren*. PhD thesis, School of Computer Science, University of Magdeburg (2007) ISBN: 3832259953

# Monitoring within an Autonomic Network: A GANA Based Network Monitoring Framework

Anastasios Zafeiropoulos<sup>1</sup>, Athanassios Liakopoulos<sup>1</sup>, Alan Davy<sup>2</sup>,  
and Ranganai Chaparadza<sup>3</sup>

<sup>1</sup> Greek Research & Technology Network, Av. Mesogion 56, 11527 Athens, Greece  
{tzafeir, aliako}@grnet.gr

<sup>2</sup> Telecommunications Software & Systems Group, Waterford Institute of Technology  
Cork Road, Waterford, Ireland  
adavy@tssg.org

<sup>3</sup> Fraunhofer FOKUS Institute for Open Communication Systems, Kaiserin-Augusta-Allee 31,  
D-10589, Berlin, Germany  
Ranganai.Chaparadza@fokus.fraunhofer.de

**Abstract.** The concept of self-managing of autonomic networks is a paradigm shift from today's management models, aiming at enabling networked nodes to self manage their behaviour within the constraints of the operator's policies and objectives. In this article, we present our approach for self-coordinating monitoring functions within such an autonomic network. This approach complies with the principles of a recently introduced *Reference Model* for autonomic network engineering/self-management within node and network architectures dubbed: the *Generic Autonomic Network Architecture (GANA)*, which aims to identify autonomic behaviours realised via hierarchical control loops among self-managing elements. The components of the proposed monitoring framework, the interactions among the identified elements and a complete use case scenario are described in detail.

**Keywords:** Autonomic network engineering, monitoring, self-management, GANA, Hierarchical Control-Loops (HCLs) framework.

## 1 Introduction

The vision of the Future Internet, is of an autonomic network whose nodes are engineered in such a way that all the traditionally so-called network management functions defined by the FCAPS (Fault, Configuration, Accounting, Performance, Security) management framework [1], as well as the fundamental network functions such as routing, forwarding and monitoring, are designed to automatically feed each other with information and effect feedback processes among the diverse functions. The feedback processes enable reactions in functions of the network and of individual nodes, in order to achieve and strive to maintain some well defined goals of the network.

Autonomic principles are required in order to accomplish management of dynamic, heterogeneous and complex networks, where each network entity needs to be able to take network optimisation decisions. FCAPS functions have to be intrinsically in-built

into node architectures apart from being part of an overall network architecture - whereby traditionally, a separate management plane is engineered separately from the other functional planes of the network. This means that the functional planes of an autonomic network would need to be (re)-defined, re-factored or even merged.

In this paper, we start by briefly presenting the key principles of the Generic Autonomic Network Architecture (GANA) [2] that is proposed in the framework of the EU FP7-EFIPSANS IP Project<sup>1</sup> [11]. We focus on the realization of an autonomic monitoring framework within the EFIPSANS network based on GANA principles. GANA sets the principles and guidelines that need to be followed according to our vision of the Future Internet design. In contrast to any other of today's best known approaches, including clean-slate approaches, such as 4D [3], CONMan [4], a Knowledge Plane for the Internet [5], FOCAL [7], a Situatedness-based Knowledge Plane for autonomic networking [8], GANA captures in a holistic way, the generic principles required for a generic autonomic network architecture.

The paper is organised as follows; section 2 presents the fundamental concepts of the GANA architecture and a methodology for designing autonomic behaviours in compliance with GANA principles; section 3 describes the EFIPSANS network monitoring and information dissemination framework; section 4 details a use case scenario in which a number of autonomic behaviours are introduced and finally section 5 concludes the paper with a discussion of current challenges and future work.

## 2 A Brief Introduction to the Generic Autonomic Network Architecture: GANA

In contrast to existing approaches to autonomic networking, an appropriate model for an autonomic networked system is needed to be introduced that would allow us to reason and think of a particular autonomic function that implements the concept of a control loop at some particular level of abstraction. This means, we should then be able to talk about autonomic networking functions such as autonomic routing, autonomic forwarding, autonomic monitoring in the sense that the elements driving the control loops, use information learnt from its required information suppliers to control the behaviour of the functionality that is then considered to be autonomic.

Fig. 1 shows a model of an autonomic networked system and its associated control loop that we developed. It is derived and extended from the IBM-MAPE model [9] specifically for autonomic networking and is the basis for the derivation of the GANA architecture. The model is generic and illustrates the possibility of the distributed nature of the information suppliers that supply a so called Decision Element (DE) and the diversity of the information that can be used to manage the associated managed resources and elements. It can be applied to different levels of functionality and abstraction within node architectures and network architectures.

We define a Decision Element (DE) as a concept that is associated with some concrete resources or functional entities (e.g. protocols) that the Decision Element manages and

---

<sup>1</sup> The EFIPSANS-IP-Project aims at exposing the features in IPv6 protocols that can be exploited or extended for the purposes of designing or building autonomic networks and services, as necessitated by GANA.

drives its control loop. Information or views are being continuously exposed by its managed resources or functional entities, together with information coming from other required or potential information suppliers of the Decision Elements. We also adopt the concept of a Managed Entity (ME) to denote a managed resource or an automated task in general, instead of a managed element. As illustrated in Fig. 1, the actions taken by the Decision Element do not all necessarily have to do with triggering some behaviour or enforcing a policy on the Managed Entities but that, some of the actions executed by the Decision Element may have to do with communication between the Decision Elements and other entities. This is indicated by the extended span of the arrows: "Downward Information flow" and the "Horizontal Information flow", as well as the fact that a Decision Element also exposes *views* such as *events* to its "upper" Decision Element and receives *policies, goals and command statements* from its "upper" Decision Element.

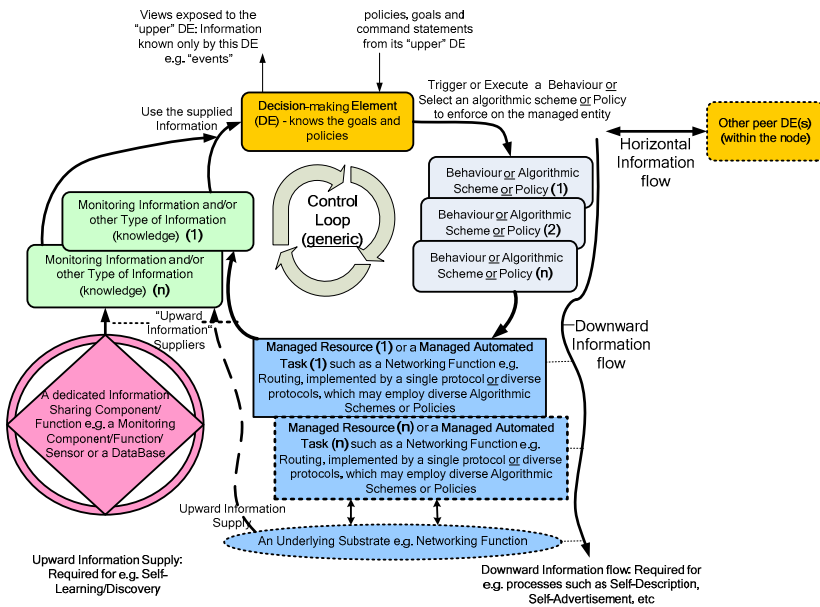


Fig. 1. A generic model for an abstract autonomic networked system

### 2.1 Types of Control Loops in GANA

GANA defines a Hierarchical Control Loops (HCLs) framework that reflects on the need for control loops operating at different levels of abstractions and complexity. Four levels of abstractions for which Decision Elements, Managed Entities and associated control loops can be designed are described below.

**Protocol Level [level 1]:** The concepts of a control loop, Decision Element, Managed Entity, as well as the related self-manageability issues may be associated with some implementation of a single network protocol or an automated task.



**Function Level [level 2]:** On a higher level of abstraction, concepts of a control loop, Decision Elements and Managed Entities may be addressed on the level of abstracted network functions, such as routing, forwarding, mobility management, QoS management, etc.

**Node Level [level 3]:** On a higher level of autonomic networking functionality, the concepts of a control loop, Decision Elements, Managed Entities, as well as the related self-manageability issues may be associated with a system or node as a whole. The node's main Decision Element has access to the "views" exposed by the lower level Decision Elements and uses its knowledge of the higher level to influence the lower level Decision Elements to take certain desired decisions, which may in turn further influence or enforce desired behaviours on their associated Managed Entities, down to the lowest level of individual protocols.

**Network Level [level 4]:** The highest level of control loops is the network level. There may exist a logically centralized Decision Element (DE) or DEs such as the ones proposed in the 4D network architecture [3] that knows the objectives, goals or policies to be enforced by the whole network. The objectives, goals or policies may actually require that the nodes' main Decision Elements of the network export "views" such as events and state information to the logically centralized Decision Element(s), in order for the network-level DEs to influence or enforce the Decision Elements of the nodes to take certain desired decisions. A distributed network level control loop is implemented following the above set-up, while another would involve the main Decision Elements of nodes working co-operatively to self-organize and manage the network without the presence of a logically centralized Decision Element(s).

The hierarchies of Decision Elements in GANA, which allow for some decisions to be taken autonomously at different levels of control and complexity, may have the following forms: Hierarchical relationships between Decision Elements where lower level Decision Elements are managed by their upper level Decision Element and Peer-relationships between Decision Elements, which facilitate communication between Decision Elements for exchanging information.

## 2.2 The Functional Planes of GANA

In GANA [2], we first take the position that the functional planes known in today's world of networking can be compressed - with merging and re-factoring some of the planes - into four functional planes that are still called: the Decision Plane, the Dissemination Plane, the Discovery plane and the Data plane, which we adopt from the 4D architecture [3], but, we re-define them for GANA because we introduce the concept of an autonomic Decision Element into the architecture of every network node/device, as opposed to the 4D, which assumes that network nodes/devices need not have Decision Elements within them.

## 3 EFIPSANS Network Monitoring Framework

Monitoring information is a fundamental part of a wide number of network functionalities, such as QoS Management, Routing, and Mobility. However there may be a

great deal of operational overhead involved in the configuration / re-configuration / optimisation of these operations to ensure that the information being supplied is of sufficient accuracy for the corresponding operations. In this section, we propose a Monitoring Framework that is aligned with GANA principles, extending autonomic monitoring principles presented in the literature. The monitoring framework describes the basic functions needed for management of monitoring activities within an autonomic network. By developing a Monitoring Decision Element responsible for the configuration of the monitoring protocols and mechanisms, other function level Decision Elements within a node and within the network can be guaranteed that relevant and sufficiently accurate information is available to drive their control loops.

### 3.1 The Monitoring Decision Element and Its Functionality

The Monitoring DE (Mon\_DE) - shown in Fig. 2 - resides in the function level and controls all the actions of an autonomic node related to monitoring. It also interacts with the rest of the Decision Elements or the Managed Entities residing in the same or other autonomic nodes, whenever these entities require specific monitoring actions to be taken or monitoring mechanisms to be activated. It is possible to define multiple control loops at a function level, each of them implementing a specific monitoring function based on decisions made by the Mon\_DE.

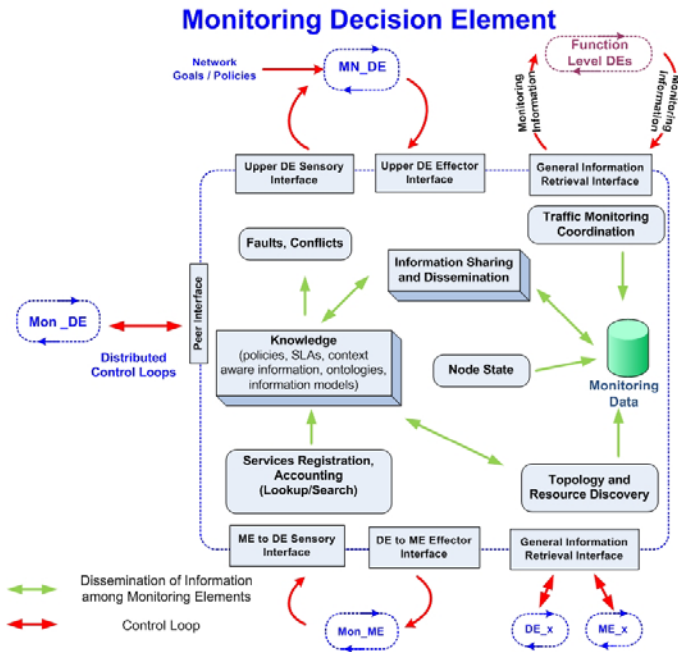


Fig. 2. The Monitoring Decision Element and its interfaces

The Mon\_DE is responsible to orchestrate Monitoring Managed Entities (Mon\_MEs) so as the node level monitoring policies are fulfilled and specific functions are realised. It is responsible for monitoring actions or mechanisms such as:

- *traffic monitoring coordination*: a) manage mechanisms for traffic monitoring and schedule monitoring measurements in such a way that they do not degrade other services –such as QoS or mobility services- or negatively affect collected results, b) activate active or passive measurements in the autonomic node in accordance to provided services and network context and c) store and publish any historical data collected via measurements for later analysis.
- *information sharing and dissemination*: collect information about stored data in node and network level and make it available to other Decision Elements or Managed Entities. According to the network topology, monitoring data may either be distributed across the network or stored in a centralized database. The Monitoring Decision Element is aware of the data collection and dissemination scheme that is selected and acts as an information supplier to other Decision Elements, providing them with guidelines for data acquisition and dissemination.
- *topology discovery*: collect information necessary for creating the topology of the network and publish topology data in a appropriate format to other Decision Elements or Managed Entities, e.g. the visualization Managed Entities.
- *resource discovery*: collect information regarding the available resources in network level. This can be proven very useful in ad hoc networks where nodes continuously join/leave the network and allocation and negotiation of different resources is dynamic.
- *monitor the state of the node*: monitor the available node resources and specific metrics such as available power, storage capacity and status of interfaces. The Monitoring Decision Element can interact with SNMP MIBs and request information through queries. SNMP alarms can also be generated in case of specific events.
- *provide context-aware information*: process any collected data according to the specified GANA ontology [6] in order to enable reasoning over the available information and extract meaningful events. By increasing the context-awareness level of monitoring data exchanged by autonomic nodes, it is possible to efficiently sense changes in the network and the provided services and proceed to corrective actions (self-healing).
- *fault and conflict management*: diagnose a) faults in node and network level, b) violation of guarantees for predefined performance metrics and c) firewalling and security alarms. The Monitoring Decision Element is able to identify conflicts in the policies imposed by other Decision Elements and trigger accordingly the responsible Decision Elements in order to resolve the conflicts.
- *accounting and registration to network services*: identify services supported in the autonomic network, enable subscription to services and access to specific information and provide related information to other Decision Elements.

### 3.2 Basic Interactions and Interfaces of the Monitoring Decision Element

The Monitoring Decision Element interacts with the following elements:

- The Main Node DE (MN\_DE) residing at the node level establishing a hierarchical relationship. Within this relationship, the Mon\_DE operates as a Managed Entity, i.e. only provides information and alarm events, and offers a management interface to receive node or network level policies.
- The various Decision Elements at the function level, such as the routing and mobility Decision Elements, establishing a peering relation where it operates as an information supplier.
- The various Monitoring Managed Entities (Mon\_MEs) and various Decision Elements and Managed Entities at the protocol level establishing a parent relationship. The Mon\_MEs implement specific (monitoring) functions required by other Decision Elements and Managed Entities to operate. The Mon\_DE operates as a Decision Element within this relationship, i.e. implements the control loops.
- Mon\_DEs at other autonomic nodes, establishing peering relationships, in order to form distributed control loops, exchange relevant monitoring information and coordinate network wide monitoring tasks.

All the Mon\_MEs at the protocol level are under the control of the Mon\_DE. Thus, when a specific Decision Element at the function level, such as the QoS\_DE, requires a Mon\_ME to be activated, it has to request it through the Mon\_DE. These restrictions are imposed in order to avoid conflicting actions, as monitoring actions consume network or node resources and may also affect other services, e.g. bandwidth aggressive measurements may cause QoS degradation to legitimate traffic. It should be noted that any Managed Entity usually participates in more than one control loops under the authority of different Decision Elements as the role of information supplier, but it is managed by a unique Decision Element.

### 3.3 Monitoring Distributed Control Loops

The GANA architecture allows the realization of distributed control loops achieved through the interactions among multiple Decision Elements located in autonomic nodes or via a logically centralized overlay of Decision Elements operating on the network-level. In the first case the decision process is distributed across multiple Decision Elements while in the second case the decision is taken by a single or a small group of Decision Elements chosen via a delegation process.

A distributed control loop, either realized via a centralized entity or through interactions of multiple entities, has to be applied in order to take any decisions on how to monitor an autonomic network and its deployed services. This loop is responsible for providing information regarding the overall state of the network. The Monitoring Decision Elements at the function level participate in the network-wide distributed control loop. Each autonomic node advertises its capabilities to other nodes relating to monitoring functions and the network-wide control loop is responsible for directing how to monitor the supported services.

## 4 A Detailed Scenario: Traffic Monitoring and QoS Control

In the following scenario, we focus on autonomicity as a feature of traffic monitoring, coupled with Quality of Service (QoS) management functions of an ingress edge

router, complying with GANA principles. As network and traffic conditions continuously change, monitoring protocols and mechanisms must be appropriately re-configured in order to facilitate the efficient QoS management of the autonomic network. The objective of QoS control at the ingress point within a DiffServ domain is to ensure that the traffic admitted to the network is appropriately classified, policed and shaped to assure performance guarantees.

#### 4.1 Description of Decision Elements, Managed Entities and Their Interactions

*Effective Bandwidth ME (EB\_ME)*: This Managed Entity is a process at the protocol level of GANA that implements the Effective Bandwidth estimation algorithm proposed in [10]. The process collects a packet trace from the network, performs a number of processing activities on it and reports an estimation of effective bandwidth for a particular QoS target of packet delay. The EB\_ME is managed by the Mon\_DE and can act as information supplier to other MEs and DEs.

*Bandwidth Availability in Real-time ME (BART\_ME)*: This is an active probing process, used to generate and inject probe packets into the network towards a destination node. The BART\_ME on the destination node estimates the amount of available bandwidth along the path between the two nodes. The management and configuration of the BART\_ME is done by the Mon\_DE on each node.

*Quality of Service DE (QoS\_DE)*: This is a function level Decision Element that participates to a node-local control loop. It aims to configure the mechanisms - such as queue management, queue scheduling, marking, policy, admission control, etc - to support service guarantees provided by the network. The QoS\_DE is responsible for configuring the node-level mechanisms and interacts with the MN\_DE and other Function level Decision Elements such as the Mon\_DE.

*Admission Control ME (AC\_ME)*: This Managed Entity ensures that traffic admitted into a node or network will not violate specified QoS performance guarantees. This process is based on the admission control algorithm presented in [10]. The AC\_ME depends on measurements of effective bandwidth of the incoming traffic, realized by the EB\_ME and is managed by the QoS\_DE.

*QoS Violation ME (QoS\_V\_ME)*: It produces estimations of performance violations for traffic exiting the network through an egress router. The violations are estimated by analysing short packet traces and results are compared with particular QoS targets. This mechanism is a modification of the EB\_ME and is managed from the Mon\_DE.

#### 4.2 Control Loops and Entities Interactions

The autonomic behaviour instilled within the ingress edge node is the ability to control the incoming traffic into a domain while maintaining a high degree of confidence in the admission decisions. This is provisioned by an interaction between the Mon\_DE and the QoS\_DE, where the traffic monitoring requirements of the AC\_ME change and the associated EB\_ME must be re-configured dynamically in order to conform to these changes. There is a dependency relationship between the lowest level MEs, i.e. EB\_ME and AC\_ME, of the ingress router, necessary for MEs to operate in an optimal manner.

It is the responsibility of the interacting control loops, managed from the QoS\_DE and the Mon\_DE, to drive the re-configuration of these lowest level MEs.

The dependency relationship is defined as follows; the AC\_ME requires effective bandwidth estimations on currently admitted traffic in order to make admission decisions. These measurements are supplied by the EB\_ME that is managed by the Mon\_DE and can be configured to supply these metrics with varying degrees of accuracy and frequency. This information along with other network state related information can help the QoS\_DE to manage the AC\_ME. However, in cases where the requests imposed by the QoS\_DE and the Mon\_DE are conflicting, interactions are established with the corresponding DEs that manage the conflicting hierarchical control loops and priorities are provided according to the specified policies. If priorities cannot be established, then the Main Node DE is responsible to resolve the conflict.

### 4.3 Information Flow and Associated Behaviours

In Fig. 3, we present the flow of information between Decision Elements and Managed Entities within the context of the GANA Monitoring Framework. The EB\_ME supplies effective bandwidth information to the AC\_ME, thus acting as an information supplier of that Managed Entity. If the QoS\_DE detects that effective bandwidth estimation on admitted traffic needs to be updated quicker to compensate for the increased number of service request arrivals, it requests from the Mon\_DE to re-configure the EB\_ME accordingly.

The BART\_ME supplies information regarding the bandwidth availability between the ingress node and the egress node to the QoS\_DE on the ingress node. As there is best effort traffic traversing the network, congestion can occur within the core network. If congestion reaches a limit, this can affect the services that can be admitted while maintaining QoS targets of the admitted traffic flows. The QoS\_DE therefore has a policy to reconfigure the AC\_ME to prioritise higher revenue services during such times of congestion within the network.

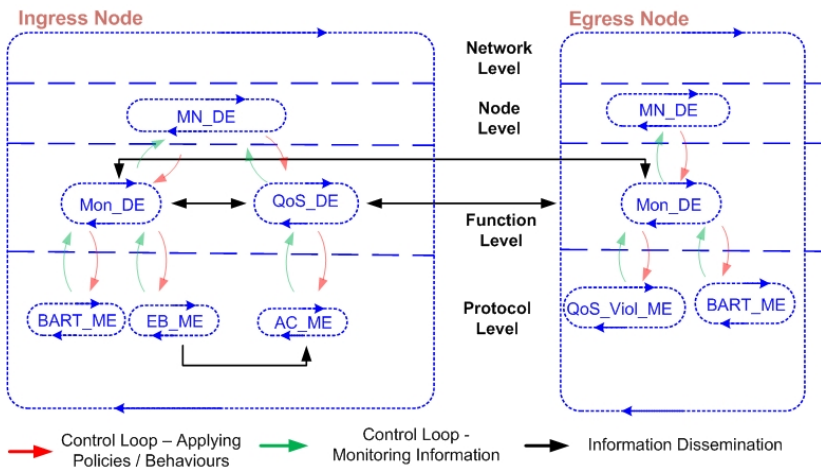


Fig. 3. Placement of DEs in different levels and their HCLs

Finally, the QoS\_V\_ME supplies information to the Mon\_DE on the ingress node for any QoS violations that occur. If, for example, QoS performance violations are experienced on one-way delay metrics, the Mon\_DE can inform accordingly the QoS\_DE, which can configure the AC\_ME to be more conservative in its admission decisions, ensuring adequate bandwidth is being reserved for admitted services. However such a configuration can impede on the AC\_ME's performance in reacting to variations in traffic. This trade off decision is the responsibility of the QoS\_DE to make and not the AC\_ME.

## 5 Conclusions and Future Work

In this paper we presented the Generic Autonomic Network Architecture (GANA), as the basis for producing autonomic behaviour specifications for selected diverse networking environments. We proposed a network monitoring framework based on GANA principles and described how the different elements and high-level entities should interact, apply policies and process information. Specific autonomic behaviours were elaborated through a detailed scenario.

In our future work, the presented scenario is going to be implemented and the designed network monitoring framework will be evaluated in the EFIPSANS testbed. Focus will be given on the instantiation of the monitoring functions and the effectiveness of the monitoring mechanisms in diverse networking environments. Furthermore, GANA will be extended by focusing on interfaces that facilitate interactions among autonomic nodes and allow administrators to define network level objectives and policies.

**Acknowledgement.** This publication is based on work carried out and funded under the framework of the European Commission ICT/FP7 project EFIPSANS.

## References

1. ITU-T, TMN Management Functions, Telecommunication Standardization sector of ITU, M.3400 (2000), <http://www.itu.int/rec/T-REC-M.3400-200002-I/en>
2. Chaparadza, R.: Requirements for a Generic Autonomic Network Architecture Suitable Requirements for Autonomic Behavior Specifications of Decision-Making-Elements for Diverse Networking Environments. In: International Engineering Consortium (IEC) Annual Review in Communications, vol. 61 (2008)
3. Greenberg, A., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., Zhang, H.: A Clean Slate 4D Approach to Network Control and Management. ACM SIGCOMM Computer Communication Review 35(5), 41–54 (2005)
4. Ballani, H., Francis, P.: CONman: A Step Towards Network Manageability. ACM SIGCOMM Computer Communication Review 37(4), 205–216 (2007)
5. Clark, D., Partridge, C., Ramming, J.C., Worclawski, J.T.: A Knowledge Plane for the Internet. In: Proc. of the 2003 conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp. 3–10 (2003)
6. Zafeiropoulos, A., Liakopoulos, A.: Context Awareness in Autonomic Heterogeneous Environments. In: INGRID 2009 (2009)

7. Jennings, B., Van der Mer, S., Balasubramaniam, S., Botvich, D., Foghlú, M.O., Donnelly, W.: Towards autonomic management of communications networks. *IEEE Communications Magazine* 45(10), 112–121 (2007)
8. Bulot, T., et al.: A Situatedness-based Knowledge Plane for Autonomic Networking. *International Journal of Network Management* 18(2), 171–193 (2008)
9. IBM, Understanding the Autonomic Manager Concept, <http://www.ibm.com/developerworks/library/acconcept>
10. Davy, A., et al.: Revenue Optimized IPTV Admission Control using Empirical Effective Bandwidth Estimation. *IEEE Transactions on Broadcasting* 54(3), Part 2, 599–611 (2008)
11. The EFIPSANS project, <http://www.efipsans.org/>



# An Extensible Monitoring and Adaptation Framework\*

Razvan Popescu, Athanasios Staikopoulos, and Siobhán Clarke

Distributed Systems Group, Trinity College Dublin, Ireland

Firstname.Lastname@scss.tcd.ie

**Abstract.** Several techniques have been defined for the monitoring and adaptation of applications. However, such techniques usually work in isolation and cannot be easily integrated to tackle complex monitoring and adaptation scenarios. Furthermore, applications may have special requirements which make it difficult to reuse such off-the-shelf approaches. In particular, these requirements may cross several application layers such as – the organisation of stakeholder roles, coordination of planned activities, and integration with third-party services.

In this paper we outline a lightweight, loosely-coupled and extensible monitoring and adaptation framework that allows application developers to integrate monitoring and adaptation techniques as units that can be linked to solve complex requirements and achieve cross-layer adaptation. In order to cater for application-tailored adaptation units, we propose a pattern-based technique for the development and integration of adaptation units.

## 1 Introduction

Complex service-oriented applications are generally heterogeneous, loosely-coupled, long-lived and continuously running and have to cope with frequent changes to their requirements and environment. In order to address such changes, applications need to be inherently flexible and adaptive and supported by appropriate adaptation infrastructures. The adaptation process serves, for example, to ensure that the application is fault tolerant or compatible with new clients. The adaptation can be enforced either at design, or at run-time, and it can be triggered by the human designer or operator of the application, or by a monitoring process.

Several techniques have been defined for the monitoring and adaptation of applications. They generally tackle issues such as interface [5], behavioural [3], quality-of service [7], service-level agreement [8], or policy mismatches [6]. However, such techniques usually work in isolation and cannot be easily integrated to tackle complex monitoring and adaptation scenarios [10]. Furthermore, applications may have special requirements which make it difficult to reuse such

---

\* This work has been supported by the EU ALIVE project [2].

off-the-shelf approaches. In particular, these requirements may cross several application layers such as – the organisation of stakeholder roles, coordination of planned activities, and integration with third-party services [2].

In this paper we outline a lightweight, loosely-coupled and extensible monitoring and adaptation framework that allows application developers to integrate monitoring and adaptation techniques as *units* that can be linked to solve complex requirements and achieve cross-layer adaptation. In order to cater for application-tailored adaptation units, we propose a pattern-based technique for the development and integration of adaptation units.

In a nutshell, the framework assumes an *event bus* through which monitoring and adaptation units can communicate following a publish-subscribe mechanism. Human operators and monitoring units trigger events under certain conditions (e.g., a repeated service invocation failure may eventually trigger the replacement of the respective service). These events lead to the execution of matching adaptation units. In order to cope with complex adaptation scenarios, the framework allows developers and integrators to construct complex adaptation units either as compositions of existing units, or by linking them through events. We employ workflows for the high-level definition of adaptation-unit behaviour, and we name such workflows *adaptation patterns*. Using these patterns, developers can either integrate existing adaptation techniques, or develop new ones. We propose the use of the Yet Another Workflow Language (YAWL [12]) as language to express adaptation patterns. An important advantage of using YAWL as high-level language is the possibility to define workflows that integrate adaptation tasks that map onto existing adaptation techniques exposed as (WSDL [13]) Web services.

Throughout this paper we shall illustrate the applicability of the framework in the context of the ALIVE project [2]. ALIVE proposes an advanced model-driven engineering framework for the disciplined and systematic development, deployment and management of service-oriented applications based on organisation and coordination mechanisms often seen in human and other societies. ALIVE distinguishes three conceptual levels of design: the *organisation level* (viz., application stakeholder roles and their relationships), the *coordination level* (viz., high-level workflows that coordinate planned activities needed to fulfil application goals) and the *service level* (viz., supporting services that enact workflow tasks). The designs are hierarchical, with highly dependent layers, where concepts in one level refer to or reuse concepts defined in another.

The remaining of this paper is structured as follows. Section 2 presents an ALIVE use case that motivates the need for an extensible, cross-layer monitoring adaptation framework. Section 3 briefly describes the proposed framework. Section 4 illustrates how the framework supports the adaptation of the use case in section 4. Finally, section 5 presents some concluding remarks and future directions of work.

## 2 Crisis Management Use Case: Problem

Figure 1 presents a generic emergency-handling system [2]. The organisation level describes the roles involved in the scenario and their relationships. The main goal

(viz., objective) of the EMERGENCY CENTRE is to rescue, assist and transport wounded to hospitals. The emergency centre fulfils this goal by dividing it into several sub-goals and delegating them to the other roles. The emergency centre receives the emergency call and acts as an orchestrator for the other roles. It delegates the rescue operations of the trapped people to the FIRE STATION role, the assistance and transport of wounded to the FIRST-AID STATION role and the traffic management operations to the POLICE STATION role.

The role of the coordination level is to describe the workflows needed to fulfil the application goals. In ALIVE [2], human and software agents fulfil organisational roles and are in charge of executing workflow tasks either manually, or by mapping them onto (Web) services. The coordination level in Figure 1 shows a possible workflow needed to fulfil the goal of assisting the wounded<sup>1</sup>. The workflow starts by assessing the magnitude of the incident. This task is performed through the invocation of a corresponding (Web) service. The workflow continues by computing a route to the location of the incident (again done through the invocation of a (Web) service) and then by instructing the medical team to proceed to the location of the incident. Next, the medical team assists the wounded. The workflow then computes a route to the nearest hospital and instructs the medical team to transport the wounded. Finally, the workflow instructs the medical team to fill in a report which is to be processed through the invocation of a (Web) service. The goal of the first-aid station – the assistance and transport of wounded – can be split into two sub-goals that can be mapped onto the successful execution of particular workflow tasks, that is, *Assist Wounded* and *Proceed to Hospital*, respectively.

In this scenario we employ the workflow in Figure 1 to rescue people affected by a hurricane. We assume that the execution of the workflow proceeds as planned up the transportation of wounded to hospitals. Due to a major flood, the ambulances cannot follow the planned route to the hospital and an alternative route would force them to lose precious time. The medical team informs the emergency centre of this issue and the emergency centre operator decides to trigger an alternative plan that involves transporting the wounded using helicopters from a nearby military base.

This issue requires reorganisation and adaptation at various levels of the application. On the one hand, the organisation level needs to introduce a new role – MILITARY BASE – whose goal is to transport the wounded to hospitals. On the other hand, the workflow at the coordination level has to reflect the changes needed to achieve this goal using the newly added role. For example, the new workflow has to find a route to the helicopter base first. These changes can be accommodated through two adaptation techniques – one that resolves organisational issues (e.g., based on [9]), and another one that resolves workflow mismatches (e.g., along the lines of [3]).

In Section 4 we shall describe how the monitoring and adaptation framework copes with such triggers and facilitates dynamic changes at various levels of the application.

---

<sup>1</sup> We have used YAWL [12] to graphically represent the application workflow.

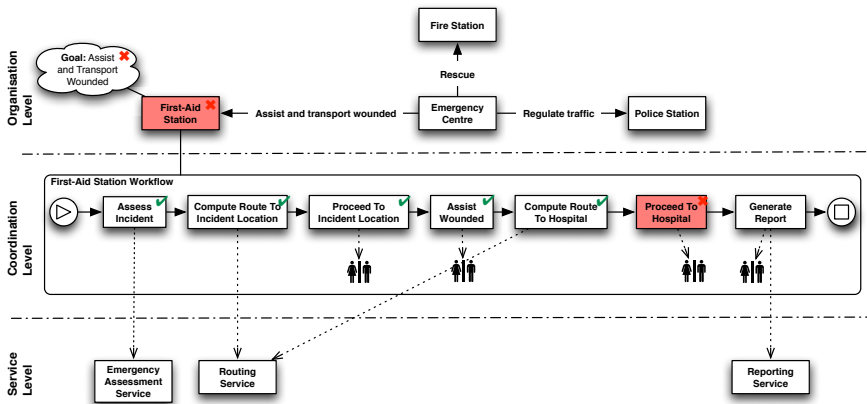


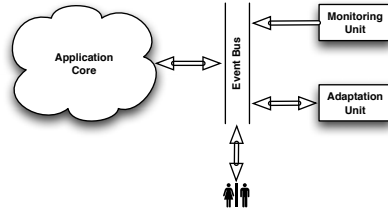
Fig. 1. ALIVE Architecture of a Crisis Management Use Case

### 3 Overview of the Monitoring and Adaptation Framework

Figure 2 presents the high-level architecture of the proposed monitoring and adaptation framework. The architecture employs an event bus (e.g., an Enterprise Service Bus [4]) to separate the core of the application (viz., the computational aspects of the application) from monitoring and adaptation units. The event bus follows a *publish-subscribe* mechanism. Monitoring and adaptation units and human application operators and designers register themselves as publishers of events. Then, other adaptation units and human application operators and designers register themselves as consumers of events by subscribing to the events of their interest.

This architecture brings the following main benefits:

- **Lightweight application architecture.** Applications can be configured to use only the monitoring and adaptation units they need. For example, a video-on-demand application may need a monitoring unit that produces *<Low Download Request>* and *<High Download Request>* events together with an adaptation unit that, upon the reception of such events modifies the upload/download bandwidth of the application.
- **Loosely-coupled monitoring and adaptation.** The event bus act as mediator between application and monitoring and adaptation units, which can be added, removed, or updated without breaking down the interoperability with the application. Furthermore, the proposed architecture allows cross-layer adaption to be performed by linking adaptation units through events (e.g., please see subsection 3.2 and section 4 for details).
- **Extensible monitoring and adaptation framework.** As applications evolve, new monitoring and adaptation units may be deployed so as to cope with changing application requirements. New and existing monitoring and



**Fig. 2.** Architecture of the monitoring and adaptation framework

adaptation techniques can be integrated into the proposed architecture by wrapping them as event producers and/or consumers.

As previously mentioned, we shall illustrate the applicability of the monitoring and adaptation framework in the context of the ALIVE project [2]. The architecture of an ALIVE application consists of three levels: organisation, coordination, and service, as shown for example in Figure 1. Recall that the organisation level describes the application stakeholder roles and their relationships. The coordination level describes high-level workflows needed to fulfil application goals. Finally, the service level describes supporting services that enact workflow tasks. In the next two subsections we present some example ALIVE events and adaptation patterns.

### 3.1 ALIVE Events

Table 1 presents some of the generic events that can be employed in the process of monitoring and adaptation of ALIVE applications. Monitoring and adaptation units can be configured to trigger adaptation events at different application levels given certain conditions are met. For example, a service level monitoring unit may trigger a *<Service Failure>* event when it notices that a service invocation has failed repeatedly. Or, an adaptation unit that tackles the adaptation of the application’s workflow may trigger a *<Match and Replace Service>* event when new workflow tasks need to be enacted by services, and a *<Critical Failure>* event if the workflow cannot be adapted to meet an application goal.

### 3.2 ALIVE Adaptation Patterns

Adaptation units can subscribe to and receive notifications of triggered events (e.g., such as the ones presented in subsection 3.1), as well as can trigger events. Adaptation patterns serve to define the high-level workflow of adaptation units. Such workflows are useful to create complex adaptation patterns from simple ones. For example, an adaptation pattern that is in charge of generating an ALIVE application workflow so as to meet one of its goals might enclose another adaptation pattern in charge of selecting and adapting services needed to enact the workflow tasks.

**Table 1.** Examples of ALIVE events

Application Level	Triggers	Events
Service Level.	Service invocation fails repeatedly, Located better service, ...	Service Failure, Match and Replace Service, Adapt Service Interface, Adapt Service Behaviour, Warning, Critical Failure, ...
Coordination Level.	Task execution has failed, ...	Task Failure, Replace Task, Generate Workflow and Replace, ...
Organisation Level.	Objective cannot be achieved, ...	Objective Failure, Adapt Organisational Structure, ...
Designer/Operator.	Requirements have changed, Environment has changed, ...	Any of the above.

As previously mentioned, we argue for the use of YAWL [12] as language to define the workflow of the adaptation patterns. YAWL is a formal language that builds on Petri nets. YAWL conditions and tasks are similar to Petri net places and transitions, respectively. A YAWL workflow specification consists of one or more extended workflow nets (or EWF-nets for short) arranged in a tree-like structure. An EWF-net is a graph where nodes are tasks or conditions, and arrows define the control-flow relation. Each EWF-net has a single input condition and a single output condition. A YAWL task may be either atomic (e.g., *Match Role to Objective* in Figure 4) or composite (e.g., *Adapt Organisational Structure* in Figure 4). An atomic task corresponds to a leaf of the tree. A composite task corresponds to an EWF-net at a lower level in the hierarchy. The EWF-net without any composite tasks referring to it is called top-level workflow and it corresponds to the root of the tree-like hierarchy. Tasks employ one join and one split construct. A join or split control construct may be one of the following: AND, OR, XOR, or EMPTY. For example, *Is Task Critical?* and *Generate Report* in Figure 3 have an XOR split and join, respectively. Tasks can be enacted either by human operators or designers (e.g., *Is Task Critical?* in Figure 3), or by (WSDL) Web services (e.g., *Generate Report* in Figure 3<sup>2</sup>).

Hereafter we present three adaptation patterns that may be used to tackle the issues raised by the use case described in section 2.

Figure 3 presents the YAWL workflow of a possible adaptation pattern that tackles a task failure. The workflow first verifies whether the task that failed (e.g., the *Proceed to Hospital* task in Figure 1) is needed to meet the objective

<sup>2</sup> Although being enacted by services, some tasks may also require inputs from the human operators or designers as well, such as *Generate Report* in Figure 3. Furthermore, note that we did not represent the actors enacting composite tasks since these tasks are usually composed of several tasks, each enacted by a different service or human operator or designer

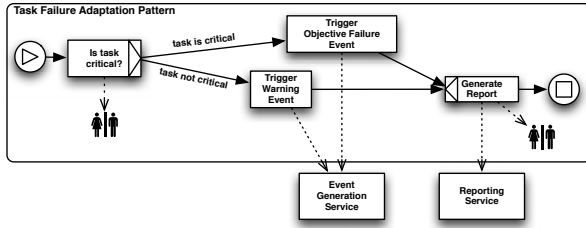


Fig. 3. TASK FAILURE adaptation pattern

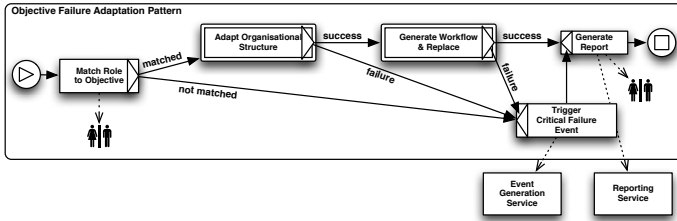


Fig. 4. OBJECTIVE FAILURE adaptation pattern

of the application workflow (viz., FIRST-AID STATION workflow in Figure 11<sup>3</sup>). If the task is critical, the workflow triggers an *Objective Failure* event, otherwise it triggers a *Warning* event. Finally, the workflow generates a report based on its execution path.

Figure 4 presents the YAWL workflow of a possible adaptation pattern that tackles an objective failure (e.g., in response to an *Objective Failure* event triggered by the TASK FAILURE adaptation pattern in Figure 3). The workflow first matches a (possibly new) organisational role that can fulfil the respective objective. Given one exists, the workflow adapts the organisational structure by adding the matched role and by taking into account its relationship with the other roles in the organisational structure, as well as the objective to be fulfilled<sup>4</sup>. A successful adaptation of the organisational structure leads to the definition of a new application workflow that makes use of the added role to meet the objective (see Figure 5). If the generation is successful, the workflow terminates with the generation of a report.

Figure 5 presents the YAWL workflow of a possible adaptation pattern that can be used to generate an application workflow to meet a given objective. The workflow first employs planning and workflow adaptation techniques to generate

<sup>3</sup> Please note that we employ YAWL workflows to define both – application workflows and adaptation pattern workflows.

<sup>4</sup> Please note that space limitations do not allow us to describe the content of all composite tasks of the workflows involved in the use case.

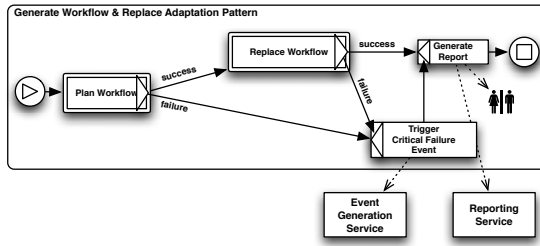


Fig. 5. GENERATE WORKFLOW & REPLACE adaptation pattern

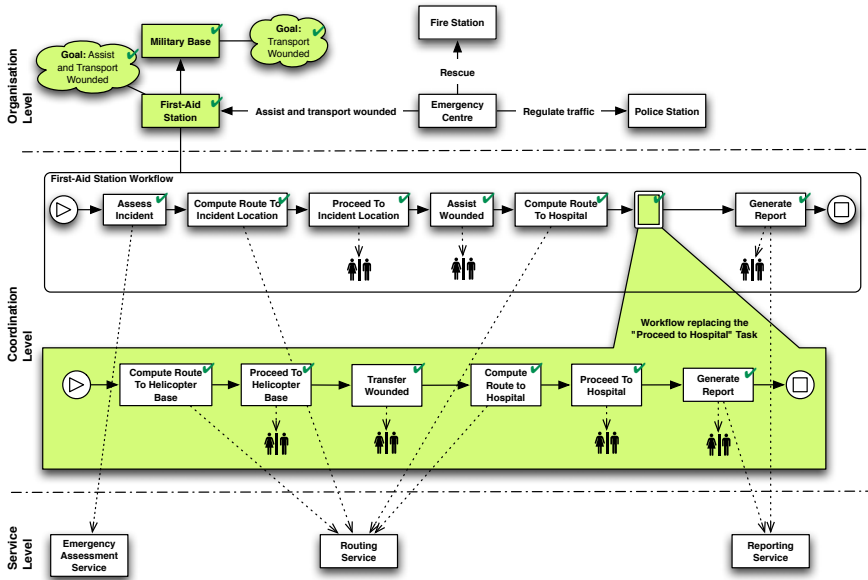


Fig. 6. Adapted ALIVE Architecture of the Crisis Management Scenario

a workflow that can meet the objective. Next, it replaces a given workflow (e.g., the workflow task *Proceed to Hospital* in Figure 1) with the newly generated workflow. Finally, it generates a report.

In the next section we describe how these three adaptation patterns can be employed to tackle the complex adaptation issues raised by the use case described in section 2.

## 4 Crisis Management Use Case: Solution

For this scenario we assume that the workflow task *Proceed to Hospital* (see the workflow in Figure 1) fails due to flooding.



The monitoring and adaptation framework first triggers a <Task Failure> event which leads to the execution of the TASK FAILURE adaptation pattern, as shown in Figure 3. Since the *Proceed to Hospital* task is critical for the fulfilment of the workflow objective, the adaptation pattern triggers an <Objective Failure> event. This event leads to the execution of the OBJECTIVE FAILURE adaptation pattern. The human operator is in charge of finding a (possibly) alternative organisational role that can fulfil the failed objective. For this scenario we assume that the operator chooses the MILITARY BASE role. The OBJECTIVE FAILURE adaptation pattern continues then with the modification of the organisational structure so as to accommodate the newly matched role. For this process we may assume that the *Adapt Organisational Structure* task of the OBJECTIVE FAILURE adaptation pattern makes use of organisational adaptation techniques based on [9]. Figure 6 illustrates the new organisational structure, in which the FIRST-AID STATION role delegates the transport of the wounded to the MILITARY BASE role.

Next, the adaptation pattern tries to generate a new workflow that takes into account the new organisational structure to meet the (sub-)objective of transporting the wounded to a hospital. This is achieved through the execution of the GENERATE WORKFLOW & REPLACE adaptation pattern, depicted in Figure 5. This pattern first employs human expertise and planning techniques to generate a workflow that can meet the given objective while following the organisational constraints. We may assume that the planning step makes use of workflow adaptation techniques along the lines of [3]. Next, the pattern replaces the failed task with a composite task that encloses the newly created workflow. This can be achieved using YAWL worklets [1].

For this scenario we assume that the planning step produces the workflow in Figure 6. The replacement workflow first routes the medical team to the military base. The medical team then transfers the wounded onto military helicopters. Finally, the workflow computes a route to the nearest hospital and it instructs the medical and military teams to proceed to the respective location.

## 5 Concluding Remarks

In this paper we have outlined a lightweight, loosely-coupled and extensible monitoring and adaptation framework that allows application developers to integrate monitoring and adaptation techniques as units that can be linked to solve complex requirements and achieve cross-layer adaptation. This is supported by an event bus through which monitoring and adaptation units can communicate following a publish-subscribe mechanism.

Developers may either wrap existing approaches as monitoring and adaptation units, or they may develop new ones. In order to assist developers during this process we have proposed the use of YAWL as language to define the high-level workflow of monitoring and adaptation units. Using YAWL, developers have the possibility to define complex unit workflows as compositions of simple ones, or to integrate existing monitoring and adaptation approaches as (WSDL) Web

services enacting workflow tasks. The workflows can be used as patterns and customised into application-tailored monitoring and adaptation units.

As future work, we mainly plan to investigate:

- The development of a monitoring and adaptation framework based on the outline given in this paper,
- The definition of a set of (common,) generic adaptation patterns following the adaptation taxonomy in [11], and
- The use of model-driven engineering techniques for the disciplined and systematic engineering of tools that support the development and deployment of application-tailored adaptation units from generic patterns.

## References

1. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 291–308. Springer, Heidelberg (2006)
2. Alive. D2.2a: Theoretical Framework (2009), [http://www.ist-alive.eu/index.php?option=com\\_docman&task=doc\\_download&gid=3&Itemid=49](http://www.ist-alive.eu/index.php?option=com_docman&task=doc_download&gid=3&Itemid=49)
3. Brogi, A., Popescu, R.: Automated Generation of BPEL Adapters. In: Dan, A., Lamersdorf, W. (eds.) ICSSOC 2006. LNCS, vol. 4294, pp. 27–39. Springer, Heidelberg (2006)
4. Chappell, D.: Enterprise Service Bus. O'Reilly Media, Sebastopol (2004) ISBN 978-0596006754
5. Dumas, M., Spork, M., Wang, K.: Adapt or Perish: Algebra and Visual Notation for Service Interface Adaptation. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 65–80. Springer, Heidelberg (2006)
6. Erradi, A., Maheshwari, P., Padmanabhuni, S.: Towards a Policy-Driven Framework for Adaptive Web Services Composition. In: NWESP 2005: Proceedings of the International Conference on Next Generation Web Services Practices, pp. 33–38. IEEE Computer Society, Los Alamitos (2005)
7. Harney, J., Doshi, P.: Speeding up Adaptation of Web Service Compositions Using Expiration Times. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 1023–1032. ACM, New York (2007)
8. Narendra, N.C., Ponnalagu, K., Krishnamurthy, J., Ramkumar, R.: Run-Time Adaptation of Non-functional Properties of Composite Web Services Using Aspect-Oriented Programming. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSSOC 2007. LNCS, vol. 4749, pp. 546–557. Springer, Heidelberg (2007)
9. Penserini, L., Aldewereld, H., Dignum, F., Dignum, V.: Adaptivity within an Organizational Development Framework. In: SASO 2008: Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 477–478. IEEE Computer Society, Los Alamitos (2008)
10. S-Cube. PO-JRA-1.2.1: State of the Art Report, Gap Analysis of Knowledge on Principles, Techniques and Methodologies for Monitoring and Adaptation of SBAs (2008), [http://www.s-cube-network.eu/results/deliverables/wp-jra-1.2/PO-JRA-1.2.1-State-of-the-Art-report-on-principles-techniques-and-methodologies-for-monitoring-and-adaptation.pdf/at\\_download/file](http://www.s-cube-network.eu/results/deliverables/wp-jra-1.2/PO-JRA-1.2.1-State-of-the-Art-report-on-principles-techniques-and-methodologies-for-monitoring-and-adaptation.pdf/at_download/file)

11. S-Cube. CD-JRA-1.2.2: Taxonomy of Adaptation Principles and Mechanisms (2009),  
[http://www.s-cube-network.eu/results/deliverables/wp-jra-1.2/CD-JRA-1.2.2\\_Taxonomy\\_ofAdaptation\\_Principles\\_and\\_Mechanisms.pdf/at\\_download/file](http://www.s-cube-network.eu/results/deliverables/wp-jra-1.2/CD-JRA-1.2.2_Taxonomy_ofAdaptation_Principles_and_Mechanisms.pdf/at_download/file)
12. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. *Inf. Syst.* 30(4), 245–275 (2005)
13. WSDL. Web Service Description Language v1.1 (2001),  
<http://www.w3.org/TR/wsdl>

# Cross-Layer Adaptation and Monitoring of Service-Based Applications\*

Raman Kazhamiakin, Marco Pistore, and Asli Zengin

Fondazione Bruno Kessler – IRST, Trento, Italy  
{raman,pistore,zengin}@fbk.eu

**Abstract.** The heterogeneity and dynamicity of services, their underlying infrastructures make the problem of adaptation and monitoring an emerging issue for service-based applications (SBA). While various approaches aim to address these problems, most of them focus on a particular element of the SBA architecture. Indeed, those approaches are fragmented and isolated; they do not consider the effect of adaptations on the whole stack of the functional layers of SBA. In this paper we study the problem of cross-layer SBA monitoring and adaptation on a series of case studies and define the requirements for the integrated approaches that provide coherent solutions to monitor and adapt the whole application. Finally we propose the mechanisms and principles that are necessary for addressing the requirements and enabling an integrated cross-layer framework.

## 1 Introduction

Service-Based Applications run in dynamic business environments and address evolving requirements. These applications should hence become drastically flexible, as they should be able to adequately identify and react to various changes. These challenges make monitoring and adaptation the key elements of modern SBA functionality.

The problem of monitoring and adaptation of various software systems has gained a lot of interest both in research community and in industry. In recent years, these aspects have attracted more and more interest in the area of SBA and in Service-Oriented Computing (SOC). However, the results and directions are still insufficient. A fundamental problem with the state-of-the-art results on monitoring and adaptation is their fragmentation and isolation; they target a particular functional layer. While these solutions are quite effective when considered in isolation, they may be incompatible or even harmful when the whole SBA is considered. Indeed, realization of different SBA layers may be highly interleaved: various artifacts at one layer may refer to the same artifacts at another, while such relations are ignored by the isolated monitoring and adaptation solutions. As a consequence, wrong problems may be detected, incorrect decisions may be made, and modifications at one layer may damage the functionality of another.

This paper aims to study the problem of cross-layer SBA monitoring and adaptation. Starting from a series of illustrative scenarios, we identify and classify the requirements that the novel cross-layer approaches should address. In particular, we identify the four

---

\* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

key problems that may arise due to the isolated monitoring and adaptation, namely lack of alignment of monitored events, lack of adaptation effectiveness, lack of compatibility, and integrity of the adaptation activities. Based on the taxonomy of those requirements, we also define *the mechanisms and principles* that are *necessary for addressing the requirements* and that facilitate an integrated cross-layer framework.

## 2 Adaptable and Monitorable Service-Based Applications

An SBA is an application that cannot be implemented by a singular service, but requires the aggregation of multiple singular or composite services in a network to guarantee the application goals [1]. Services are independent entities that can achieve a specific task or a group of tasks and be re-utilized by different applications in different contexts.

### 2.1 Functional Layers of Service-Based Applications

An SBA can be illustrated by its three functional layers, business process management (BPM), service composition and coordination (SCC) and service infrastructure (SI) [2]. At the BPM layer the application activities, constraints and requirements are described without design details. The basic workflow constructed at BPM is refined at the SCC layer by the composition of suitable services. Finally, SI provides the underlying runtime environment. Below, we will introduce the key elements of each layer that we have identified as critical for the cross-layer monitoring and adaptation.

**BPM Layer:** Workflow, key performance indicator (KPI) and agile service network (ASN) are the key elements relevant for BPM. *Workflow* is the abstract model of the business process defining logical decision points, sequential or parallel work routes and exceptional cases. While business activities constitute the workflow, business rules together with business policies have an effect on the specification of business processes. *KPI* is a metric that shows quantitatively if business performance meets the pre-defined business goals. *ASN* is a model used to illustrate cross-organizational interactions among companies, which collaborate to construct distributed complex SBAs. *ASN* depicts the most abstract level where partners are nodes and their offerings and revenues are edges on model.

**SCC Layer:** Service composition, process performance metric (PPM) and service metrics are the key elements relevant for SCC. *Service composition* is a combination of services to realize a workflow. The designer needs to know descriptions, interfaces and supported protocols of available services for composition. *PPM* measures performance of a process or its parts in terms of cost, quality or duration. *Service metrics* are basically QoS metrics which talk about non-functional properties of services.

**SI Layer:** Service registry, discovery and selection mechanisms constitute infrastructural facilities to find and select the services required by composition. *Service registry* is the information system where service descriptions are kept as a searchable repository. *Service discovery and selection* are the basic functionalities that serve SCC for the selection of most suitable services for SBA realization. *Service realization* corresponds to the run-time environment on top of where services are executed (e.g. grids, clusters, data servers, software, protocols and network infrastructure).

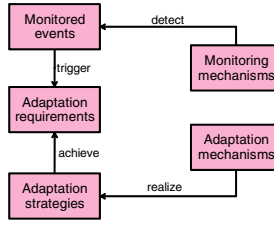


Fig. 1. Conceptual A&M framework

## 2.2 Monitoring and Adaptation

A general framework for SBA monitoring and adaptation [2] is represented in Figure 1. *Monitoring Mechanism* is **any** means to check whether the actual situation corresponds to the expected one. The meaning is very broad; it may include run-time verification and testing, post-mortem analysis, data mining, etc. With these mechanisms one may detect *Monitored Events*, i.e., the events that deliver the relevant information about the application execution, evolution, and context. In turn, events trigger *Adaptation Requirements*, which represent the necessity to change the underlying SBA in order to remove the difference between the actual (or predicted) situation and the expected one. In order to meet requirements *Adaptation Strategies* are defined, which in turn are realized with the appropriate *Adaptation Mechanisms* provided by the underlying SBA.

A wide range of the adaptation and monitoring approaches at different functional SBA layers can be mapped onto this framework. Such approaches should then be used by the integrated cross-layer adaptation and monitoring framework as building blocks. Note, however, that currently not for all of the presented elements appropriate mechanisms have already been defined. New approaches still have to be provided by the research in order to enable building blocks of cross-layer monitoring and adaptation.

**Monitoring in Different SBA Layers:** At the BPM layer the monitoring approaches focus on monitoring business activities (BAM, [34]) and KPIs [56]. At SCC layer the monitoring engines and frameworks provide means to observe the execution of composed services (e.g., in BPEL), including functional and non-functional properties [78], or the constituent services [9]. Monitoring at SI layer may be realized on top of Grid monitoring solutions or based on middleware for component-based systems [10].

While the existing approaches may cover a wide spectrum of the problems and application domains, they are usually considered in isolation from each other; exploit different models of monitored events, and different platforms. As we will see in the following sections, this may lead to inconsistencies in functioning of the SBA.

**Adaptation in Different SBA Layers:** At BPM, adaptation strategies cover business process, KPI and ASN adaptation [1112]. Mostly ad-hoc modifications are applied as an adaptation mechanism. At SCC, adaptation approaches focus on either the composition [1314] or the services [1516]. Automated composition, model-driven transformation and dynamic service binding are common adaptation mechanisms for this layer. Finally, adaptation at the SI may be service-discovery or service-realization-related [1718]. To achieve SI adaptation, various self-\* techniques can be used.

**Table 1.** Monitoring at different functional SBA layers

Layer		Subject	Events	Mechanisms
BPM	Business Process monitoring	business process model; transaction protocol; data and control flow	violation of correctness properties of instance executions; correspondence to the model; violation of transactional property	BAM/BI; process log analysis, process mining
	KPI monitoring	KPI	KPI violations	BAM/BI, special approaches
SCC	Monitoring functional properties	service composition; data and control flow	violation of functional properties of a composition; violation of functional properties of a constituent services	special purpose monitoring engines based on temporal logics; rule languages; calculi
	Monitoring non-functional properties	PPMs, utility functions, QoS metrics	violations of expected values/thresholds, SLA violations	special purpose monitoring engines, service monitors
SI	Grid monitoring	grid infrastructure (sites, virtual organizations); grid applications	wide range of infrastructural and application events	grid monitoring platforms and architectures
	Monitoring of component-based systems	components (state, bindings, messages, internal data), component platform (performance, dependability, state of resources)	component- and middleware-related events	middleware monitoring mechanisms, internal component monitoring mechanisms

**Table 2.** Adaptation at different functional SBA layers

Layer		Requirements	Strategies/actions	Mechanisms
BPM	Business process adaptation	optimize process; recover from unforeseen execution; customize process	modify business process control flow (add, delete, replace process tasks and process fragments), modify business process data flow (change data dependencies, values); process re-design	ad-hoc modifications (performed by business analysts); evolution
	KPI adaptation	adjust to changed business goals, business context, ASN elements	add or remove KPI, change KPI values	ad-hoc modification; negotiation; evolution
	ASN adaptation	optimize costs; transactionality; accommodate to ASN changes	change transaction protocol; change service; re-negotiate for an offering	ad-hoc modification, negotiation, evolution
SCC	Composition-related adaptation	adjust to changed process model or KPI, optimize process, recovery	re-composition; control/data flow changes; PPM changes	automated composition; model-driven transformations; fragmentation
	Service-related adaptation	service changes; optimization; SLA violation	replace a service; re-execute a service; re-negotiate QoS	dynamic binding; negotiation
SI	Service discovery-related adaptation	optimization, adjust to business requirements	change registry; update registry (new services, new descriptions); change discovery mechanism; change selection mechanism	platform-specific; reputation management
	Service realization-related adaptation	optimization, adjust to infrastructural failures	modify/re-configure service platform (software, OS, virtual machines, physical platforms); modify/re-configure service resources (allocate/release resources, load balancing); adapt resource management (change resource broker, re-configure/re-execute grid application)	ad-hoc changes; self-* techniques

Although the state-of-the-art approaches may address a variety of SBA adaptation needs, none of them are complete. They focus on a local solution for a specific adaptation requirement without taking into account its dependencies or effects on different SBA layers. As illustrated in next section, such ad-hoc approaches are not promising in terms of addressing a proper solution for the whole SBA.

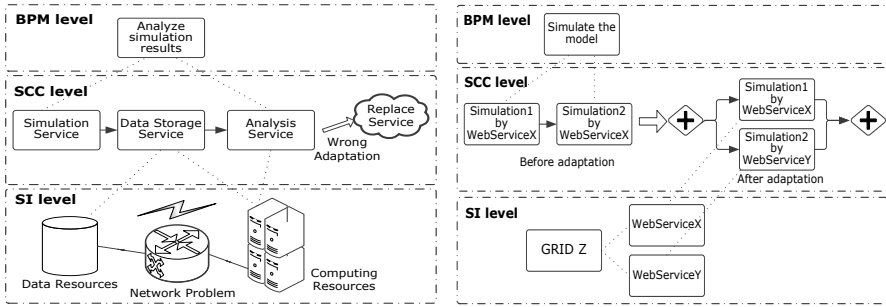


Fig. 2. Cross-layer requirements: lack of alignment and effectiveness

### 3 Requirements for Cross-Layer SBA Adaptation and Monitoring

In this section we illustrate a range of problems occurring due to the isolation of adaptation and monitoring at different SBA layers. We use scenarios from the automotive domain case study. For each problem we demonstrate the model of the SBA used, the scenario leading to the problem, and the requirements that the cross-layer adaptation and monitoring framework should meet in order to resolve that problem.

#### 3.1 Lack of Alignment of Monitored Events

If the monitoring is performed by specific mechanisms provided at different layers in isolation, then the events are not aligned and the critical information is not propagated across layers. This may lead to the wrong identification of the source of problem.

*Model:* At BPM layer the relevant part of the SBA is the “manufacturing” business process. The goal is to design and simulate new models and move to mass production after simulation verification. Activities are realized by the appropriate services. Underlying infrastructure comprises computational resources.

*Scenario:* The “analyze simulation results” activity takes abnormally much time, the composition PPM is violated. In order to compensate the PPM violation, the analysis service is substituted with another service (Figure 2, left part).

*Problem:* The expected effect is not achieved: the problem is not the service, but the network problem during the transfer of a large amount of data. The real problem occurs at SI, while detected at SCC, so it is incorrectly diagnosed. A solution here would be to monitor it also at SI and perform adaptation there (e.g., by allocating network resources). Note that monitoring the same problem at different layers is not enough: the events still have to be aligned to avoid conflicting adaptations executed in isolation.

*Requirements:* The problems above require the following: (i) providing monitoring mechanisms at all the layers, where the problem of interest can be observed (ii) providing means to propagate the monitored information across the layers to properly diagnose the actual problem (iii) aligning and correlating the events across layers to avoid spontaneous adaptations at different layers.



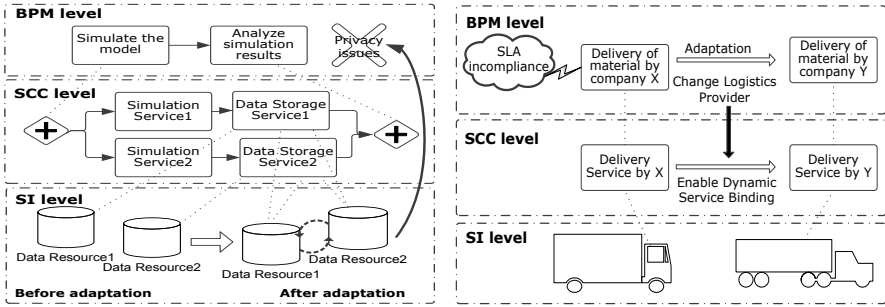


Fig. 3. Cross-layer requirements: lack of compatibility and integrity

### 3.2 Lack of Adaptation Effectiveness

Another possible problem is that the adaptation activities may fail to achieve the expected effect, since they do not take into account the properties of other layers.

*Model:* At the BPM layer the relevant part of the SBA is the “manufacturing” business process. At SCC business activities are realized by services provided by ASN partners. E.g. “simulate the model” business activity can be realized by a service that accepts the simulation requests and runs them on top of a HPC grid. SI layer provides the run-time environment, i.e., the grid computing resources where simulation tests are run, and the storage resources where the test data and results are kept.

*Scenario:* At SCC layer, the average duration of simulations is not met because the simulation runs take too much time. The adaptation requirement is to reduce the total simulation time to the required PPM. To achieve that, it is decided to parallelize simulation tests at service level by adding a new service (Figure 2 right part).

*Problem:* The new service uses the same grid resources to run the simulation tests as the original one. In this case there will not be an improvement in total performance.

*Requirements:* In order to overcome this problem, it is crucial to take into account features of the whole SBA stack when the adaptation requirements and strategy are identified. In the example this would mean to relate the requirement to parallelize the activity in SCC layer with the allocation of service resources in SI layer so that independent services are used.

### 3.3 Lack of Adaptation Compatibility

The problem of compatibility of adaptation refers to the situation, where the adaptation performed at one layer is not compatible with the constraints posed by other layers.

*Model:* As in previous scenario, the BPM layer is represented by the “manufacturing” business process, the SCC layer is realized as a composition of the appropriate services, while at the SI layer a grid infrastructure underlies the simulation services.

*Scenario:* At the SI the availability QoS value is violated for some of the storage resources. The adaptation requirement in this case is to compensate the QoS degrade. It is achieved by performing load balancing of the storage resources (Figure 3 left part).

*Problem:* The applied adaptation strategy may, however, violate privacy issues stated at the BPM layer. In particular, a special business rule may require that simulation data for each automobile model must be kept at a separate server with access control. This implies that such distribution of data is not permitted.

*Requirements:* To address this problem, it is necessary to consider how the identified adaptations influence the SBA as a whole. It is crucial to be able (i) to define certain “boundaries” for each layer and (ii) to check that adaptations do not cross those boundaries. In the above scenario it would be necessary to be able to analyze how the resource re-allocation may affect the business rules. Indeed, this implies that both elements are appropriately modeled, and the analysis is done before the adaptation is executed.

### 3.4 Lack of Adaptation Integrity

The problem of adaptation integrity is that, an adaptation at a particular layer only is not enough but several actions at different layers should be performed.

*Model:* At the BPM layer, we are interested in “plan and purchase material from suppliers” business process. The goal is to acquire the required components before manufacturing. The “decide on supplier” business activity is provided by a service that keeps information about available suppliers and selects the most appropriate. The “delivery of material” business activity has to be performed by a delivery service, probably involving some other services. The SI layer is the appropriate service execution platform.

*Scenario:* At BPM, the logistics provider, responsible for the delivery, does not comply with the SLA. The adaptation requirement is to compensate the SLA violation, and can be achieved by switching to a new logistics provider (Figure 3, right part).

*Problem:* During the adaptation action, several process instances might have already been started for some material. Change of the logistics provider will indeed affect these instances as the “delivery of material” activity has to be performed by the new one. Thus, it is also necessary to adapt at the SCC layer by changing the composition instance accordingly: it is needed to bind to the new services, align with the new interface if it is different for the new provider, etc. This may also require adaptations at the SI layer as the new service may have particular low-level protocols.

*Requirements:* To address this problem, novel mechanisms are necessary to identify and aggregate wide range of adaptation actions available at different functional layers. Indeed, these actions should be performed in a coordinated manner; some centralized mechanism should be able to control and manage isolated layer-specific tools. In our example, it should analyze which layers are “affected”, identify adaptation actions at different layers, and schedule a coordinated procedure that executes them.

## 4 Towards Cross-Layer Adaptation and Monitoring

In this section we will show how the presented requirements for cross-layer adaptation and monitoring are positioned in the general A&M framework and what kind of mechanisms and principles are necessary to achieve them. We remark that our goal here is not to come out with a concrete solution for this problem, but rather to agree on a common vision of the problem and to identify the most prominent directions.

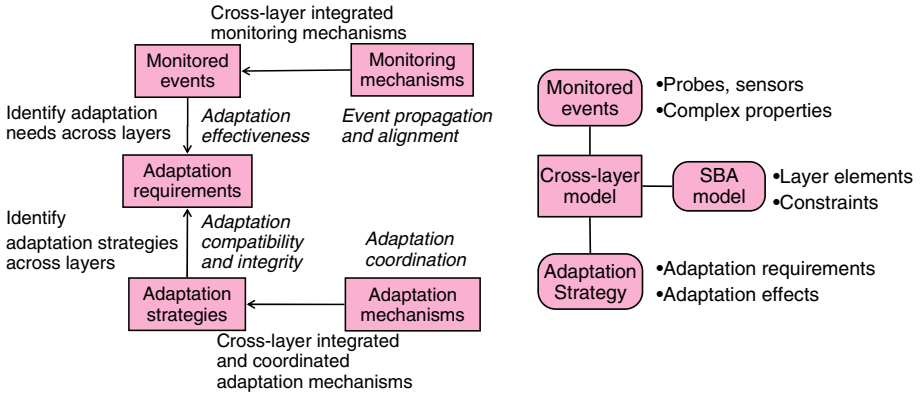


Fig. 4. Conceptual cross-layer adaptation and monitoring framework

#### 4.1 Required Mechanisms for Cross-Layer Adaptation and Monitoring

Left part of Figure 4 represents cross-layer requirements in the general A&M framework. It also shows mechanisms and means necessary to achieve requirements.

**Cross-layer integrated monitoring mechanisms.** To guarantee the requirements of alignment and propagation of monitored events across functional layers, novel mechanisms have to be provided. These mechanisms will be built on top of the monitoring capabilities of the isolated functional layers; they will provide an integrated infrastructure for the SBA monitoring. In particular, they should (i) relate events and mechanisms of different layers to each other to enable their correlation, aggregation and alignment and (ii) provide an infrastructure for subscribing, detecting, and propagating the relevant events across monitoring engines defined at different functional layers.

**Cross-layer integrated and coordinated adaptation mechanisms.** Cross-layer mechanisms should guarantee that adaptation activities being triggered at different layers are properly coordinated. In order to avoid spontaneous, concurrent and conflicting adaptations in isolation, there is a need to provide centralized mechanisms that are able to (i) aggregate and coordinate different adaptations when they appear to be related or triggered by the same events, (ii) control (e.g., activate or deactivate) adaptation mechanisms at different layers and (iii) validate different adaptation activities to check if they are in conflict, interleaved, or have interdependencies.

**Means to identify adaptation needs across layers.** Cross-layer adaptation mechanisms should be capable of properly identifying the necessary adaptation requirements, when the changes concern not a single layer, but the whole application. That is, these mechanisms would address the problem of adaptation effectiveness by providing ways to (i) identify the actual problem and the corresponding adaptation requirements and (ii) map these requirements onto the relevant functional layers.

**Means to identify adaptation strategies across layers.** Finally, the mechanisms should ensure adaptation compatibility and integrity. They should support the proper identification and selection of the adaptation strategies. In particular, novel approaches should (i) validate the adaptation strategies against the whole SBA model to avoid

actions that have undesired side-effects with respect to other layers (compatibility problem) and (ii) foresee whether the adaptation strategies are sufficient to achieve the corresponding requirements or some other actions are needed (integrity problem).

## 4.2 Required Principles for Cross-Layer Adaptation and Monitoring

To provide a basis for cross-layer SBA monitoring and adaptation and to enable various mechanisms described in subsection 4.1, it is necessary to explicitly relate different conceptual elements to each other and across different layers. This vision is reflected by the conceptual model represented in right part of Figure 4. It shows that various elements of SBA, of adaptation mechanisms and capabilities, and of monitoring mechanisms and capabilities should be related to a centralized, high-level cross-layer model. More precisely, the following should be defined by the cross-layer framework:

**Cross-layer representation of monitored events.** Cross-cutting aspects of the SBA should be used to capture and represent relevant monitored events, monitoring mechanisms and information sources at a particular layer; to abstract from the low-level details; to characterize the input of monitoring engines in terms of those cross-cutting aspects thus making the cross-layer event relations transparent.

**Cross-layer representation of adaptation strategies.** To enable the analysis of adaptation integrity, effectiveness, and compatibility, the adaptation strategies and mechanisms of different layers should also be characterized in terms of the relevant cross-layer models. In particular, this representation should describe key aspects of the adaptation strategies (the requirements to be achieved, the effects of the strategy enactment) abstracting away the specific information about how the adaptation strategy is represented and realized in the approach.

**Cross-layer representation of the SBA model.** In order to understand the relation and the impact of the adaptation activities on the elements of the SBA architecture, the latter should also be characterized in terms of some cross-cutting models. In this way, the specific requirements and constraints, posed by the SBA engineers on different layers, will be related to the monitored events and to the adaptations.

## 5 Conclusion

Adaptation and monitoring is a vital issue for SBA, and its functional layers BPM, SCC and SI. In this paper we identified and illustrated the problems caused by the isolation of current approaches in adaptation and monitoring. We provided a set of requirements to address these problems and proposed a uniform conceptual model that could facilitate a holistic cross-layer adaptation and monitoring framework.

In our future work we will construct an adaptation manager, where various layer-specific monitored events and adaptation mechanisms are coordinated. In order to realize such a manager we will propose a novel architecture that can address the cross-layer A&M requirements, namely alignment of monitored events, effectiveness, compatibility and integrity. The conceptual framework that we introduce in this paper will help us identify main elements of this architecture.

## References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: A research roadmap. *International Journal of Cooperative Information Systems* 17, 223–255 (2008)
2. Hielscher, J., Metzger, A., Kazhamiakin, R. (eds.): Taxonomy of Adaptation Principles and Mechanisms. S-Cube project deliverable (2009), S-Cube project deliverable: CD-JRA-1.2.2, <http://www.s-cube-network.eu/achievements-results/s-cube-deliverables>
3. Beeri, C., Eyal, A., Milo, T., Pilberg, A.: Monitoring Business Processes with Queries. In: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 603–614 (2007)
4. Roth, H., Schiefer, J., Schatten, A.: Probing and Monitoring of WSBPEL Processes with Web Services. In: CEC-EEE 2006: Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, p. 30 (2006)
5. Jeng, J.J., Schiefer, J., Chang, H.: An Agent-based Architecture for Analyzing Business Processes of Real-Time Enterprises. In: EDOC 2003: Proceedings of the 7th International Conference on Enterprise Distributed Object Computing, p. 86 (2003)
6. Castellanos, M., Casati, F., Shan, M.C., Dayal, U.: iBOM: A Platform for Intelligent Business Operation Management. In: ICDE 2005: Proceedings of the 21st International Conference on Data Engineering, pp. 1084–1095 (2005)
7. Barbon, F., Traverso, P., Pistore, M., Trainotti, M.: Run-Time Monitoring of Instances and Classes of Web Service Compositions. In: IEEE International Conference on Web Services (ICWS 2006), pp. 63–71 (2006)
8. Mahbub, K., Spanoudakis, G.: Monitoring WS-Agreements: An Event Calculus-Based Approach. In: Baresi, L., Nitto, E.D. (eds.) *Test and Analysis of Web Services*, pp. 265–306. Springer, Heidelberg (2007)
9. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *J. Network Syst. Manage.* 11 (2003)
10. Andreozzi, S., Bortoli, N.D., Fantinel, S., Ghiselli, A., Rubini, G.L., Tortone, G., Vistoli, M.C.: GridICE: a monitoring service for grid systems. *Future Generation Computer Systems* 21, 559–571 (2005)
11. Brogi, A., Popescu, R.: Automated Generation of BPEL Adapters. In: Dan, A., Lamersdorf, W. (eds.) *ICSOC 2006. LNCS, vol. 4294*, pp. 27–39. Springer, Heidelberg (2006)
12. Ly, L.T., Rinderle, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. *Data Knowl. Eng.* 64, 3–23 (2008)
13. Kongdenfha, W., Saint-paul, R., Benatallah, B., Casati, F.: An aspect-oriented framework for service adaptation. In: Dan, A., Lamersdorf, W. (eds.) *ICSOC 2006. LNCS, vol. 4294*, pp. 15–26. Springer, Heidelberg (2006)
14. Motahari Nezhad, H.R., Benatallah, B., Martens, A., Curbera, F., Casati, F.: Semi-automated adaptation of service interactions. In: *WWW 2007: Proceedings of the 16th international conference on World Wide Web*, pp. 993–1002. ACM, New York (2007)
15. Harney, J., Doshi, P.: Speeding up adaptation of web service compositions using expiration times. In: *WWW 2007: Proceedings of the 16th international conference on World Wide Web*, pp. 1023–1032 (2007)
16. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: Paws: A framework for executing adaptive web-service processes. *IEEE Softw.* 24, 39–46 (2007)
17. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. *IEEE Trans. Softw. Eng.* 33, 369–384 (2007)
18. Pernici, B., Rosati, A.M.: Automatic learning of repair strategies for web services. In: *ECOWS 2007: Proceedings of the Fifth European Conference on Web Services*, Washington, DC, USA, pp. 119–128. IEEE Computer Society, Los Alamitos (2007)

# Towards a Unified Architecture for Resilience, Survivability and Autonomic Fault-Management for Self-managing Networks

Nikolay Tcholtchev<sup>1</sup>, Monika Grajzer<sup>2</sup>, and Bruno Vidalenc<sup>3</sup>

<sup>1</sup> Fraunhofer FOKUS Institute for Open Communication Systems, Berlin, Germany

<sup>2</sup> Telcordia Poland, Poznan, Poland

<sup>3</sup> Alcatel-Lucent France, Paris, France

Nikolay.Tcholtchev@fokus.fraunhofer.de, mgrajzer@telcordia.com,  
Bruno.Vidalenc@alcatel-lucent.com

**Abstract.** The emergence of self-managing networks can be seen as an enabler for increased dependability, reliability and robustness of the network layer. All these features are significant for the services and applications relying on the network infrastructure. This paper explores the links between traditional Fault-Management functions belonging to the management plane and the fundamental network functions for Resilience and Survivability embedded inside the protocol modules of a node/device. This results in an architectural framework that allows nodes/devices to implement the converging aspects of Fault-Management (now becoming autonomic), Resilience and Survivability in a self-managing network. The components and adaptation mechanisms of the proposed framework will make the network layer more robust and application/service aware. Thus, the dependability, reliability, and adaptability of the upper layer services and applications are expected to increase.

**Keywords:** Resilience, Survivability, Autonomic Fault-Management, Risk Assessment, GANA, self-managing networks.

## 1 Introduction

The ability of a network to introduce self-healing mechanisms and actions in order to preserve both - network and service layer functionality in the presence of faulty conditions, failures and malfunctioning is of paramount importance. Introducing autonomicity into the self-healing processes enables a network to implement novel functionalities specifying network behaviour under abnormal conditions while reducing the involvement of the network operation personnel. This becomes crucial especially in extreme networking environments, e.g. after a large scale disaster damaging the network infrastructure.

Resilience is arguably considered a necessary feature of communication system functional entities which enables the self-healing of a network facing challenging conditions. A diversity of approaches in this field is implemented in different network layers, protocols and technologies. Resilient mechanisms can be found in protection and restoration schemes as in (G)MPLS and SONET/SDH, or can be intrinsically

implemented in the protocol modules, e.g. ICMP and OSPF - with regard to the automated reaction to link failures. Early initiatives, aimed at bringing the variety of methods under a common umbrella, have followed the path of creating a framework of protocols for self-healing resilient networks [2]. Works such as [3] [4] review the Resilience of routing within an Autonomous System (AS) and across multiple ASs, and capture the overall picture of the applied mechanisms, problems, and general proactive and reactive approaches. More generic views on Resilience in telecommunications and especially IP networks are provided through the concept of Multi-layer Resilience [5] [6].

The Fault-Management functions of the management plane are strongly related to the Resilience features of the network components. The evolvement of Fault-Management towards Autonomic Fault-Management [7] requires that the processes of Fault-Detection, Fault-Isolation and Fault-Removal must be automated and combined with each other. Until recently, the research in this area has mostly concentrated on techniques for Fault-Detection and Fault-Isolation which, to the authors' best knowledge, have not been merged with Fault-Removal. Consequently, in current practices the Fault-Management processes are mainly performed manually. To address this issue, architectures targeting the idea of Autonomic Fault-Management, such as Uni-FAFF [7] and Omega [8], are on the rise.

Our goal is to combine some of the research results related to Autonomic Fault-Management with the diversity of the network resilience mechanisms towards a unified architectural framework. The challenge in designing an integrated architecture for Resilience, Survivability and Autonomic Fault-Management is to accommodate the plethora of existing intrinsic multi-layer resilient mechanisms into a framework that targets the goal of overseeing the network and reacting to incidents and alarms, without interfering with the inherent self-healing mechanisms of functional entities (i.e. protocols). This is the point where the converging aspects of Resilience, Survivability and Fault-Management need to be merged. In this paper we present our work in progress on a unified architectural framework that puts across design principles and defines components from the Resilience and Autonomic Fault-Management perspective. The emerging architecture is expected to increase the dependability, reliability and robustness of the network layer and to provide better conditions to the services and applications running on top of it. We believe that such an architecture, incorporating the properties of Autonomic Fault-Management, Resilience and Survivability, is a highly beneficial and promising solution that we introduce to the domain of Autonomic Networking.

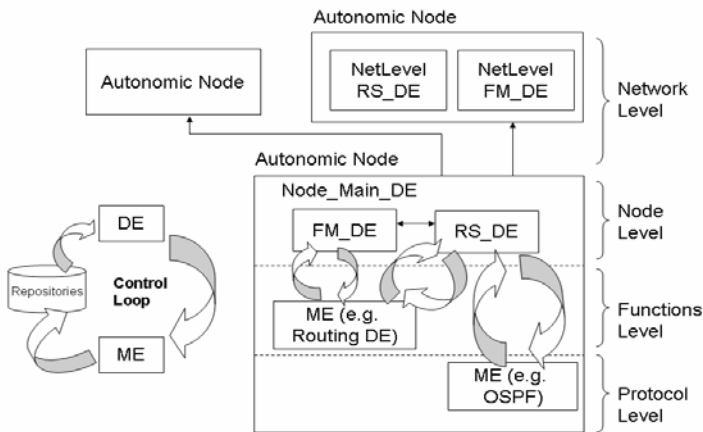
The rest of this paper is organized as follows: Section 2 presents the GANA reference model for Autonomic Networking, which we employ for the establishment of our architecture. Section 3 conducts a requirements analysis and presents the fundamental components of our framework. Section 4 presents the overall set of components belonging to the framework and elucidates the way they inter-work. Finally, section 5 concludes the paper, and gives an outlook on the future research.

## 2 The GANA Architecture in Brief

In this section, we present the *GANA (Generic Autonomic Network Architecture)* reference model [1] for autonomic network engineering, which is exploited to define

the architecture for Resilience, Survivability, and Autonomic Fault-Management in self-managing networks. In order to manage node and network functionalities, the GANA specifies a *control loop* by introducing two basic concepts – a *Managed Entity (ME)* and a *Decision Element (DE)*. A DE is conceptually an autonomic element responsible for managing a set of MEs (e.g. protocol modules), and realizes a control loop based on its continuous learning cycle. Through a specified control loop a DE is able to exercise specific algorithmic schemes and/or policies on its ME(s), while the ME(s) provide feedback to the DE by sharing information related to their current state. Information exchange can be realized over a logical link using the concept of repositories and notification/callback mechanisms.

Taking into account that control loops can be implemented at different levels of abstraction – e.g. on functions or node level, GANA provides the *Hierarchical Control Loops (HCLs)* framework consisting of four levels (Figure 1) at which DEs and associated control loops can be introduced, i.e. *Protocol Level* (e.g. a control loop inside a single protocol), *Functions Level* (in-built management of basic networking functions such as Forwarding), *Node Level* and *Network Level*. Thereby DEs, i.e. control loops, on a higher level manage DEs on a lower level (an example is given in Figure 1). For more information we refer the reader to [1].



**Fig. 1.** An overview of a GANA control loop and an example of its placement in an autonomic node/network

### 3 Components of the Targeted Architectural Framework

The diverse factors that should be taken into account while designing the unified architecture for Resilience, Survivability and Autonomic Fault-Management are listed below. We believe that this set reflects the plethora of issues, brought up in literature, related to Autonomic Fault-Management, Resilience and Survivability:

**Requirement #1:** The proposed architecture must take into account the intrinsic resilient mechanisms implemented inside the network components.



**Requirement #2:** The architecture needs to be application/service aware and thus provide facilities for implementing the concept of *Survivability Requirements (SR)*. SR will provide the possibility for applications to express their needs for obtaining vital information regarding incidents/alarms in the network and should enable the application layer to react accordingly and survive in the presence of faulty conditions.

**Requirement #3:** The proposed framework must consider the dynamic dependencies among protocols and services in the network.

**Requirement #4:** The framework must consider the causality relations (fault-propagation models) regarding root causes (faults) and their propagation as one or many errors until eventually resulting in observable symptoms (failures and alarms).

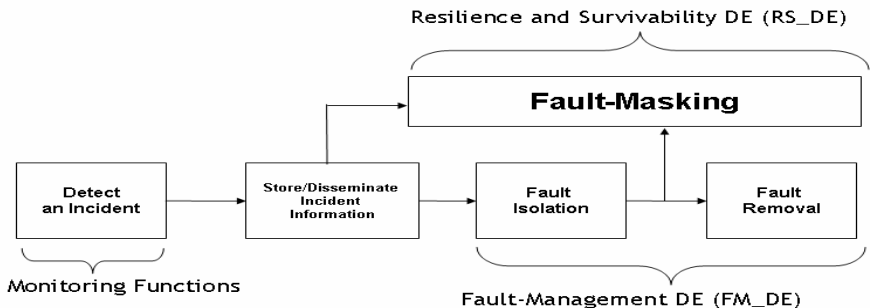
**Requirement #5:** In order to realize *Proactive Resilience*, the architectural framework should provide facilities for estimating the risk that various components and functional entities are likely to fail in the near future. Such a *Risk Assessment* can be employed to avoid the use of components which are likely to fail and to proactively sustain (e.g. by proactive rerouting) the traffic in the case of challenging conditions.

**Requirement #6:** The architecture needs to ensure that the actions undertaken by different autonomic entities do not contradict and the decision making process effectively contributes to the overall fitness of the network. Hence, a component is required that synchronizes the actions issued by diverse autonomic entities in an appropriate way.

**Requirement #7:** The effectiveness of the undertaken management actions must be verified.

**Requirement #8:** The autonomic entities of the architectural framework must be given exclusive access to all functional entities inside a node, in order to manage them with respect to incidents and alarms.

Based on these requirements and the control loop structures of GANA, a number of components and inter-working mechanisms are defined, which eventually result in a unified architecture for Resilience, Survivability, and Autonomic Fault-Management. First, we present the two major components that make the core of the proposed architecture – the *Resilience and Survivability DE (RS\_DE)* and the *Fault-Management DE (FM\_DE)*



**Fig. 2.** The sharing of responsibilities among the RS\_DE and the FM\_DE

DE (FM\_DE). The basic separation of their operational domains is illustrated in Figure 2. The RS\_DE is responsible for an immediate reaction to failures and faults aimed at sustaining an acceptable QoS level by employing diverse fault-tolerant mechanisms, i.e. it performs Fault-Masking. In parallel, the FM\_DE is responsible for automatically repairing the damaged network infrastructure by diagnosing the underlying root cause (fault) and removing it, i.e. the FM\_DE performs Fault-Isolation and Fault-Removal. Additionally, the RS\_DE implements proactive mechanisms based on the estimated fitness of the network components.

### 3.1 Fault-Management Decision Element

The autonomic entity realizing Fault-Isolation and Fault-Removal during the (long-term) operation of the network is the Fault-Management DE (FM\_DE). In a GANA autonomic node/device, the FM\_DE is required to have exclusive access to all other entities (MEs and DEs) inside the node. This requirement can be realized only if the FM\_DE belongs to the highest GANA control loop layer within a device, which is the node level. Since the node level is conceptually considered to contain only the Node\_Main\_DE, the FM\_DE is naturally specified as a sub-DE of the Node\_Main\_DE (Figure 1).

Figure 3 depicts the constellation of entities that are involved in realizing Autonomic Fault-Management inside the network devices. Autonomic Fault-Management is triggered whenever symptoms/anomalies indicating the presence of faults in the network are detected. The corresponding process of Incident-Detection is realized by specially instrumented Monitoring Components/Functions inside the autonomic nodes/devices (Figure 2) or by some other MEs or DEs (as illustrated in Figure 3) which have the capabilities to share alarm/incident information with the components of the proposed architecture. Upon detection of particular symptoms (alarms/failures) the Monitoring Functions (or the aforementioned MEs/DEs) report their observations to a set of common fault/error/failure/alarm repositories which in turn notify the FM\_DE about the newly arrived information. Consequently, the FM\_DE performs Fault-Isolation by employing the features of its embedded *Fault-Diagnosis /Localization/Isolation block (FDLI functions)*. The FDLI functions aim at finding the root cause for the observed symptoms based on fault propagation models represented by a *Causality Model* and stored in the *Causality Model Repository (CMR)* inside an autonomic node. Once the Fault-Isolation is successfully completed, the FM\_DE removes the isolated root cause (fault) by employing the *Fault-Removal Functions (FRF)* block. The FRF functions perform actions such as “reloading a functional entity”, “rebooting a node/device” or “reconfiguring a functional entity”. After executing the intended Fault-Removal actions, the FM\_DE ensures that the taken decision was indeed the right one and the source of the problem has been successfully eliminated. This is realized by the *Fault Removal Assessment Functions (FRAF)* part of the FM\_DE. The assessment can be realized either by employing some probing and network debugging methods, or by using the risk assessment services provided by the RS\_DE, as pointed out in the next sections. However, before executing a Fault-Removal function, the FM\_DE ensures that its tentative action is indeed contributing to the overall fitness of the network and is not contradicting with other actions intended by the RS\_DE. Therefore the FM\_DE and the RS\_DE refer to the *Action Synchronization Functions (ASF)*, embedded in the FM\_DE, before

performing an action on the MEs inside a node. The ASF block is responsible for ensuring the stability of the FM\_DE and RS\_DE control loops running in parallel. Moreover, in the course of Fault-Removal, the FM\_DE must be aware which entities are affected by the undertaken actions, in order to remove the fault in a stable way without introducing additional problems to the network. Thus, knowledge about the dependencies among nodes/devices, protocol modules, and services is required, which is provided by the *Dependability Models Repository* of an autonomic node.

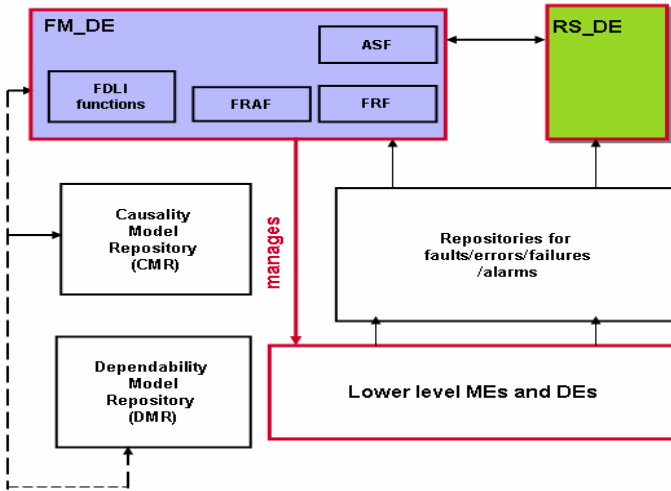


Fig. 3. The entities involved in Autonomic Fault-Management

### 3.2 Resilience and Survivability Decision Element

To support generic Resilience and Survivability functions within an autonomic node, the Resilience and Survivability DE is proposed as a core part of the presented architecture. The relations between RS\_DE and other architectural components involved in resilience and survivability functionality are depicted in Figure 4. Following the same line of arguments as in the case of the FM\_DE, we introduce the RS\_DE at the node level inside the GANA reference model (Figure 1).

The operation of the RS\_DE is based on information from the FM\_DE and from other diverse functional entities that supply information to the fault/error/failure/alarm repositories of an autonomic node. The Monitoring Components or some alarm/incident information sharing DEs and MEs, as depicted in section 3.1, supported by the node repositories, provide data related to the directly detected faults, errors and failures or generated alarms. The direct connection with the FM\_DE ensures that the RS\_DE has the up-to-date knowledge about isolated faults and the Fault-Removal process.

Having information from the FM\_DE and dedicated repositories, the RS\_DE triggers diverse reactive fault-masking mechanisms depending on the situation in the network. This process is controlled by the *Fault Masking Functions* (FMF) module

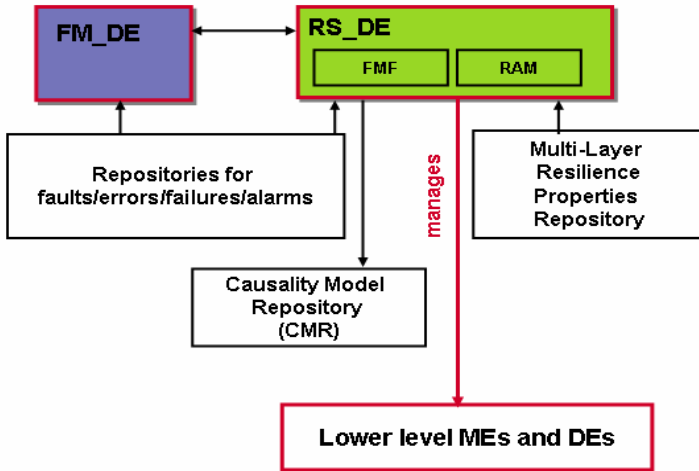


Fig. 4. The entities involved in Resilience and Survivability functionality

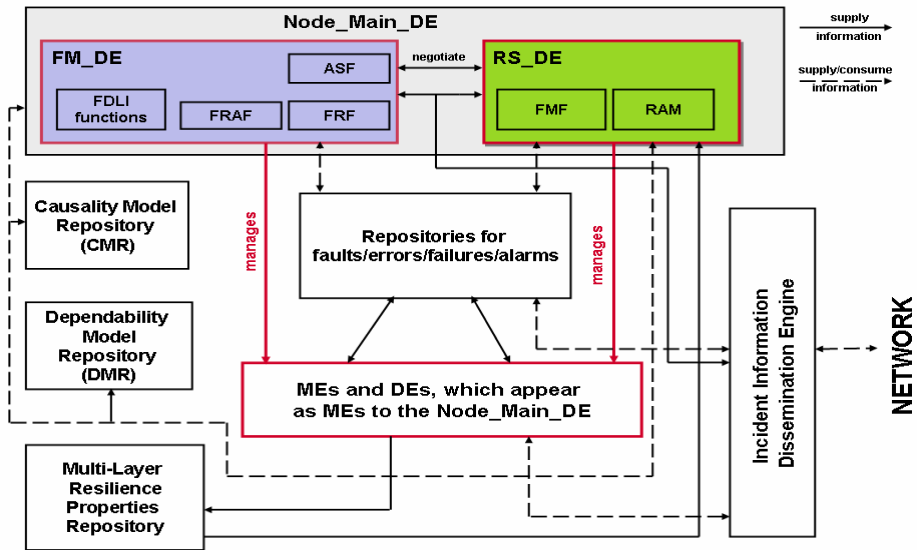
realizing *Reactive Resilience* functionalities. The role of RS\_DE is to trigger specific fault masking functionalities that would be best applicable as a reaction to the current incident(s). However specific fault masking algorithms/procedures/behaviours are not implemented within RS\_DE, but their functionality is specific to the underlying lower level MEs and DEs (e.g. protocols, mechanisms, etc.). After being informed about an incident or a fault, the RS\_DE synchronizes, i.e. negotiates, its tentative action with the FM\_DE. Based on the negotiation result, the RS\_DE discards or performs the Fault-Masking action on the corresponding MEs allowing them to execute their embedded fault masking behavior. Thus, the RS\_DE acts in a fault-tolerant manner and orchestrates resilient mechanisms of the selected node entities (MEs of this DE) to ensure acceptable QoS and sustain network functionality. In order to select appropriate solutions and to manage the resilient behaviour of a node, the RS\_DE must be aware of the resilient mechanisms implemented intrinsically inside the MEs which are under its control (protocol modules and other lower-level DEs). This knowledge is stored and retrieved from a dedicated repository – *Multi-Layer Resilience Properties Repository*. Moreover, before imposing its own decisions, the RS\_DE is required to give the low level MEs a chance to recover based on their own intrinsic resilient mechanisms. However, managerial responsibilities belong to the RS\_DE and thus it has the right to intercept the intrinsic actions undertaken by the functional entities if they are contradicting.

While the FMF module of the RS\_DE takes reactive actions upon detected incidents, the *Risk Assessment Module* (RAM) tries to predict the occurrence of certain incidents in the near future. As a result, the RS\_DE is able to proactively steer the node entities such that the node and/or network can prepare for the problems that are likely to manifest in the future. As a consequence, it is possible to avoid future fault activations. The RAM module enables the FM\_DE and the RS\_DE to shift from basic, reactive only schemes, to new proactive ones where prediction allows the framework to anticipate failures as well as dynamically improve the Causality Model at the same time.

## 4 The Overall Architecture for Autonomic Fault-Management, Resilience and Survivability for Self-managing Networks

All the modules of the architecture, as well as their relationships are presented in Figure 5. As it can be observed, the FM\_DE and the RS\_DE, along with their embedded components, play a central role inside a device. The MEs and DEs inside a node are managed by means of instantiated GANA control loops and exchange knowledge with the core DEs of the architecture over a set of faults/errors/failures/alarms repositories.

*Autonomic Node:*



**Fig. 5.** The unified node architecture for Resilience, Survivability and Autonomic Fault-Management for self-managing networks

There are several points where the FM\_DE and the RS\_DE interact with each other. The fault/error/failure/alarm repositories are one of the links between both DEs. Secondly, the RAM module inside the RS\_DE exposes its knowledge to the CMR repository, such that the adjustment of parameters inside the Causality Model is based on changes in the likelihood for some incidents or node/device failures. Moreover, the information from the RAM can be used to assess whether a Fault-Removal action undertaken by the FM\_DE was successful or not. That is, by monitoring the probability (risk) for devices to fail, the *Fault Removal Assessment Functions (FRAF)* can verify the effectiveness of the Fault-Removal process. Another interaction between the RS\_DE and the FM\_DE takes place during the synchronization of management actions, which is realized by the ASF module of the FM\_DE. Besides the above mentioned mechanisms and components, we argue that the sharing and dissemination of incidents/alarms between the functional entities of a node/device and

the network as a whole is vital for implementing Autonomic Fault-Management, Resilience and Survivability. Therefore, similar to [7] we propose a component called the *Incident Information Dissemination Engine*. This is also the component which is suitable for accommodating the notion of *Survivability Requirements (SR)* as presented in section 3. Hence, besides the general benefits for the applications due to the increased robustness of the network layer, the *Incident Information Dissemination Engine* would enable applications and services to increase their adaptability according to the current faults/errors/failures/alarms events across the network.

In addition to the node intrinsic mechanisms presented above, we now elaborate on Autonomic Fault-Management, Resilience and Survivability in the network as a whole. First, a distributed collaboration of FM\_DEs and RS\_DEs is expected. The core architectural DEs located in different nodes would strive to have a consistent view of the knowledge related to the incidents in the network by employing the services of the *Incident Information Dissemination Engine*. Thereafter, each FM\_DE in a GANA node would proceed with the Fault-Isolation, and eventually remove the isolated fault (root cause) if it is directly or closely related to the local node for which the FM\_DE is responsible. Alternatively, if none of the node-level FM\_DEs are able to resolve the problem, a centralized network-level FM\_DE (Figure 1) would be requested to react in the best possible way or even escalate the problem to the network operation personnel. The same procedure is applicable to the RS\_DE, which in parallel to the FM\_DE, would try to mask the activated faults and would ensure an acceptable QoS in the network while the faults are being removed.

## 5 Conclusions and Future Work

As self-managing networks are emerging, the aspects of Resilience, Survivability and Fault-Management, now becoming autonomic, are converging and require a unified architectural framework enabling them to inter-work harmoniously in an autonomic fashion. Our work aims at developing such an architecture by identifying the design principles and the major components that nodes/devices inside an autonomic network must be equipped with. In order to introduce the reader to our vision, we presented briefly the GANA, which is the reference model for autonomic network engineering that we instantiate. Furthermore, we conducted a requirements analysis of the factors that need to be addressed by the unified architecture for Resilience, Survivability and Autonomic Fault-Management for self-managing networks. Taking into account the diverse requirements, we defined the major decision making components, as well as additional entities that support the processes inside our architectural framework. The proposed framework brings a number of advantages to the services and applications running on top of the network layer. Besides the expected improvement in the robustness, dependability, and reliability of the network, our architecture provides components for application/service awareness inside the network layer. Such components enable the cross-layer information sharing and provide the means for services and applications to express survivability requirements with respect to incidents and alarms. This opportunity would enable the adaptability of the application layer to the challenging conditions the network is exposed to.

At the current stage we are developing some components of the presented architecture with the aim of conducting evaluations in diverse network environments – fixed and MANETS (Mobile ad hoc Networks). Case studies related to the impact of the proposed mechanisms on the application layer are also envisioned. In the near future, we will be working towards providing a proof of concept that our proposed architecture indeed improves the performance and dependability of the network and thus helps maintaining the QoS provided to the end user.

**Acknowledgment.** This work has been partially supported by EC FP7 EFIPSANS project (INFSO-ICT-215549).

## References

1. Chaparadza, R.: Fraunhofer FOKUS Institute for Open Communication Systems Berlin, Germany: Requirements for a Generic Autonomic Network Architecture (GANA), suitable for Standardizable Autonomic Behavior Specifications for Diverse Networking Environments, Published by the International Engineering Consortium (IEC). Annual Review of Communications 61 (December 2008)
2. Green, C.J.: Protocols for a self-healing network. In: IEEE Conference Record, Military Communications Conference, MILCOM 1995, November 6, vol. 1, pp. 252–256 (1995)
3. Larrabeiti, D., et al.: Multi-Domain Issues of Resilience. In: The proceedings of the 7th International Conference on Transparent Optical Networks, Barcelona, Catalonia, Spain, July 3-7 (2005)
4. Rai, S., Mukherjee, B., Deshpande, O.: IP resilience within an autonomous system: current approaches, challenges, and future directions. IEEE Communications Magazine 43(10), 142–149 (2005)
5. Suwala, G., Swallow, G.: SONET/SDH-like resilience for IP networks: a survey of traffic protection mechanisms. IEEE Network 18(2), 20–25 (2004)
6. Touvet, F., Harle, D.: Network Resilience in Multilayer Networks: A Critical Review and Open Issues. In: Proceedings of the First International Conference on Networking-Part 1, July 09-13, pp. 829–838 (2001)
7. Chaparadza, R.: UniFAFF: A Unified Framework for Implementing Autonomic Fault-Management and Failure-Detection for Self-Managing Networks. International Journal of Network Management (2008)
8. Baliosian, J., Sailhan, F., Devitt, A., Bosneag, A.-M.: The Omega Architecture: Towards Adaptable, Self-Managed Networks. In: Proceedings of the 1st Annual Workshop on Distributed Autonomous Network Management Systems, Dublin, Ireland (June 2006)

# Replacement Policies for Service-Based Systems

Khaled Mahbub and Andrea Zisman

Department of Computing, City University London, Northampton Square,  
London EC1V 0HB, UK

{k.mahbub,a.zisman}@soi.city.ac.uk

**Abstract.** The need to change service-based systems during their execution time has been recognized as an important challenge in service oriented computing. There are several situations that may trigger changes in service-based systems such as unavailability or malfunctioning of services; changes in the functional, quality, or contextual characteristics of the services; changes in the context of the service-based system environment; emergence of new services; or changes in the requirements of the system. However, in order to support dynamic changes in service-based systems, it is necessary to have *replacement policies* describing *what* needs to be changed, and *how* and *when* the changes should be executed. In this paper, we describe replacement policies to support dynamic changes in service-based systems. These replacement policies are used in our service discovery framework that supports proactive identification of services in parallel to the execution of the system. A prototype tool has been implemented in order to illustrate and evaluate the framework. The results of some initial evaluation are also described in the paper.

**Keywords:** service discovery, replacement policies, service adaptation, queries.

## 1 Introduction

Dynamic changes in service-based systems are considered a major research challenge for service oriented computing [18][19]. There are several situations that may trigger the need to change service-based systems during execution time. Examples of these situations are: (i) changes in the context of the service-based system environment or their participating services, (ii) changes in functional and quality aspects of services participating in service-based systems, (iii) failures in or unavailability of services participating in service-based systems, (iv) emergence of new services, or even (v) changes in or emergence of new requirements. To deal with the above situations, service-based systems need to be (a) *self-configuring*, systems that are able to automatically identify, select, and combine new services with which to interact; (b) *self-optimizing*, systems that can select the best services with which to interact in order to become more efficient; and (c) *self-healing*, systems that can detect and react to violations of functional and quality requirements, failure and unavailability of services, or changes in its context environment.

In order to provide support for dynamic changes in service-based systems, it is necessary to use *replacement policies* specifying what needs to be changed (*what*), the



ways that the changes need to be executed (*how*), and the moment that the changes should be performed in the systems (*when*).

Changes in a service-based system can range from the replacement of a service by another service, or a composition of services, to changes in the execution process (e.g., conditions, loop statements, variables, exception handlers). The changes in a service-based system can be performed by stopping the system, making the necessary changes, and resuming the system [2]. Other approaches can be used when replacing a service by another service in a system such as binding partner links during execution time of the system [12]; using proxy services as place holders for the services in a composition, instead of having concrete services referenced in the system [3][4][14]; or even using an adaptation layer based on aspect oriented programming with information about alternative services [17].

Furthermore, the moment to execute changes in a service-based system should also be considered in order to avoid (1) making changes when it is not really necessary, or (2) making changes that may cause the system to behave incorrectly. As an example of case (1), consider the situation when a service S1 is replaced by a service S2, but the execution process of the system never accesses the execution path that contains S1. For an example of case (2), consider the scenario in which a service-based system currently uses a direct debit service (Sa) to assist with payments of purchased items and a new service that supports credit card payment (Sb) becomes available when the system is debiting a user's bank account. In this case, it may be better to wait to replace service Sa with service Sb, instead of risking charging the user twice. However, if the direct debit service becomes unavailable it needs to be replaced so that the system can continue its operation.

Replacement policies should take into consideration (a) the situations that trigger changes in the system (e.g., situations (i) to (v) above), (b) the type of changes that needs to be performed in the system, and (c) if the parts in the system that require changes are being used. Recently, few approaches have been proposed to support adaptation of service-based systems [2][3][4][6][11]. Adaptation in these approaches consists of replacing services in service-based systems with alternative services, or composition of services. However, these approaches do not consider the problem that triggers the need for changes, and when and how to execute the changes.

In this paper, we describe different types of replacement policies for situations (i) to (v) above. In our work, changes in a service-based system consist of replacing a service participating in the system by another service. We use our proactive service discovery framework that allows for the identification of replacement services in parallel to the execution of the system due to cases (i) to (v) above [24]. We extend the framework to allow the use of proxy services to support changes in the system during execution time, avoiding changes in the original service-based system. In the framework we consider cases when (1) changes are required to be performed so that the system can continue its operations; (2) changes that can wait to be performed after the current execution of the system; and (3) no changes are required. A prototype tool incorporating the replacement policies and the deployment of the changes in service-based systems using proxy services has been implemented in order to illustrate and evaluate the work. Initial evaluations of the work comparing the performance of executing a service-based system without the need for changes with the performance to

execute the system when changes are required using the replacement policies being described is also presented.

The remaining of this paper is organized as follows. In Section 2, we provide an overview of our proactive service discovery framework with its extensions. In Section 3, we describe the replacement policies that we use to support changes in the system. In Section 4, we discuss implementation aspects and results of some initial evaluation of the work. In Section 5, we present related work. Finally, in Section 6, we present some conclusions and discuss future work.

## 2 Overview of Service Discovery Framework

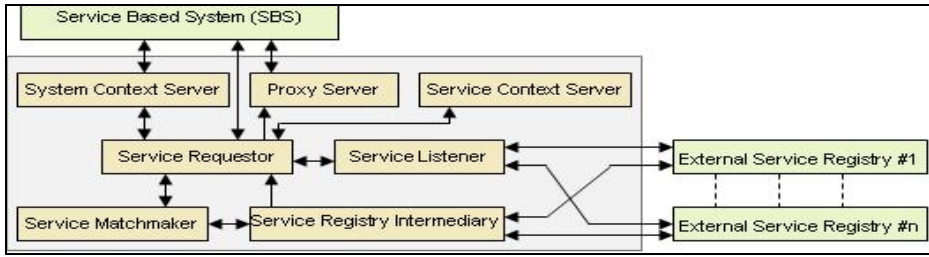
Our service discovery framework allows a proactive identification of services to replace services participating in service-based systems during execution time of these systems. The discovery framework supports the identification of services based on structural, behavioral, quality, and contextual criteria represented as complex queries. The queries are specified in an XML-based query language named SerDiQueL [25]<sup>1</sup>. The framework assumes services in registries specified as multi-faceted descriptions as proposed in the SeCSE project [20]. Figure 1 presents the architecture of our framework with its main components, namely: (i) service requestor, (ii) service matchmaker, (iii) service listener, (iv) service context server, (v) system context server, (vi) proxy server, and (vii) service registry intermediary.

The *service requestor* orchestrates the functionality offered by the other components in the framework. It (a) receives a service request from a service-based system and context information about the services and system environment, (b) prepares service queries to be evaluated, (c) organises the results of a query and returns these results to a service-based system, (d) manages subscriptions of queries and services, (e) receives information from listeners about services that become available or changes to existing services, and (f) invokes the service matchmaker to execute a query and the proxy server to execute replacement policies.

The *service matchmaker* is responsible to parse the different criteria of a query and evaluate these criteria against service specifications in the various service registries. The different criteria in a query are: (1) *structural*, describing the interface of a required service; (2) *behavioral*, describing the functionality of a required service; and (3) *constraints*, describing additional conditions of a service to be discovered. The constraints in a query can be *contextual* or *non-contextual*. A contextual constraint is concerned with information that changes dynamically during the operation of the service-based system and/or the services that the system deploys; while a non-contextual constraint is concerned with quality aspects of a required service. The non-contextual constraints can be *hard* or *soft*. A hard constraint must be satisfied by all discovered services for a query while the soft constraints do not need to be satisfied by all discovered services, but are used to rank candidate services for a query.

---

<sup>1</sup> Due to space limitation we do not describe SerDiQueL in this paper. Detailed information about SerDiQueL can be found in [25].



**Fig. 1.** Architecture overview of service discovery framework

The *service context server* and the *system context server* send updates to the service requester whenever changes of context occur in the services or system.

The *service listener* polls the external service registries at regular interval and notifies the service requester about a new service that becomes available or about changes in the characteristics of a service.

The *proxy server* supports replacement of services in service-based systems. It receives calls from the service-based system when a participating service needs to be invoked and events from the service requester when a service needs to be replaced. The proxy server keeps track of the execution of a service-based system and, when necessary, replaces a service by using the replacement policies.

The *service registry intermediary* supports the discovery of services stored in various registries by providing an interface to access services from these registries. The framework supports services from registries that are based on a faceted structure as developed in the SeCSE project [20]. In this structure, a service is specified by a set of facets representing its different aspects such as (i) structural facets describing the operations of a service with their data types in WSDL [23], (ii) behavioral facets describing behavioral models of services in BPEL [5], (iii) quality of service facets describing quality aspects of services represented in XML-based schema, and (iv) context facets describing the types of context information for a service represented in XML-based ontologies.

In the framework, the evaluation of a query against services in the registries is based on the computation of distances. More specifically, a query is executed in a two-stage process. In the first stage (*viz. filtering stage*), hard constraints of a query are matched against services in the registries returning a set of candidate services that are compliant with these constraints. In the second stage (*viz. ranking stage*), overall distances are computed between the query and services returned by the first stage in the process.

The overall distance between a query  $Q$  and a service  $S$  is denoted as  $d(Q,S)$  and is computed by considering the average of three partial distances, namely *structural\_behavioral*, *soft\_noncontextual*, and *contextual*. The structural evaluation is based on the matching of the signatures of the operations in a query and services by comparing graphs of the data types of the parameters of the operations and the linguistic distances of the names of the operations and parameters. The behavioural evaluation is based on the comparison of paths representing the behavioral criteria in the query and paths extracted from state machines representing behavioral specifications of services. The *soft\_noncontextual* and *contextual* partial distances are computed by evaluating

constraints in a query against service specifications. This evaluation compares context information provided by context services to context conditions in a query. Details of the computation of the overall and partial distances are described in [24].

The framework supports the execution of queries in pull and push modes. The pull mode is executed to identify services that are initially bound to a service-based system, as a first step in the push mode of query execution (to identify an initial set of candidate services), or when a client application requests a service to be discovered. The push mode is executed when the system is running and a service needs to be replaced due to cases (i) to (v) as described in Section 1. The push mode of query execution requires the system environment, the services participating in the system, and the queries associated with these services to be subscribed. In this case, candidate services for subscribed services and queries are identified in parallel to the execution of the system in a proactive way. These candidate services are maintained in an up-to-date set in which services are organized in ascending order of distances with the query. The algorithms for identifying services in a proactive way due to the different cases are described in details in [24]. When notifications about changes in the subscribed services or system environment are pushed to the listeners, the first service from the set of candidate services is selected and may be replaced in the service-based system depending on the replacement policy.

### 3 Replacement Policies

The replacement policies used in our work specify when participating services should be replaced in service-based systems due to (a) unavailability or malfunctioning of services; (b) changes in the structural, functional, quality, or context of services; (c) availability of a new service; and (d) changes in the context of the system's environment, changes in requirements, or emergence of new requirements<sup>2</sup>.

The replacement policies take into consideration the position of a service  $S$  that may need to be replaced with respect to the current execution point of the system.

There are three different positions that are considered, namely:

- *not\_in\_path*: when service  $S$  is not in the current execution path of the system, i.e.,  $S$  appears in a different branch of the system's execution path or before the current point in the execution path;
- *current*: when service  $S$  is in the current execution point of the system;
- *next\_in\_path*: when service  $S$  is in the current execution path of the system, and will be invoked some time in the future.

We describe below the replacement policies for each case (a) to (d) above. The replacement policies will be executed based on events concerned with these cases.

In order to follow the description of the policies consider  $P$  the process of the service-based system being executed;  $S_p$  a subscribed service being used in  $P$  that may need to be replaced;  $Q$  a subscribed query associated with  $S_p$  that is composed of structural,

---

<sup>2</sup> Please note that in reference to the situations described in Section 1, case (a) is concerned with situation (iii), case (b) is concerned with situations (i) and (ii), case (c) is concerned with situation (iv), and case (d) is concerned with situations (i) and (v).

behavioural, quality or contextual constraints;  $d(Q,S)$  the distance between a service  $S$  and a query  $Q$ , as introduced in Section 2;  $d_{max}(Q)$  the threshold of acceptable distance values between services and query  $Q$ ;  $Set\_S$  the set of subscribed candidate services for  $Q$ , including  $S_p$ , ranked in ascending order of the distances between the services and query  $Q$ ;  $d_{\Delta}(Q)$  the threshold used to decide about the replacement of  $S_p$  in  $P$  in different situations, as explained below;  $pos(S)$  a function that returns the position of service  $S$  in process  $P$ . The candidate services  $S_i$  in  $Set\_S$  are services that match the criteria of query  $Q$  and have distances with  $Q$  that is less or equal to the threshold distance ( $(d(Q,S_i)) \leq d_{max}(Q)$ ). All distance values are within  $[0, 1]$ . The returned values of  $pos(S)$  can be *not\_in\_path*, *current*, or *next\_in\_path*, as explained above.  $mark(S,P,CURRENT)$  is a function that marks  $S$  for replacement when  $S$  is accessed in the current execution of  $P$ ;  $mark(S,P,FUTURE)$  is a function that marks  $S$  for replacement when  $S$  is accessed in a future execution of  $P$ ;  $mark\_or\_replace(S1, S2, P)$  is a function that replaces a service or marks a service for replacement in  $P$ . The description of  $mark\_or\_replace$  function is shown in Figure 2.

```

mark_or_replace(S1, S2, P)
  If pos(S1) == not_in_path then mark(S1, P, FUTURE)
  If pos(S1) == current then replace S1 by S2;
  If pos(S1) == next_in_path then mark(S1, P, CURRENT)

```

Fig. 2. mark\_or\_replace function

**Case (a): A subscribed service  $S$  becomes malfunctioning or unavailable.** In this case,  $S$  can be either a service participating in process  $P$  ( $S == S_p$ ) or a service in the set of candidate services for  $S_p$  that is not being used in the process. In any case,  $S$  needs to be removed from the set of candidate services and unsubscribed.

```

E(unavailable/malfunctioning, Q, S)
  If S is being used in P then //S == S_p
  If S_Set is empty then // There are no available candidate services to replace S_p
    If pos(S_p) == not_in_path then mark(S_p, P, FUTURE)
    If pos(S_p) == current then
      If exist_exception(P) then execute exception handler in P;
      else throw exception(unavailable/malfunctioning,S) ;
    If pos(S_p) == next_in_path then mark(S_p, P, CURRENT);
  If S_Set is not empty then mark_or_replace(S_p, S_0, P);

```

Fig. 3. Replacement policy for unavailable or malfunctioning services

Figure 3 shows the replacement policy for case (a). As shown in the Figure, in the case that  $S$  is a service in process  $P$ , it may be necessary to replace  $S$  by another candidate service from  $Set\_S$ , if the set of candidate services is not empty (i.e., service  $S$  that was removed from  $Set\_S$  was the only service in the set). The replacement of service  $S$  by the first service in  $Set\_S$  (the one with the smallest distance), will be performed when  $S$  is currently being executed by process  $P$  (i.e.,  $pos(S) == current$ ). However, when service  $S$  is not in the current execution path (i.e.,  $pos(S) == not\_in\_path$ ) or  $S$  is in the current execution path, but should only be invoked in the

near future (i.e.,  $pos(S) == next\_in\_path$ ),  $S$  is marked to be replaced.  $S$  will be replaced by a service in  $Set\_S$ , through the proxy server, when the execution process reaches the point of  $S$  in  $P$  and  $Set\_S$  is not empty at this stage.

In the situation that  $Set\_S$  is empty and  $S$  is currently being executed by process  $P$ , an exception will be thrown. This exception can be either from process  $P$ , when such exception exists in  $P$ , or an exception specified in the replacement policy. In the case that  $S$  is not in the current execution path, or  $S$  is in the current execution path but should only be invoked in the future,  $S$  is marked to be replaced and an attempt to replace  $S$  will be executed in the future, when the process reaches the respective point of execution. In the situation in which  $S$  is not being used in  $P$ , nothing needs to be done with respect to replacement policy.

**Case (b): A subscribed service  $S$  has its structural, behavioral, quality, or contextual characteristics changed.** As in case (a),  $S$  can be either a service participating in process  $P$  ( $S == S_p$ ) or a service in the set of candidate services for  $S_p$  that is not being used in the process. A new distance between  $S$  and  $Q$  needs to be calculated and if this distance is below the threshold,  $Set\_S$  is re-ordered. Otherwise,  $S$  is removed from  $Set\_S$  and unsubscribed.

Figure 4 shows the replacement policy for case (b). As shown in the figure, when  $S$  is a service in process  $P$ , but the set of candidate services is empty (i.e.,  $S$  was the only service available for  $Q$ , but changes in its characteristics caused it not to be suitable anymore),  $S$  is marked to be replaced when the execution process reaches the point of  $S$  in the process, for the situations in which  $S$  is neither in the execution path nor being currently executed. Otherwise, an exception is thrown.

When the set of candidate services is not empty, it is necessary to verify if  $S$  is still the best service to be used in the process. In positive case, nothing needs to be done. However, when  $S$  is not the best service anymore, it is necessary to verify how bad it will be to continue using  $S$ , instead of replacing  $S$  by the best service (i.e., service  $S_0$ ).

```

E(change,Q,S):
if  $S$  is being used in  $P$  then //  $S == S_p$ 
  if  $Set\_S$  is empty then // There are no available candidate services to replace  $S_p$ 
    if  $pos(S_p) == not\_in\_path$  then mark( $S_p$ ,  $P$ , FUTURE)
    if  $pos(S_p) == current$  then
      if  $exist\_exception(P)$  then execute exception handler in  $P$ ;
      else throw exception(no available candidate service for  $S$ ) ;
    if  $pos(S_p) == next\_in\_path$  then mark( $S_p$ ,  $P$ , CURRENT)
  if  $Set\_S$  is not empty then
    if  $S_0 == S$  then do nothing; // despite the changes,  $S$  is still the best service
    else
      if  $d(Q, S) - d(Q, S_0) \leq d_{\Delta}(Q)$  then mark( $S_p$ ,  $P$ , FUTURE)
      else //  $d(Q, S) - d(Q, S_0) > d_{\Delta}(Q)$ 
        mark_or_replace( $S_p$ ,  $S_0$ ,  $P$ )
  if  $S$  is not being used in  $P$  then //  $S != S_p$ 
    if  $S_0 != S$  then do nothing; //after the changes,  $S$  is not the best service
    if  $S_0 == S$  then //  $S$  is now best service
      if  $d(Q, S_p) - d(Q, S_0) \leq d_{\Delta}(Q)$  then mark( $S_p$ ,  $P$ , FUTURE);
      if  $d(Q, S_p) - d(Q, S_0) > d_{\Delta}(Q)$  then mark_or_replace( $S_p$ ,  $S_0$ ,  $P$ );
  
```

**Fig. 4.** Replacement policy for changes in the structural, behavioral, quality, or contextual characteristics of a service

This verification is done by calculating the value of the difference of the distance between  $S$  and query  $Q$  and the distance between the current best service in  $\text{Set\_S}$  ( $S_0$ ) and query  $Q$ , and verifying if this value is acceptable (i.e., if the value is less or equal to  $d_{\text{delta}}(Q)$ ). If this is the case,  $S$  should not be replaced, but it should be marked to be replaced in the future. If the difference of the distance is not acceptable (i.e.,  $d(Q,S) - d(Q,S_0) > d_{\text{delta}}(Q)$ ),  $S$  should be replaced by the current best service  $S_0$ , when  $S$  is in the current point of process execution. Otherwise,  $S$  should be marked to be replaced in the future.

In the case that  $S$  is not being used by process  $P$  ( $S$  is a service in  $\text{Set\_S}$ ), it is possible that  $S$  is still not the best service and, therefore, nothing should be changed. However, if  $S$  becomes the best service in  $\text{Set\_S}$ , it is necessary to verify the difference of the distances of the current service in the process related to  $Q$  and service  $S$ . When this difference is acceptable, the current service in process  $P$  ( $S_P$ ) is marked for replacement in a future execution of  $P$ . But, when the difference is beyond the acceptable threshold ( $d_{\text{delta}}(Q)$ ),  $S_P$  needs to be replaced by  $S$ , when it is currently being executed in  $P$ . Otherwise,  $S_P$  is marked for future replacement.

**Case (c): A new service  $S$  becomes available.** This case is concerned with the situation in which a new service  $S$  is provided and inserted in the service registry or an existing service in the registry, that is not part of a set of candidate services for subscribed services and queries, has been modified. The distance between service  $S$  and query  $Q$  is calculated and if the distance is above the threshold,  $S$  is not considered as a candidate service for  $Q$  and nothing needs to be done. When the distance is below the threshold,  $S$  is added into  $\text{Set\_S}$ .

<pre> <b>E(new,S):</b> <b>If</b> <math>S</math> is in <math>\text{Set\_S}</math> <b>then</b> //the new service is a candidate service   <b>If</b> <math>S_0 \neq S</math> <b>then</b> do nothing; // the new service <math>S</math> is <b>not</b> the best service   <b>If</b> <math>S_0 == S</math> //the new service <math>S</math> is the best service     <b>If</b> <math>d(Q, S_P) - d(Q, S_0) \leq d_{\text{delta}}(Q)</math> <b>then</b> <b>mark</b>(<math>S_P, P, \text{FUTURE}</math>);     <b>If</b> <math>d(Q, S_P) - d(Q, S_0) &gt; d_{\text{delta}}(Q)</math> <b>then</b> <b>mark_or_replace</b>(<math>S_P, S_0, P</math>); <b>If</b> <math>S</math> is not in <math>\text{Set\_S}</math> <b>then</b> do nothing; //the new service is not a candidate service </pre>
--

**Fig. 5.** Replacement policy for a new service that becomes available

Figure 5 shows the replacement policy for this situation. In the case that  $S$  was included in  $\text{Set\_S}$ , but is not the best service for  $Q$ , nothing needs to be done. However, when  $S$  is the best match for query  $Q$  ( $S == S_0$ ), it is necessary to verify the difference of the distances of the current service in the process ( $S_P$ ) and service  $S$ . When this difference is acceptable, the current service in process  $P$  ( $S_P$ ) is marked for replacement in a future execution of  $P$ . But, when the difference of the distances is beyond the acceptable threshold ( $d_{\text{delta}}(Q)$ ),  $S_P$  needs to be replaced by  $S$ , when  $S_P$  is in the current execution point of  $P$ . Otherwise,  $S_P$  is marked to be replaced in the future.

**Case (d): Changes in the context of the system environment, changes in the requirements, or emergence of new requirements.** In this case there is a change in the criteria of query  $Q$  and a new query  $Q'$  is created. Therefore, it is not always the case

```

E(constraint,Q,Q'):
If Set'_S is not empty then
  If Sp is in Set'_S then //Sp is a candidate service for Q'
    If Sp == S0 then do nothing;
    else
      If d(Q', Sp) - d(Q', S0) <= ddelta(Q') then mark(Sp, P, FUTURE)
      If d(Q', Sp) - d(Q', S0) > ddelta(Q') then mark_or_replace(Sp, S0, P)
    If Sp is not in Set'_S then mark_or_replace(Sp, S0, P) //Sp is not a candidate service for Q'
If Set'_S is empty then //There are no available candidate services for Q'
  If pos(Sp) == not_in_path then mark(Sp, P, FUTURE)
  If pos(Sp) == current then
    If exist_exception(P) then execute_exception_handler in P;
    else throw exception(no available candidate service for S);
  If pos(Sp) == next_in_path then mark(Sp, P, CURRENT);

```

**Fig. 6.** Replacement policy when there are changes in the context of the system environment, changes in the requirements, or new requirements

that the current set of candidate services for query  $Q$  are still candidate services for  $Q'$ , and that the subscribed service used in process  $P$  associated with  $Q$  is a candidate service for  $Q'$ . A new set of candidate services for  $Q'$  needs to be created ( $Set'_S$ ), and the distance between  $S_p$  and  $Q'$  is calculated.

Figure 6 shows the replacement policy for this situation. As shown in the figure, when there are available candidate services for  $Q'$  ( $Set'_S$  is not empty) and  $S_p$  is still the best candidate service for  $Q'$ , nothing needs to be done. However, when  $S_p$  is a candidate service for  $Q'$ , but not the best service, and the difference in the distances of  $S_p$  and the best candidate service for  $Q'$  is acceptable, then  $S_p$  is not replaced, but it is marked to be replaced in a future execution of process  $P$ . When the difference in the distances of  $S_p$  and the best candidate service for  $Q'$  is not acceptable, then  $S_p$  is replaced by the best candidate service when  $S_p$  is in the current execution point. Otherwise,  $S_p$  is marked to be replaced in the near future. In the situation in which  $S_p$  is not a candidate service for  $Q'$ , or there are no candidate services for  $Q'$  ( $Set'_S$  is empty), either an exception is thrown or  $S_p$  is marked to be replaced in the future, when accessed during the execution of the process.

## 4 Implementation Aspects and Evaluation

A prototype tool of the framework has been implemented in Java. The tool is available as a web service and can be deployed by any client that can produce service requests in the format required by the framework. The subscription of the services is supported by WS-Eventing [22] and by an event receiver. The external service registry uses eXist [8] database. Communication with the registry is through the use of Remote Method Invocation (RMI). The proxy server has been implemented as an HTTP server using Java socket programming.

The work was initially evaluated to measure the delay in the execution process that may be caused when using the approach. More specifically, we measure the times to execute a service-based system without the need for changes and compare this value with the times to execute the system when changes are required using our replacement



policies. The times were calculated as the average of 60 executions using a Pentium 2.33 GHz with 3.23 GB RAM machine.

In the experiment we have used a *Route-Planner* service-based system specified as a BPEL [5] process that allows users to request information from a PDA about optimal routes to be taken when driving. More specifically the system offers services that (i) identify the exact current location of a user (S\_Loc), (ii) allow users to find an optimal route for a certain location given the exact location of the user by using a *Global Positioning Service* (S\_GPS), (iii) display colored electronic maps of the area where the user is located and the route to be taken supported by the use of e-AZ Map service (S\_e-AZ), (iv) provide traffic information in the area where the user is located and in the route that the user is supposed to take to get to his/her destination by using *Road Traffic Service* (S\_RT), and (v) compute new routes at regular intervals due to traffic changes (S\_Route).

The query used in the experiment was specified in SerDiQueL [30] and is concerned with the identification of candidate services to replace the Global Positioning Service (S\_GPS) in the system. This query has structural, behavioral, and soft quality constraints. The structural constraint is concerned with the interface of the S\_GPS service, the behavioral constraint is concerned with the existence of a certain operation (e.g., get\_location()), and the quality constraint specifies that the service should be available 24 hours per day.

We have executed the query for situations when (a) service S\_GPS becomes unavailable; (b) there is a change in service S\_GPS and this service is available only 12 hours per day, instead of 24 hours; and (c) a new better service S\_GPS' becomes available. For case (c), we consider the scenario in which S\_GPS initially used in the system was available only 12 hours and the new service (S'GPS) is available 24 hours. We used a distance threshold ( $d_{\max}(Q)$ ) in the experiment such that there is always an average of four services in the set of candidate services, and a threshold for replacement condition ( $d_{\delta}(Q)$ ) of value zero.

**Table 1.** Results of average response time in seconds

No required changes	Service unavailable	Change in service	New service
0.48	0.56	0.52	0.54

Table 1 presents the times in seconds for the experiment. The results show that although there is an increase in the average response time when using our replacement policies to support changes in the service-based system, this value is small when compared to the ideal situation when no changes are necessary to be executed in the system. Moreover, the results also demonstrate that the small penalty regarding the use of the policies and changes in the system is very similar for all the cases considered in the experiment and changes in the system is very similar for all the cases considered in the experiment, due to the similarity of the policies for each case.

## 5 Related Work

Several approaches have been proposed to support service discovery and adaptation of service-based systems. We present below some of these approaches.

Approaches for service discovery can be based on semantic matchmaking and use logic reasoning over terminological concept relations defined by ontologies [1][13]. Other approaches specify requests and services using graph transformation rules [10]. The approach in [13] focuses on operation signature checking based on string matching, but it is limited since it cannot account for changes in the order or names of the parameters. The approach in [9] advocates the use of (abstract) behavioural models of services to increase the precision of the discovery process. In [7], context information is represented by key-value pairs attached to the edges of a graph representing service classifications. This approach does not integrate context information with behavioural and quality matching and, context information is stored explicitly in a service repository that must be updated following context changes.

Overall, most of the proposed approaches support service discovery for only specific types of service criteria. Unlike them, our framework supports dynamic service discovery based on a comprehensive set of service and application properties including structural, functional, quality, and contextual properties. It also provides proactive service discovery mechanisms, optimising service replacement during the execution of an application.

In [15][16] mechanisms are presented for policy based adaptation of networks. In these approaches, reconfiguration of hardware (e.g. alter the queuing strategy in a router, increase the buffer size) or software (e.g. restrict unauthorized access, enable different encoding) components is suggested based on a set of policies to optimise the performance of the network. These approaches may require execution of alternate workflow depending on the adaptation policy, whereas in our approach we ensure the execution of the same workflow by amending the workflow.

Recently, a few approaches that support adaptation of service-based systems have started to appear. The dynamic binding approach described in [4][6] provides binding and reconfiguration rules to support evolution of service compositions during runtime.

The works in [2][3] propose approaches towards self-healing for services compositions based on monitoring and recovery actions. In [2], the recovery actions involve *retry* of the process task, *redo* of the process task, *substitute* the service by a candidate service, or *compensate* an executed task by a compensation action. Contrary to our work, in this approach, when a fault is detected, the process is suspended and moved to a repair mode. When the recovery actions are completed, the process is resumed.

The VieDAME framework [17] uses an aspect-oriented approach to allow adaptation of service-based systems for certain QoS criteria based on various alternative services. In the framework, a service participating in the system can be marked as replaceable to indicate that alternative services can be invoked instead of the original one, when necessary.

In [10], the authors propose PROSA, a proactive adaptation approach for service-based systems based on online testing. Although, the focus of this paper is on how to detect the need for changes in service-based systems before they occur, and not how to modify the system and when services should be replaced, this work is interesting and we intend to extend our framework to support proactive detection of necessary changes in service-based systems together with the proactive identification of candidate services to replace participating services.

Our framework complements existing approaches for adaptation of service-based systems. Contrary to existing approaches, our framework provides replacement policies

for different situations that may require changes in the system. These policies avoid replacing services unnecessarily. In addition, the services to be replaced are identified in parallel to the execution of the system in a proactive way.

## 6 Conclusion and Future Work

In this paper we presented replacement policies to support changes in service-based systems due to different situations such as (i) changes in functional and quality aspects of services participating in service-based systems, (ii) failures in or unavailability of services participating in service-based systems, (iii) emergence of new services, (iv) changes in the context of the service-based system environment or their participating services, or even (v) changes in or emergence of new requirements. The replacement policies consider the cases in which changes need to be performed so that the system can continue its operations; changes can wait to be performed after the current execution of the system; and no changes are required. These policies have been used in a service-discovery framework that we have developed in order to support proactive identification of services in parallel to the execution of the service-based system, in terms of structural, behavioral, quality, and contextual characteristics. A prototype tool has been implemented in order to illustrate and evaluate the work. Initial experiment of the work has shown that the use of the replacement policies does not cause an overhead in the performance when compared to the execution of a service-based system that does not require changes to be performed.

We are currently extending the replacement policies to support cases in which changes to the service-based system may be concerned with modifications of the execution process or to the replacement of a service by a composition of services. We are also executing more experimentation of the work to compare the time necessary to use the policies when changes in the system are executed by instrumentation of the BPEL [5] engine and by changing the code of the system.

## Acknowledgement

The research leading to these results has received funding from (a) the European Community's Seventh Framework Programme [FP7/2007-2013] under Grant Agreement 215483 (S-Cube) and (b) the European Commission under the IST Programme as part of the project GREDIA (contract FP6-34363).

## References

1. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: Constraint Driven Web Service Composition in METEOR-S. In: International Conference on Services Computing (2004)
2. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A Framework for Executing Adaptive Web-Service Processes. *IEEE Software* 24(6) (2007)
3. Baresi, L., Ghezzi, C., Guinea, S.: Towards Self-Healing Compositions of Services. *Studies in Computational Intelligence*, vol. 42. Springer, Heidelberg (2007)
4. Baresi, L., Di Nitto, E., Ghezzi, C., Guinea, S.: A Framework for the Deployment of Adaptable Web Service Compositions. *Service Oriented Computing and Applications* 1(1) (April 6, 2007)

5. BPEL4WS,  
<http://www128.ibm.com/developerworks/library/specification/ws-bpel/>
6. Colombo, M., Di Nitto, E., Mauri, M.: SCENE: A Service Composition Execution Environment Supporting Dynamic Changes Disciplined through Rules. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 191–202. Springer, Heidelberg (2006)
7. Doulkeridis, C., Loutas, N., Vazirgiannis, M.: A System Architecture for Context-Aware Service Discovery. *Electr. Notes Theor. Comput. Sci.* 146(1), 101–116 (2006)
8. eXist, <http://exist.sourceforge.net>
9. Hall, R.J., Zisman, A.: Behavioral Models as Service Descriptions. In: International Conference on Service Oriented Computing, ICSOC, New York (2004)
10. Hausmann, J.H., Heckel, R., Lohman, M.: Model-based Discovery of Web Services. In: International Conference on Web Services (2004)
11. Hielscher, J., Kazhamiak, R., Metzger, A., Pistore, M.: A Framework for Proactive Self-Adaptation of Service-based Applications Based on Online Testing. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave 2008. LNCS, vol. 5377, pp. 122–133. Springer, Heidelberg (2008)
12. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of A Web Ontology Language. *Journal of Web Semantics* 1(1), 7–26 (2003)
13. Keller, U., Lara, R., Lausen, H., Polleres, A., Fensel, D.: Automatic Location of Services. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 1–16. Springer, Heidelberg (2005)
14. Kim, J., Lee, J., Lee, B.: Runtime Service Discovery and Reconfiguration using OWL-S based Semantic Web Service. In: Proceedings of the 7th IEEE International Conference on Computer and Information Technology (2007)
15. Lymberopoulos, L., Lupu, E., Sloman, M.: An Adaptive Policy-Based Framework for Network Services Management. *Journal of Network and Systems Management* 11(3) (September 2003)
16. Marshall, A., Hussain, S.A., Chieng, D., Gu, Q.: Dynamic Network Adaptation Techniques In An Open Network Environment. In: Intl. Conference on IT and Communications (AIT 2000), Bangkok, Thailand (August 2000)
17. Moser, O., Rosenberg, F., Dustdar, S.: Non-Intrusive Monitoring and Service Adaptation for WS-BPEL. In: 17<sup>th</sup> Int. World Wide Web Conference, WWW, China (April 2008)
18. Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M., Pohl, K.: A Journey to Highly Dynamic, Self-Adaptive, Service-based Applications. *ASE Journal* 15 (2008)
19. Papazoglou, M.P., Traverso, P., Dustdar, S., Leyman, F., Kramer, B.: Service-Oriented Computing Research Roadmap,  
[ftp://ftp.cordis.lu/pub/ist/docs/directorate\\_d/st-ds/services-research-roadmap\\_en.pdf](ftp://ftp.cordis.lu/pub/ist/docs/directorate_d/st-ds/services-research-roadmap_en.pdf)
20. SECSE Project, <http://secse.eng.it>
21. Subramanian, V., Gilberti, M., Doboli, A.: Online adaptation policy design for grid sensor networks with reconfigurable embedded nodes. In: Design, Automation & Test in Europe Conference & Exhibition, DATE 2009 (2009)
22. WS-Eventing, <http://www.w3.org/Submission/WS-Eventing>
23. WSDL, <http://www.w3.org/TR/wsdl>
24. Zisman, A., Spanoudakis, Dooley, J.: A Framework for Dynamic Service Discovery. In: IEEE Int. Conference on Automated Software Engineering, ASE, Italy (September 2008)
25. Zisman, A., Spanoudakis, Dooley, J.: A Query Language for Service Discovery. In: 4<sup>th</sup> Int. Conference on Software and Data Technologies, ICSoft, Bulgaria (July 2009)

# Retry Scopes to Enable Robust Workflow Execution in Pervasive Environments\*

Hanna Eberle, Oliver Kopp, Tobias Unger, and Frank Leymann

University of Stuttgart, Institute of Architecture of Application Systems  
Universitätsstraße 38, 70569 Stuttgart, Germany  
lastname@iaas.uni-stuttgart.de

**Abstract.** Recent workflow languages are designed to serve the needs of business processes running in an unambiguous world based on unambiguous data. In contrast to business processes, processes running in a real world environment have to deal with data uncertainty and instability of the execution environment. Building a workflow language for real world flows based on a workflow language for business processes therefore may need additional modeling elements to be able to deal with this uncertainty and instability. Based on a real world process scenario we analyse and derive requirements for workflow language extensions for real world processes. The contributions provided by this paper are at first to investigate, how a workflow language can be extended properly followed up by the definition of workflow language extensions for real world processes, whereas the extensions are motivated by the real world process scenario. In this paper we use the Business Process Execution Language (BPEL) as extension foundation.

## 1 Introduction

Processes running in a real world environment usually have to deal with environment instability and data uncertainty. Environment instability is caused by the fact, that people and devices are moving around in space, which affects the execution environment, e.g. a network connection to a device may break down. Data uncertainty is caused by the fact that data representing situations are derived from the interpretation of sensor and other context data and therefore has a probabilistic and uncertain character.

This instability and the probabilistic character of situations and the environment might result in runtime faults, because e.g. assumed situations might turn out wrong. These faults might force the process to terminate followed up by the compensation of work already done. For example, resources crucial for a process execution such as workers and tools are moving around in space. During the process execution it might happen that a necessary resource is currently not available, which results in the fact that an *availability fault* gets thrown. But just in the moment the process starts terminating its faulted execution, the fault causing situation might change, since the missing resource, e.g. a worker arrives at his destination and becomes available for the process. Unfortunately, the process is not able to react to the new situation since it is terminating already.

---

\* This work is partially funded by the ALLOW project. ALLOW (<http://www.allow-project.eu/>) is part of the EU 7<sup>th</sup> Framework Programme (contract no. FP7-213339).

As we demonstrated above, today's process modeling is not robust enough for real world process requirements. Since the resource availability might change during execution, as shown in the example shown above, an easy way to deal with an *availability fault* is the retry of the faulting activity, which might result in a successful completion of the activity and consequently the process, due to the fact that resources are now available.

A fundamental property of pervasive computing is to support people in their daily actions in an unobtrusive way. This leads to the challenge that first of all the system must be aware of peoples' actions. One attempt to overcome this challenge is to combine the pervasive computing paradigm with the business process management (BPM) paradigm [1]. Business processes are used to capture peoples' daily actions. But as described above existing process modeling languages are lacking in some features required for modeling pervasive processes. For example peoples' behavior is often erroneous [2]. People are starting activities. If they recognize that they are doing wrong they simply restart the execution of the activity. Furthermore, people sometimes are erratic. They jump from one activity to another activity without completing the first. Later on, they jump back to the first activity and have to start the execution of the activity from scratch. This implies that in the process instance the first activity has to be restarted.

As a result, we have to extend exiting process modeling languages in order to provide a higher flexibility. The flexibility is required in order to enable the modeling of pervasive systems using business processes. As the paradigm of service oriented computing provides inherently a great flexibility, pervasive systems more and more are realized as service based applications. The Business Process Execution Language (BPEL) [3] the most popular process modeling language for modeling business process within service-based applications. However, BPEL has its origin in the BPM domain. It was designed to model and execute business process as service orchestrations. In this case reacting to high dynamic properties like network connections has not got much point during BPEL's design phase. Compared with pervasive environments data centers of enterprises are configured to maintain certain properties like availability using redundancy by default. Hence, a lot of faults arise rarely and consequently are not considered in process models.

In this paper we provide an approach to make process models more flexible and robust in a way, that allows processes to deal with such situations in instable environments as described above. We introduce a new concept which enables to define a set of activities within a process model, which should be retried in case a certain fault happens. We realize this new concept as extension to BPEL. BPEL provides several ways how extensions can be made to the language. Hence, we discuss several approaches how BPEL can be extended with retry semantics.

For that purpose, we extend BPEL with a new modeling element while maintaining BPELs' execution semantic. The modeling element enables and extends scopes to react on faulting situations and falsely assumed situations in a more flexible way than conventional BPEL scopes are able to react today.

The paper is organized as follows. In Section 2 we examine some suitable scenarios in detail, to foster a better understanding of the concept and the problems statement of this paper. This section is followed by a section providing for the understanding

necessary background information such the current BPEL scope semantics in Section 3. Our approach is presented in Section 4. We discuss the related work in Section 5. The achievements of this paper and the outlook of our work are summarized in Section 6.

## 2 Scenarios

We investigate the pharmaceutical development of a new medication as sample scenario. This medication development may include steps, which may fail, but whose successful completion is inevitable to the successful completion of the medication development. Figure 1 shows a simplified excerpt from a process for mixing and testing a new medication. First, the ingredients to test have to be fetched. Afterwards, the ingredients are mixed and tested. After the test the process has two alternatives to continue execution. First alternative is chosen if the mixture did not fulfill the requirements. Second alternative gets chosen in case the test is successful. The three steps, fetching, mixing and testing are grouped to indicate a desired all-or-nothing behavior. During the execution of the testing, a fault might happen. For example, wrong ingredients were fetched or the ingredients were mixed in the wrong order. In this case, the laboratory has to be cleaned up and the activities restarted. The syntax used to present the process is BPMN 1.2 [4].

A slightly different behavior can be observed in following scenario. A process is designed to support a worker of a delivery service in delivering some packages at some locations in town. All navigation instructions are computed relative to the workers current location. If the delivery man loses his way and also contact to the navigation process he might drive as long as he gets connected again. The navigation service computes the route depending on the delivery mans current location, and the delivery man tries to execute the delivery task once again. In the scenario described above the process for the delivery man to deliver some packages has following characteristics. During delivery the execution of the delivery process something might go wrong, but instead of going the whole way back to the delivery starting location the process gets the new location of the worker as input and *restarts* the delivery of the package.

Both scenarios something unforeseen happens during process execution causing a fault. The first scenario includes some repair actions in case a failure happens, which are performed before the scope gets retried. The process state is repaired to put the process in a state to be able to perform the activities once again. The second scenario restarts the activities of a scope without any repair actions. The state of the process is used as new input to the new execution of the scope.

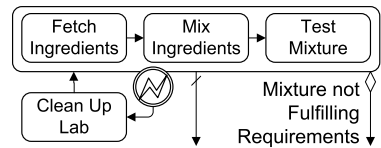


Fig. 1. Medical Test

## 3 Background Information

In this section we lay the foundation for a better understanding of this paper by introducing the basics of the BPEL's scope concept and having a deeper look at modeling elements, which play an important role in the realization of our concept.

### 3.1 BPEL Scope Semantics

Most important to foster the understanding of this paper is the BPEL `scope` activity. The scope activity is basically used to model transactional behavior. A scope encompasses a subset of activities. Additionally to these activities a scope enables to annotate business logic, which gets called in case a fault happens or the scope needs to get compensated or other events occur. Fault handler define the process logic to be performed in case a fault happens within a scope, whereas compensation handler define what has to be done to reverse successful work done already in preceding scopes. Event handler process events received by the process on scope or process level and termination handler provide additional control on terminated scopes. As other activity types in BPEL scope activities must be properly nested.

The semantics of a conventional scope can be best explained by the means of a state diagram shown in Figure 2. An activity runs through a set of states during its life-cycle. The state diagram shows the states and also what states can be reached after leaving a certain state. The BPEL specification does not impose any state model for activities. Hence, we use a simplified model, which is based on the models presented in [5, 6, 7]. This state diagram is used by the most existing BPEL implementations (e.g. Apache Ode). Depending on the engine implementation the scope activity instance is created either at process creation time or later during process execution. If the process instance gets created during the scope's instance is created, too, the scope's state is set to inactive. The latest point in time a scope instance must be created is, when the first incoming link of the scope activity is evaluated and the scopes state set into state inactive. The state inactive means here that all necessary resources for the scope's execution is allocated and it just need a trigger to set the scope into running. Once all incoming links are evaluated the activity state is set to running. During the running state the execution of the activities contained in the scope is performed and the event handlers attached to the scope get activated. After all contained activities and running event handlers are completed, the scope's state is set to state *complete* as well. Entering the state complete triggers a snapshot to be taken and the compensation handlers to be installed. The snapshot stores all scope specific data, which is needed in case the scopes compensation gets triggered, e.g. variable values at completion time, partnerLink values. If a fault happens within the scope while in state running the scope gets faulted, which means it changes to state faulted and all running activities and scopes surrounded by the faulted scope will terminate. Terminating means here to stop all surrounded running activities and scopes and so on. If all activities and scopes are terminated the scope enters the state terminated and finishes.

### 3.2 Design Goals for BPEL Extensions

The aim of this paper is to define new modeling elements for BPEL, which allows to model retry and rerun semantics. Hence, one major challenge of this work is, to find a way, how BPEL can be extended with the semantics of retry and rerun scopes in a way that complies with BPEL's extension rules. To be able to evaluate the different possible realization approaches, we need to identify design goals for BPEL extensions, which is done in the following. The design goals are partially derived from the BPEL specification and compliance rules for extensions. Other design goals are determined by usability



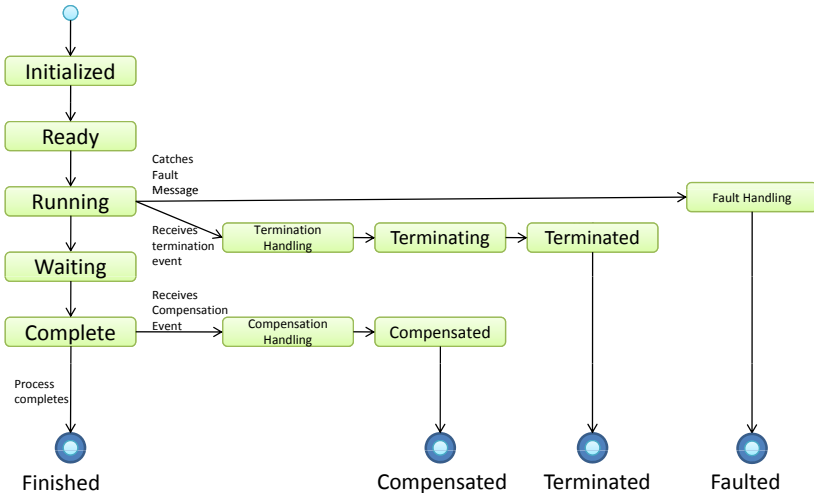


Fig. 2. Scope’s Simplified State Diagram

requirements for the new modeling element. BPEL 2.0 specification is designed extensible, which leaves us the possibility to extend any existing BPEL modeling element. Syntactically, there are no restrictions made. BPEL 2.0 specification additionally defines a framework for activity extensions. Although, syntactically any extension can be made, semantically the extensions must not change any of existing BPEL semantics. It is allowed to add functionality to the BPEL specification, but not to change it. Consequently, if a scope activity gets extended, the extension need also to fit into BPEL’s **Fault-Compensation-Termination Handling**, e.g. a completed retry/rerun scopes should be compensate-able like any other scope.

In BPEL functionality is designed as various activity types. There are activity types for data handling operations, activity types to define the control flow and basic activity types. To summarize, what has just been stated; BPEL’s design paradigm follows the paradigm to design functionality as activity types, and not e.g. as attributes to a very basic activity type. Making all these functionalities explicit in that way makes it easier to be understood by the modeler. The usability of every extension must be investigated separately. From the usability point of view, we argue, that an extension to BPEL should be easy to understand and use.

Therefore to define an extension to the existing workflow language BPEL is a tightrope walk between following requirements:

1. the modeling element should as far as possible follow BPEL’s modeling paradigm
2. the modeling element should support the required semantics as precise and concise as possible
3. the modeling element shall support the modeler’s work and should be therefore intuitive comprehensible and usable
4. the modeling element should not define semantics redundant to functionality already defined.

## 4 Retry/Rerun Scopes: The Concept Extending BPEL with Retry/Rerun Scopes

In the following we define our realization approach for the scenario based on the results of the scenario analysis and the short introduction to BPEL and BPEL scopes. But first we shortly recall the results of the scenario analysis. We extracted the following two types of behavior.

1. Retry behavior says, that some activities need to be redone due to a faulting situation, but before those activities can be redone, some things need to get ‘cleaned up’.
2. The restart behavior shows almost the same characteristics, but is a bit simpler due to the fact that conducted activities need not be ‘cleaned up’.

The purpose of this paper is to define new modeling elements in BPEL to support the modeler in modeling the behavior as presented in the scenarios and to specify the semantics of these BPEL extensions.

### 4.1 Discussing Possible Realization Approaches

There are several possible ways to extend BPEL with *retry* and *restart* options. One way to integrate *retry* and *restart* options to BPEL is to introduce two new extension activity types. *retryScope* and *rerunScope* implement each either the rerun or the retry capabilities for scopes. The rerun/retry is scheduled if the scope gets faulted. It has to be defined how these extension activities integrate into the fault and compensation handling, which basically copies the behavior defined for normal scopes. Considering the design goals defined in section 3.2 this approach is not smart in that way, that much functionality must be redefined due to the fact that *retryScope* activity needs to redefine the whole scope semantics. Another critical point in this realization approach is, how these new scope activities can be integrated into the execution. Because all fault messages are sent to a normal scope. To integrate the new extension activity types into the execution the fault messaging mechanism has to be bending to suit the desirable behavior properly. Another integration option is to extend the scope activity with an attribute, e.g. *restartOption*. Default setting of this attribute is *none*. The scope is executed as usual. If the *restartOption* is set to *rerun*, the scope in case of an error is compensated and restarted. If the *restartOption* option is set to *restart*, the scope is just started again after terminating enclosed running activities. But regarding this modeling approach closer it appears to be inflexible in the way that it is not possible to react with different behaviors to different types of fault messages, which is possible in normal scopes of the BPEL specification. Every time a fault is thrown the scope is rerun/restarted. In this case we break with the design paradigm that every functionality is implemented as an activity type, because we indicate functionality as attribute to an existing activity type.

Retry/rerun behavior can be also modeled implicitly as extended fault handler reaction. Using this option the rerun option is triggered if the according fault message gets caught. This way we have the opportunity to force the scope to terminate immediately, because the fault is too serious. Other faults, are not that serious, but expected trigger the scope, the fault is caught, to be retried and rerun respectively. We introduce a new *rerun*

activity type. This activity can be used within a catch-block of a fault handler. Using the *rerun* activity in sequence after a *compensate* activity implements the retry option. A disadvantage of this approach is that it breaks with the BPEL design restriction that control flow modeled within a fault handler must not have a link, which points into the scope the fault handler is attached to. This approach enables also a differentiating treatment of different types of faults. The *rerun* activity type is modeled into the fault handlers catch-block, where each catch-block handles a different fault message. The retry/restart options are modeled rather implicitly. Additionally it is possible to force the scope to terminate immediately, if the fault is a serious one and gets caught by a corresponding catch block. Other faults are not that serious but expected to trigger the *rerun*/restart of the scope. A fault handler can define different reaction activities based on the type of fault which is thrown. The modeling restriction for the new activity type is, that it is only allowed to be used within a fault handlers catch-block. To reduce redundant definitions we argue to model the retry behavior option as a sequence, containing a *compensate* activity followed by a *restart* activity. This approach implements functionality as activity type and already defined scope behavior need not to be redefined.

## 4.2 Realization

Considering all possible realization approaches as discussed above we argue for the most flexible and less redundant approach, which uses a *restart* activity within a fault handler's catch-block. In the following we introduce and define the detailed semantics of the restart activity type, which triggers the re-execution of its scope.

**Restart Activity Type.** The restart activity has a syntax of the following kind: `<restart times="5" />`. The number of restarts of a fault handler for a certain fault type is restrained by the value of the `<times>` attribute. This attribute is needed to avoid causing an endless loop, which gets restarted over and over again by the restart activity. Note here, that the counter is not increased on each restart of the scope activity, but on the execution of the restart activity, where the restart activity is part of the fault handler. The counter is initialized during the initialization of the activity. Every execution of the restart activity with the same activity ID and process ID increases the counter.

```

<faultHandlers>
  <catch faultName="NoUserFound"?
    faultVariable=" BPELVariableName ">
    <sequence>
      <compensate />
      <wait />
      <restart times="5" />
    </sequence>
  </catch>
</faultHandlers>

```

**Listing 1.** Fault Handler with Restart Activity and Retry Semantic

In case the `retry/rerun` scope gets called by the same failure exceeding the boundary set by the `<times>` attribute, a fault is thrown. Therefore, we introduce new standard fault called *RestartScopeFault*. Using this fault implies that the extension must be declared with `mustUnderstand="true"`.

Listing 11 shows an example on how to use the restart activity to model *retry* behavior. The catch-block consists of the sequenced activities, `compensate`, `wait` and `restart`. The `wait` is used here to delay the restart of the scope activity. The *restart* activity must be the last activity in a branch of a catch block. The respective catch-block of a fault handler must not have any links pointing into another part of the process.

**Retry/Rerun scope semantics — Fitting Retry/Rerun scopes into standard BPEL semantics.** The introduction of the restart activity influences the conventional scope semantics since a scope must be able to change from state fault handling to state running.

A walk-through of example of Listing 11. First of all, if a fault gets caught by a fault handler the running child scopes are terminated. The `<compensate/>` activity triggers the compensation of completed child scopes. After compensation we wait as long as defined in the `wait` activity. This `wait` activity is followed by a `restart` activity. The `restart` activity sends a restart event to its corresponding scope. After this the `restart` activity completes. In the meantime the scope receives this event and changes its state from *faulted* to *running*. Therefore the scope state diagram must be extended by a transition to from state from *faulted* to *running*.

In case a fault is thrown during execution of the fault handlers catch block, the BPEL standard behavior is terminate the current execution of the fault handler and to catch the fault if possible by a fault handler of the same scope or a surrounding scope. Enclosed scopes are treated no differently as enclosed in conventional scopes, e.g. in case a fault is thrown, enclosed running scopes is terminated and enclosed already completed scopes is compensated.

We can distinguish two cases, on how the `<restart/>` activity can be used. Either it is used after a `<compensate/>` activity or without a `<compensate/>` modeled in the fault handler's catch block. In case the scope is not compensated, we need to make sure that data of the faulted execution is not lost, as described in the following section.

**Data Handling Concerning Restart Activities.** The data handling of retried scopes in case of a restart without compensation can be distinguished into two cases.

1. The restart of the scope is done without compensation and without taking over any computed knowledge during the former execution of the scope.
2. The restart of the scope is done without compensation, but with taking over of computed knowledge during the former execution of the scope. Where computed knowledge stands for data values of *variables* of the scope and *partnerLinks* respectively.

To enable option two we extend the restart activity type with a Boolean attribute *keepState*. The default value of *keepState* is `no`. If *keepState* is set to `yes`, (`<restart <keepState="yes"/>`) the values of the scopes' local data, like local `<variables>` and local `<partnerlinks>` is saved before the restart is triggered by the restart activity. This additional snapshot is needed, because the restarted scope has not been completed yet,

therefore no normal snapshot has been taken, when changing state from faulted to running. This *keepState*-knowledge is injected into the new scope instance.

Of course in the case that transactional behavior is wanted, both options described above are not suitable.

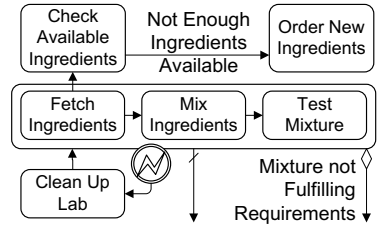
**Defining Default Compensation Semantics of Restart Scopes.** The default compensation of retry/rerun scopes diverges from the compensation of normal scopes. BPEL standard behavior takes a snapshot after a scope has completed. This snapshot is taken directly after the scope gets completed and is needed for compensation purposes. Therefore the snapshot saves the values of the scopes' local data, like local `<variables>` and local `<partnerlinks>`. This data is needed in case the scope gets the request to be compensated. A snapshot is taken after the scope enters the state *completed*. The default compensation of scopes is done by executing the compensation handlers of enclosed activities ordered by the backward control dependencies and executed instance traces. Control dependencies are not enough information due to the fact, that a scope might be enclosed by a loop activity. Every executed instance of the enclosed scope must be compensated. In case a compensation handler is defined the business logic in the compensation handler is executed, not caring about control dependencies and instance traces. The implications of the modeling options described above to compensation do also differ, the way compensation has to be performed. In the first case, where the `<restart/>` activity is sequenced after a `<compensate/>`, the compensation steps do not diverge from ordinary compensation steps, because, the state and the execution history of this scope does not diverge from an ordinary scope execution. Every executed step was compensated and therefore does not exist in the execution history any more. This does not happen to be the case, if no compensation is executed before a restart of a scope, which holds for both cases without snapshot and with snapshot. A possible execution history of a scope S sequencing the activities A, B, C and D might look like: A, B, C, A, A, B, A, B, C, D; scope S gets completed after completion of the enclosed scope activity D. The enclosed scope activity A got executed four times, scope B three times, C two times and D only one time. All instances in the execution history are compensated ordered by reverse control flow and instance history, like the scope instances created by a loop-activity.

**Navigation Implications to Cross Boundary Links.** Ordinary scopes in BPEL allow links to break through the boundary defined by activities belonging to the scope and activities, who do not. Links, which connect activities from within the scope with activities, not belonging to the scope, are called cross boundary links. In BPEL loop activities do not allow links to cross the loops boundary neither from the inside nor from the outside.

We allow cross boundary links to be modeled for rerun/retry scopes. But this has some serious implications on the navigation of the scopes enclosed activities. First we describe the problems that appear, if we do not adopt the navigation behavior, which is followed by our solution approach. If the scope has to be restarted, but some links crossing the boundary from scope to the outside scope were evaluated already and execution might have proceeded already. What happens with the second evaluation of the cross boundary link? If a scope gets executed twice, the cross boundary link needs to

be executed twice. Now, two cases can be distinguished. Either the incoming links are evaluated before the second execution of a scope is performed, which indicates, that the process has proceeded with its execution in this branch, or the other incoming links have not been evaluated so far, which indicates that the second value of the transition is taken into account by the *joinCondition* evaluation. We propose following approach in this paper. There are no modeling constraints for cross boundary links. Cross boundary links are not evaluated until the scope completes successfully. This way the cross boundary link receives the proper input values to its evaluation. Evaluation of cross boundary links get delayed until the scope completes. Note that the delay of link evaluation might cause deadlocks, if cyclic dependencies appear.

An example of a cross boundary link is shown in Figure 3. The retry scope consists of the activities, *Fetch Ingredients*, *Mix Ingredients* and *Test Mixture*. The mixture shall go into production as soon as possible. Therefore, the ingredients for production shall be ordered as soon as the ingredients are determined. The cross boundary link semantics is defined as follows. The evaluation of the link needs to wait till the scope has completed successfully, which means, that the link will be evaluated after the successful completion of *Test Mixture*.



**Fig. 3.** Medical Test with Automatic Ordering of Ingredients for Production

## 5 Related Work

In this paper we realize these concepts as BPEL modeling options and extend the BPEL semantics. A concept of rerun and retry scopes can be found in [8]. The idea of retrying activities until they succeed is also mentioned in [9]. [10] presents a retry solution for BPEL based on ECA rules: ECA rules are attached to BPEL constructs. These rules are then transformed to a standard BPEL process. Thus, retrying is not a first-class modeling construct, whereas the approach shown in this paper smoothly integrates in BPEL.

Reruns are also supported in a platform for scientific workflows called *Kepler* [11]. Scientific workflows are very data and computing power intense processes used for simulation [12]. Simulations usually have a quite respectable amount of input parameters. If a simulation with a certain parameter configuration appears to show the expected result, it is possible to rerun the process with a slightly different parameter setting. The rerun in Kepler supports ‘smart’ rerun capabilities especially to reuse already computed data which does not need to be recomputed. Kepler uses a special kind of snapshot mechanism. The approach used in Kepler focuses on the reuse of already computed data, whereas our approach focuses some actions to be redone in case a failure happens. An assessment of BPEL to scientific workflows is done in [13]. There, the need of rerunning activities is also identified. Some other workflow specifications provide a modeling element for repeats as reaction to failures. MQWF [14] FDL defines for each activity an exit condition, which is evaluated after the execution of the activity. If the exit condition does not hold, the activity is restarted. Also the ADOME [15] framework offers

the possibility to re-execute activities similar to FDL. ADOME distinguishes several activity types, among others also a repeatable activity type. Once an activity is selected for execution but fails, it can be re-executed by the same agent or alternate agents, or using alternate resources; but the WFMS should not try other alternate paths because of some reasons. (E.g. due to set up overhead for a production line or a process etc.) Both the MQWF and the ADOME framework do not support repeats over more than one activity: It is not possible to model. In MQWF the decision whether an activity is restarted is triggered by the evaluation of a condition over the input and output data of the activity after completion and not immediately on the occurrence of a fault.

Using Business Process Management Notation (BPMN) [4] restart behavior can be modeled using a work around. All activities which should be restarted can be placed in a Collapsed Sub-process. An Intermediate Error triggers the fault handling and eventually the compensation actions. Thereafter a Data-based Exclusive can be used to determine whether the process goes on or the activities are restarted by forming a loop which triggers the Collapsed Sub-process another time.

BPMN does not provide native modeling elements to model the scenarios described in this paper. The modeler has to map restart semantics to normal BPMN constructs, which does not support the modelers work in an optimal way. Same applies if using [16].

## 6 Conclusions

The crucial question of extending a Turing-complete process modeling language with new modeling elements is the question of the benefit it provides.

In our work, the benefit of our extension constitutes in the fact, that implementing the very scenarios using our extensions, the process modeler can directly model what he has in mind, whereas modeling the same behavior with a modeling workaround using a loop activity. Furthermore, during runtime we can directly monitor the executed retries and do not have to count back them from the loop cycles. As a consequence of extending BPEL also the execution engines have to be extended, which can be seen as a disadvantage. Also the modelers have to learn and use the new modeling elements. However, in our opinion the pro-language-extension-arguments are stronger than the arguments against. In particular, that the extensions allow the modeler to model what he has in his mind, argues in favor of the extension solution.

In this paper we decided to define a new modeling element and its semantics as extension to BPEL. This new modeling element can be directly motivated by the pervasive scenarios described in Section 2. The modeling element helps to model processes closer to pervasive scenarios, whereas it would be possible to model the processes anyhow without the new modeling element as well. We argue that our modeling element eases the modeling of regarded scenarios. The modeling element supports to model and comprehend process models for real world processes more intuitively. Additionally the modeling element increases the flexibility of a scope definition by adding the modeling element `<restart/>` activity. The modeling element can also consider as adaptation support. Adaptation is triggered by the fault message. Adaptation is computed during the re-execution of the scope by choosing more suitable paths. In contrast to loops, the restart option of repeatable scopes is triggered exclusively by faulting situations, which

appear to be a more suitable modeling option to some scenarios. Further research needs to be done concerning the cross boundary link semantics. In this paper we provided the simplest way on how to deal with cross boundary link evaluation at runtime. Especially we will regard the question, how can cross boundary link evaluation be done with respect to performance concerns. As there are lots of possible way to do this, this was out of scope of this paper.

## References

- [1] Herrmann, K., Rothermel, K., Kortuem, G., Dulay, N.: Adaptable Pervasive Flows - An Emerging Technology for Pervasive Adaptation. In: Workshop on Pervasive Adaptation (PerAda) (October 2008)
- [2] Norman, D.: The Design of Everyday Things. Owner inscription on fep edn. Doubleday Business (February 1990)
- [3] Organization for the Advancement of Structured Information Standards (OASIS): Web Services Business Process Execution Language Version 2.0 (March 2007)
- [4] Object Management Group (OMG): Business Process Modeling Notation (BPMN) Version 1.2 (January 2009), <http://www.bpmn.org/>
- [5] Kloppmann, M., Koenig, D., Leymann, F., Pfau, G., Rickayzen, A., von Riegen, C., Schmidt, P., Trickovic, I.: WS-BPEL Extension for Sub-processes – BPEL-SPE. In: IBM, SAP (2005)
- [6] Steinmetz, T.: Ein Event-Modell für WS-BPEL 2.0 und dessen Realisierung in Apache ODE. Diplomarbeit, University of Stuttgart, Faculty of Computer Science, Electrical Engineering, and Information Technology, Germany (August 2008)
- [7] Karastoyanova, D., Khalaf, R., Schroth, R., Paluszek, M., Leymann, F.: BPEL Event Model. Technical Report 2006/10, University of Stuttgart, Faculty of Computer Science, Electrical Engineering, and Information Technology, Germany (2006)
- [8] Leymann, F.: Supporting Business Transactions Via Partial Backward Recovery In Workflow Management Systems. In: BTW, pp. 51–70 (1995)
- [9] Greenfield, P., Fekete, A., Jang, J., Kuo, D.: Compensation is Not Enough. In: EDOC 2003: Proceedings of the 7th International Conference on Enterprise Distributed Object Computing, Washington, DC, USA, p. 232. IEEE Computer Society, Los Alamitos (2003)
- [10] Liu, A., Li, Q., Xiao, M.: A declarative approach to enhancing the reliability of bpeL processes. In: IEEE International Conference on Web Services (ICWS 2007), pp. 272–279. IEEE Computer Society, Los Alamitos (2007)
- [11] Altintas, I., Barney, O., Jaeger-Frank, E.: Provenance Collection Support in the Kepler Scientific Workflow System. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 118–132. Springer, Heidelberg (2006)
- [12] Bharathi, S., et al.: Characterization of Scientific Workflows. In: Proceedings of the 3<sup>rd</sup> Workshop on Workflows in Support of Large-Scale Science, WORKS (2008)
- [13] Akram, A., Meredith, D., Allan, R.: Evaluation of bpeL to scientific workflows. In: CC-GRID 2006: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Washington, DC, USA, pp. 269–274. IEEE Computer Society, Los Alamitos (2006)
- [14] IBM: MQSeries Workflow
- [15] Chiu, D.K.W., Li, Q.: A Meta Modeling Approach for Workflow Management System Supporting Exception Handling. Information Systems 24, 159–184 (1999)
- [16] Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Workflow Exception Patterns. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 288–302. Springer, Heidelberg (2006)



# Integrating Complex Events for Collaborating and Dynamically Changing Business Processes

Rainer von Ammon<sup>1</sup>, Thomas Ertlmaier<sup>1</sup>, Opher Etzion<sup>2</sup>,  
Alexander Kofman<sup>2</sup>, and Thomas Paulus<sup>1</sup>

<sup>1</sup> CITT GmbH, Konrad-Adenauerallee 30, D-93051 Regensburg, Germany

<sup>2</sup> IBM HRL, Haifa University Campus, IL-31905 Haifa

{Rainer.Ammon, Thomas.Ertlmaier, Thomas.Paulus}@citt-online.com

{Opher, Kofman}@il.ibm.com

**Abstract.** Business processes must become agile, respond to changes in the business environment in a timely manner and quickly adapt themselves to new conditions. Event-Driven Business Process Management (ED-BPM) is an enhancement of Business Process Management (BPM) by concepts of Service Oriented Architecture (SOA) and Complex Event Processing (CEP). The most important enhancement is the integration of services accessible via the Internet that fire events into global event clouds. The events can be processed by event processing platforms for aggregating the information into higher value complex business events. These events can be modeled in a business process execution language within a process driven Business Process Management System (BPMS) to trigger changes in control flow of a process or start other services. A reference model and a reference architecture for ED-BPM are presented, based on the NEXOF Reference Architecture. A taxonomy for classifying changes to process flow is proposed. Enhancements have to be applied to the existing standards in the BPM field, including both the design-time and the runtime. A scenario from the banking domain illustrates the main concepts and principles.

**Keywords:** Business Process Management, Complex Event Processing, Business Activity Monitoring, Software as a Service, Event Driven Architectures, Business Process Execution Language, Business Process Modeling Notation.

## 1 Introduction: Future Internet, Internet of Services and the (Re-) Active Internet of the Future

In the last years a number of voices have been arising around the topic of the so called “Future Internet” [1, 2]. Not only the foreseen limitations of the current Internet makes us think about its future evolution, but also new concepts and applications appearing recently will put the Internet to its limits. Among these new concepts is the “Internet of Services” [3]. In a world in which everything is connected to the network, the next step is thinking of the network as a worldwide trusted ecosystem of service providers, consumers and brokers buying, selling and composing services for different needs. This associates one of the most complex problems to be solved in the future Internet: How can all those elements and services be coordinated?

This paper discusses a coordination based on events. Each element or service can generate different events providing information to other elements as well as each element can consume events from other elements. In this context each actor is able to react in a number of manners to the received information leading to a new concept that we call the “(Re-) Active Internet”. To achieve this, the concept of business processes is applicable to the “(Re-) Active Internet”, whereas a business process itself can be considered as a service that sends and reacts on events.

Defining a number of business processes we can set up how the elements within the Internet react to the different events they receive. We have to deal especially with the problem that it is not feasible to define a business procedure for every possible situation. Therefore business processes have to deal with e.g. unexpected situations. Taxonomies have been defined for process flexibility like in [4] in which the authors differentiate among four types of flexibility in business processes: flexibility by design, deviation, underspecification and change. Descriptions, examples and a deep analysis of each type of flexibility are provided. An appropriate taxonomy will be discussed in the following sections. In Sec. 2 we first describe the model of Event-Driven Business Process Management (ED-BPM) and also a reference architecture as the basis of realizing ED-BPM platforms. In Sec. 3 a taxonomy for use cases as examples from different domains is presented. Sec. 4 provides an overview on current and future business process modeling and execution standards which may influence ED-BPM. In Sec. 5 we investigate the problem to map the process models to a workflow engine in a runtime system for executing and for dynamically changing the flow of a business process or for coordinating collaborating processes accordingly. For this reason WS-BPEL as a standard for the execution of business processes has to be enhanced. The conclusion in Sec. 6 also shows the next related steps in our research work.

## 2 ED-BPM as the Basic Technology Platform

A business process is a structured set of activities designed to produce a specified measurable result for a particular customer or market. The common understanding behind Business Process Management (BPM) is that each company’s unique way of doing business is captured in its business processes. They are seen as the most valuable corporate asset.

The term “Event-Driven Business Process Management” is a combination of actually two different disciplines: Business Process Management (BPM) and Complex Event Processing (CEP) [5]. In this context BPM means a software platform which provides companies the ability to model, manage and optimize their processes for significant gain. As an independent system, Complex Event Processing (CEP) is a parallel running platform that defines, analyses, processes events. The BPM- and the CEP-platform interact via events which are produced by the BPM-workflow engine, by the IT services and other event sources that have an influence on the business process. The definition of ED-BPM and its historical background is described in [6, 7, 8]. In the following, we roughly outline the basic components needed for operational ED-BPM systems. As shown in the reference model [9], basic elements can be taken from BPM platforms as well as CEP applications. ED-BPM comprises two modeling

layers for business processes as well as for events. Basically this connects the pre-operational abstraction of process modeling with the runtime observations of business process-related events occurring at execution time so the principle of how an ED-BPM platform works is on the basis of events.

With this architecture in mind and given the recent availability of commercial CEP platforms, there will be two different kinds of ED-BPM specialists in the future: Workflow modelers responsible for designing business processes and event modelers responsible for identifying and designing events as well as complex event patterns with the aim of detecting relevant situations occurring in business processes. Events and event patterns are designed on the basis of an Event Processing Language (EPL). At present no standard for an EPL is available although different commercial solutions exist on the market. They imply an SQL-like language or provide a rule based approach. Currently the appropriate standard is discussed by the CEP community (e.g. [10, 11]).

The combination of service oriented and event-oriented thinkings are drawn closely together by the emergence of event driven architectures (EDA), hence dealing with different issues and problem solving approaches. Within this movement the currently most known proposal for reference architecture is the Networked European Software & Services Initiative (NESSI) approach called NESSI Open Service Framework – Reference Architecture (NEXOF-RA) [12, 13, 14]. This approach has been enhanced with event processing capabilities in each level of the layered architecture as shown in [15]. Event processing itself may be layered to display increasing abstractions of events from a low level (technical layer) to higher level (business process layer) as proposed by David Luckham's event hierarchies [5]. The extensions on the existing NEXOF-RA have been proposed in aspects of integrating event processing and are described in detail in [15].

### **3 Challenges Regarding Collaborating and Dynamically Changing Business Processes**

The need for process flexibility has already been recognized since the middle of the nineties of the last century as a critical quality of effective services or business processes in order to provide organizations the ability to adapt to changing business circumstances [16, 17, 18]. In this section we show that the challenges in the case of collaborating and dynamically changing business processes respectively dynamically starting, stopping, terminating or changing Internet services need another taxonomy.

#### **3.1 Related Work regarding Taxonomies of Service and Process Flexibility**

Effective services and business processes must be able to accommodate changes to the environment in which they operate, e.g. new laws or changes in business strategy. The ability to encompass such changes is termed process flexibility [4]. The notion of flexibility is often viewed in terms of the ability of an organization's processes and supporting technologies to adapt to these changes [19, 20]. An alternate view advanced by [21] is that flexibility should be considered from the opposite perspective, i.e. in terms of what stays the same not what changes. Indeed, a process can only be

considered to be flexible if it is possible to change it without needing to replace it completely [22]. Hence flexibility is effectively a balance between change and stability that ensures that the identity of the process is retained [21, 23]. There have been a series of proposals for classifying flexibility, both in terms of the factors which motivate it and the ways in which it can be achieved within business processes.

According to [16, 24, 25], flexibility can be classified with respect to the types of changes it enables. The taxonomy presented in [22] includes three orthogonal dimensions: the abstraction level of the change, the subject of change, and the properties of the change, which include extent, duration, swiftness, and anticipation.

[4] take a different look into the various ways in which flexibility can be achieved and propose to distinguish “flexibility by design” for handling anticipated changes in the operating environment, where supporting strategies can be defined at design-time. A different category is called “deviation” for handling occasional unforeseen behavior, where differences with the expected behavior are minimal. Another category is “underspecification” for handling anticipated changes in the operating environment, where strategies cannot be defined at design-time, because the final strategy is not known in advance or is not generally applicable. The last category of “change” aims either for handling occasional unforeseen behavior, where differences require process adaptations, or for handling permanent unforeseen behavior.

The knowledge model of the S-Cube project [26] defines an adaptable service-based application as an application augmented with a run time control loop that monitors and modifies itself on the basis of adaptation strategies designed by the system integrators. An adaptation can be performed either because the monitoring of these services has revealed a problem or because the application identifies possible optimizations or because its execution context has changed. The context can be defined by the set of services available to compose the service-based applications, the parameters and protocols being in place, user preferences, and environment characteristics (e.g. location, time). Examples of adaptation strategies are re-configuration, re-binding, re-execution or re-planning (see also [27]).

The S-Cube project is contributing to the Nessi NEXOF-RA in which - during its preparation phase - our approach enhanced the ED-BPM-components (see chap. 3 as well as [28] and [29]). In the sense of S-Cube a service-based application is composed by a number of possibly independent services available in a network, which perform the desired functionalities of the architecture. The services can be provided by third parties and not necessarily by the owner of the service-based application. S-Cube defines a service-based application as a profound difference with respect to a component-based application, because the owner of the component-based application owns and controls all its components, while in contrast the owner of a service-based application does not own, in general, the component services, nor can he control their execution.

The approach of this paper is more comprehensive and follows a broader understanding. As the following use case shows exemplarily, a concert of independent applications and their services or processes is managed and controlled by the analysis of global event clouds (as “posets“ (partially ordered set of events)) or event streams (as “tosets“ (totally ordered set of events)) as described in [6, 7, 8].

### 3.2 A Sketch of a Taxonomy of Collaborating, Dynamically Invoking and Changing Processes

The following section describes a first sketch of a taxonomy that deals with the dynamic changes in collaborating processes. In future work it will be further elaborated taking additional criteria into consideration. The purpose of this taxonomy is twofold. First of all, it helps to systematize different criteria and requirements involved in driving dynamic changes in collaborating processes. Second, we use this taxonomy to investigate the gaps in modeling and execution process languages and to propose the corresponding enhancements.

The main focus of such process collaborations are events generated by the single process steps respectively by the service invocations and sent to a global event cloud (see e.g. [9]). For this aim processes and services have to be enhanced appropriately or the upcoming Enterprise Services Bus (ESB)-generation will be able to automatically generate such events only by configuration (see e.g. [11]). In the following we do not differentiate between the terms “process” and “service”, in fact a business process can also be perceived as a coarse grained service in the Internet.

Depending on the defined event patterns according to the use cases of a specific domain, we distinguish the following situations:

- an existing process instance is {stopped, continued, terminated, changed},
- a new process instance of the {same, different} process type is {instantiated, new defined},
- a task in an existing process instance is {stopped, continued, aborted}.

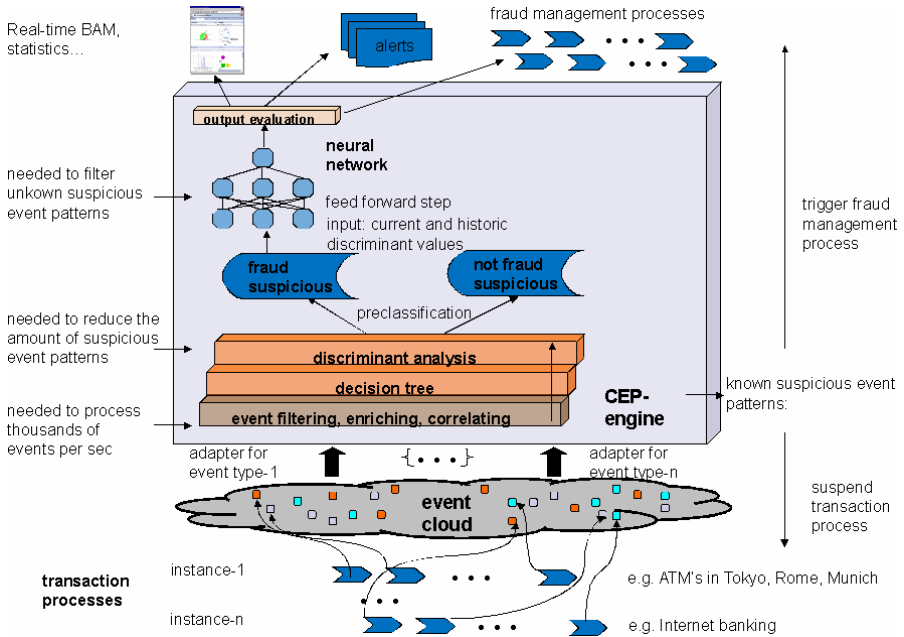
It could also make sense to include more criteria e.g. for subprocesses; such questions need still to be investigated (see e.g. the Semel-approach of the European projects SUPER (FP6-026850) and SOA4All (FP7-215219) [30]).

### 3.3 Use Case “Fraud-Management” in the Banking-Domain

In order to make it more explicit, Fig. 1 shows a reference model for ED-BPM-based fraud management for different domains like banking, insurance or retail. We use this model throughout the paper to explain possible scenarios involving event-driven collaboration between loosely coupled business processes.

The following scenario is describing a “cash withdrawal” process derived from [31], which in connection with a potential ‘fraud’ event pattern and related processes exemplifies the ED-BPM principle:

1. An instance of a transaction process starts to process a withdrawal at a certain ATM.
2. A lot of process instances of the same type are instantiated in a certain time-frame at different ATM’s.
3. Each process step generates an event that contributes to the event cloud using some messaging infrastructure (e.g. JMS publish/subscribe).
4. The global event cloud is analyzed in real-time by the CEP-system and optionally by some “intelligent” components like discriminant analysis and neural networks. A suspicious event pattern is detected because the same credit card is used more than once at different locations within a certain period of time. The following describes the behavior that should take place as a result of this fraud attempt detection.



**Fig. 1.** A reference model of ED-BPM-based, non-deterministic service and process adaptive-ness, e.g. fraud management in different domains like banking, insurance or retail [31]

5. All running instances of the same process type “cash withdrawal” for the suspicious card must be stopped as well as all their corresponding subprocesses.
6. A new process instance of the type “alert an affected branch” has to be started in order to check the customer at the ATM.
7. A new process instance of the type “fraud management” will be started. This process calls the appropriate subprocesses to update the statistics of the real-time BAM dashboard, to lock/disable the suspicious card, etc.
8. The fraud suspicion must be examined and an employee must make a decision regarding the following procedures: If no decision is made within the specified time period, a new process instance must start to escalate the problem to a higher layer in the decision making hierarchy.
9. If the decision is evaluated as a “false positive”, an ‘unlock card’ process will be instantiated, and the execution of all the process instances that were stopped as a result of the ‘fraud’ situation must be resumed.
10. If the “false positives” are too frequent (e.g. the percentage of ‘false positives’ is greater than some predefined threshold) the process type has to be modified. The process type can be changed automatically or manually. Changes may contain modifications to the workflow (e.g. adding or removing activities) or changes to values used by business logic, or both. An example of changing a workflow would be: “If the frequency of false positives is above a certain value, add a new activity of asking the client a special question”. An example of business logic modification would be: “If the frequency of the false positives is above the certain threshold, increase the maximal allowed distance between

ATMs by 5 percents.” An event serves as a trigger of a process change procedure (see pattern 5 in Sec. 5.4).

There are a lot of different and even more sophisticated event patterns of fraudulent withdraw trials which would have to be modeled accordingly [32]. Similar event patterns of fraud scenarios are also known in other domains like insurance [33] or retail [34, 35].

### 4 Current and Future Movement on Standards

With the development of ED-BPM (Event-Driven Business Process Management) a number of existing and future standards as well as different research projects, whose coherence is shown in Fig. 2. will play a major role. The coherence between the different standards and research projects are shown in Fig. 2. In January 2009 OMG released BPMN 1.2 – a business process modeling notation. Thereupon requirements and developments were leading to a new major release. While this paper was written the OMG was about to release BPMN2.0 [36]. BPMN provides a special symbol for every event type and BPMN provides some enhancements for choreography and conversations for their modeling at design time [37]. BPMN 2.0 does not provide a facility to model process external event patterns and how to react on them.

The HPI (Hasso Plattner Institute) introduced BEMN (Business Event Modeling Notation) [38] in 2007, a graphical notation for modeling complex events in business processes. According to a statement from the HPI, this project has also reached a final state and will not be enhanced anymore. Nonetheless this work may have some influence on BPMN – not concerning BPMN 2.0 but a next major release – as well as on EMP (Event Metamodel and Profile) [39].

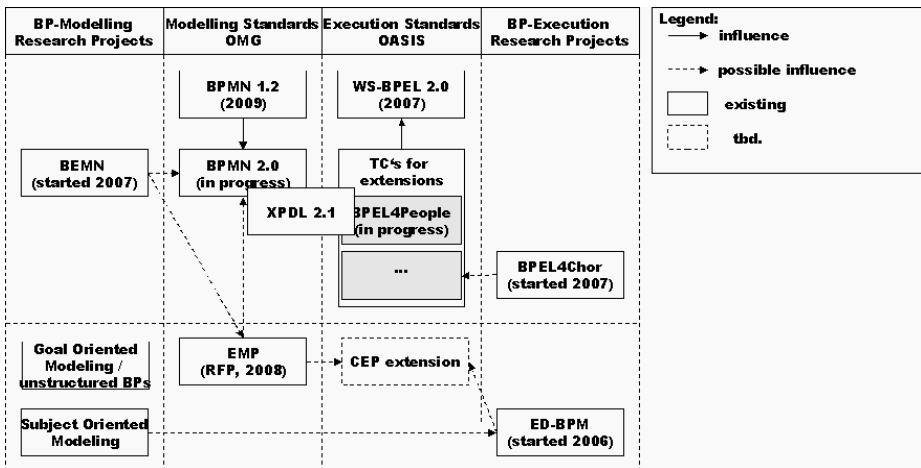


Fig. 2. Existing standards and current research projects related to ED-BPM

Another interesting approach regarding ED-BPM is called “subject oriented modeling”, a BPM modeling notation based on [40, 41]. This notation may be used due to conceptual characteristics like direct execution of modeled processes for ED-BPM technologies. First prototypes of ED-BPM are developed using this method in 2009 [42]. Nowadays are also some published approaches for the so called flexible unstructured business processes or Goal Oriented BPM, based on already existing BPM-tools [43]. A similar distinction was already known in the middle of the nineties of the last century in order to categorize workflow engines according to production workflows, ad hoc-workflows and administrative workflows. These BPM tools for unstructured BPM will not be based on the mentioned standards, but those platforms will be based on their own proprietary notations for their special focus.

A common standard for executing business process models is the WS-BPEL 2.0 standard. The actual version 2.0 is a final release and was released in April 2007. It will not be enhanced anymore and the TC (technical committee) has finished its work [44]. Further development and enhancements to WS-BPEL will be accomplished by particular BPEL extensions that will be maintained by autonomous TCs.

BPEL4People is an extension to expand WS-BPEL – in this case to support human interactions. The development of this extension is currently in progress and will be arranged by an OASIS TC [45]. Thus the extensions are enhancements to the standardized WS-BPEL.

BPEL4Chor as a research project from IAAS [46] might influence another extension of WS-BPEL regarding the challenge of choreography. Interconnecting information systems of independent business partners requires the specification of the interaction behavior the different partners have to adhere to. Choreographies define such interaction constraints and obligations and can be used as starting point for process implementation at the partners sites. The BPEL4Chor project showed how the BPMN and WS-BPEL can be used during choreography design. Step-wise refinement of choreographies to the level of system configuration is supported through different language extensions as well as a mapping from BPMN to BPEL4Chor.

A corresponding modeling environment incorporating the language mapping was also investigated. Complex choreographies as intended in the use cases of our approach cannot be created within a single step. Whenever many different business partners and many interactions are involved, choreography design must be split up into different phases, each addressing different issues of the model as described in [47] and [48]. This approach can also be evaluated in our work.

XPDL is another standard for executing business process models. The WfMC (Workflow Management Coalition) Steering Committee voted on 23 April 2008 to approve version 2.1 of XPDL [49]. This current release includes all new functionality that was accepted by the working group. Also included is new functionality to update the BPMN to version 1.1. As mentioned, BPMN is a visual process notation standard from the OMG, endorsed by WfMC. The BPMN standard defines only the look of how the process definition is displayed on the screen. How those process definitions are stored and interchanged is outside the scope of the BPMN standard. XPDL provides a file format that supports every aspect of the BPMN process definition notation including graphical descriptions of the diagram, as well as executable properties used at run time. With XPDL, a product can define a process definition with full fidelity,



and another product can import and reproduce the same diagram that was sent. XPDL is used today by more than 80 different products to exchange process definitions.

BPEL and XPDL are entirely different yet complimentary standards. BPEL is an "execution language" designed to provide a definition of web services orchestration. It defines only the executable aspects of a process when he is dealing exclusively with web services and XML data. BPEL does not define the graphical diagram, human oriented processes, subprocesses, and many other aspects of a business process, but it is going to be enhanced accordingly. Also XPDL has to be enhanced for the integration of CEP in future XPDL-based ED-BPM platforms.

## 5 ED-BPM-Based Enhancements for BP-Execution

Business process execution can be implemented on the basis of so called imperative or declarative implementation languages (see e.g. [4]). The "imperative" style means to implement exactly "how" a process has to be executed while the "declarative" approach describes "what" the process shall do. In our work, we concentrate on the implementation language WS-BPEL as an imperative language. The challenges of Sec. 3 are tried to get realized by appropriate enhancements of WS-BPEL. Our studies will also investigate declarative languages and the application of "constraints" respectively rules that can be inserted in a process e.g. as placeholders instantiated at runtime (see [4]), but this approach is not in the focus of the paper.

### 5.1 Requirements for ED-BPM Execution

As a result of the taxonomy described in Sec. 3, new demands are recognized regarding to the business process execution languages. An event-driven process execution language must provide appropriate language constructs to implement the behavior patterns listed in para. 3.3. Additionally, in order to receive events of interest, the business process execution environment (including the business process execution language) must allow *subscribing and unsubscribing* to various events. To support business processes as event producers, a process execution language should provide corresponding constructs for explicit *event reporting*.

Although in many cases it is better to decouple the complex event definition from the process specification. In some cases it would be beneficial to incorporate event patterns into the business process definition. Therefore, event-driven process execution languages should allow incorporating event pattern definitions into the process definition. Different programming languages provide different levels of abstraction [50]. Depending on the abstraction provided by the language, different languages suit better or worse for solving different problems in different domains. The right abstraction for event-driven process definition languages should express aspects of event-driven behavior regarding to the process workflow.

### 5.2 WS-BPEL as ED-BP Execution Language

WS-BPEL [51] is a common language for process specification and execution for SOA. It is also called a "language for business process orchestration based on web services" [52] because the distributed entities participating in BPEL processes are

collaborating by using web service interfaces. In this paper the final version BPEL 2.0 is discussed.

WS-BPEL provides some event-driven behavior capabilities e.g. event handlers, fault and compensation handling mechanisms. However in its standard form it can not completely support event-driven process execution. Interactions between partners in BPEL are peer-to-peer by their nature – even asynchronous ones [51]. The event in BPEL is, in fact, an asynchronous request issued by one of the collaborating parties. The binding between physical entities is performed based on partner link definitions; thus no subscription mechanism is available. There is no construct for publishing or broadcasting a message over the network without targeting any specific consumer in the standard language. Hence no event reporting mechanism is available. The use of events and event patterns in a business process is impossible with BPEL because of the lack of appropriate constructs and mechanisms as well as a semantic mismatch of the event concept. Last but not least, the level of abstraction provided by the WS-BPEL language corresponds to the aspects of web services orchestration rather than event-driven behavior.

Although WS-BPEL in its standard form can not execute event-driven processes, developing a completely different execution language for ED-BPM would be problematic because the adoption costs for existing enterprises that currently use WS-BPEL might be too high.

### 5.3 Enhancing WS-BPEL for ED-BPM

The following explains an approach of enhancing WS-BPEL for supporting event-driven business processes.

*Event Subscription:* We introduced a new element called *eventSubscriptions*, under a *scope* or a *process* element. This element contains a list of subscriptions to be activated before starting the first activity enclosed in the scope. Each *subscription* element specifies the event source, expiration period and an optional filter expression to narrow the stream of the incoming events down to those of interest.

*Event Reporting:* We introduce the *reportEvent* extension activity in order to allow a business process to report events. It explicitly reports about events that occur at the “business level”. The *reportEvent* element specifies the qualified name of the reported event, as well as how the event contents should be built. The event message can be composed based on a process variable, a part of a variable, or an expression involving multiple variables.

*Event Patterns:* The *eventPattern* element declares an event pattern inside the process definition. The WS-BPEL standard allows using pluggable expression language in expressions. We adopt this approach in regard to the expression languages for event patterns or event filter expressions (used in the *eventSubscription* element).

The *expressionLanguage* attribute specifies the expression language used to define the pattern. The *pattern-expr* element specifies the pattern expression using the corresponding expression language. The list of (optional) *to-spec* elements allows initializing variables with the values derived from event instances that made the pattern.

```

<edbpel:eventPattern expressionLanguage="anyURI">
  pattern-expr
  to-spec*
</edbpel:eventPattern>

```

*Other enhancements:* We further use the enhancements introduced above to extend other elements of BPEL. All the message-handling activities (e.g. receive, pick) as well as event handlers are extended to allow the specification of an event or event pattern instead of an incoming request to serve as a trigger for the activity.

```

<onEvent>
  <edbpel:eventPattern
    expressionLanguage="http://example.com/evtPatternLanguage">
    <all>
      <event name="et:TradingOpportunity" alias="to" />
      <where>to.instrument.name == "Example"</where>
    </all>
    <to variable="TradOptInstrValue">
      <query queryLanguage=
        "http://example.com/evtPatternLanguage">
        to.instr.value
      </query>
    </to>
  </edbpel:eventPattern>
  <scope ...>...</scope>
</onEvent>

```

#### 5.4 Implementing the ‘Fraud Management’ Use Case

The use case in para. 3.3 serves as basis for transforming it into a set of executable processes implemented in WS-BPEL enhanced as described above. Details of the implementation for all of the patterns of the event-driven behavior shown in the scenario are provided in our future work. Instead, we demonstrate five patterns as a short illustration, leaving the rest for a separate discussion.

##### *Pattern 1: Stop one or more running processes upon event*

This pattern implements item 5 of para. 3.3. This could be achieved by using the fault handling mechanism available in WS-BPEL [51]. In order to activate the corresponding fault handler we can use the event handler that is activated by a complex event occurrence. The event handler throws a fault that stops all the currently running activities. A strong benefit of using a loosely coupled event-driven solution is the ability to stop any number of running processes upon a single event: all the processes subscribed to this event will be affected.

##### *Pattern 2: Start a new process instance upon event*

This pattern implements items 6, 7, and 9 of para. 3.3. Standard WS-BPEL allows starting a new process instance upon an incoming message (via the *receive* element). We extend the *receive* element with the ability to specify an event name or an event pattern instead of a message. In order to activate a new process instance the enhanced *receive* version could be used.

*Pattern 3: Activate a task or a process in the absence of the expected event within some time period*

This pattern implements item 8 of para. 3.3. The absence of some event is an event pattern by itself. It can be specified in an event processing language and detected by an event processing system. Therefore, this pattern is a special case of the above one (start a new process upon event).

*Pattern 4: Suspend a process and resume upon event*

This pattern can be used to suspend processes until a certain event occurs as described in items 7 and 8 of para. 3.3. This pattern can be implemented using one of the synchronous BPEL constructs, i.e. *receive* or *pick*. For this specific use case ('fraud') *pick* is preferable because it allows handling time out situations, that is what happens if the 'false positive' decision is never made.

*Pattern 5: Start a process modification/adaptation upon event*

This pattern solves case 10 of para 3.3. The pattern can be implemented using the enhanced version of the event handler mechanism. The event handler is activated upon an instance of the specified trigger event (e.g. 'Too Many False Positives'). The enclosed activity can modify the BPEL variable that participates in the 'fraud detection' pattern calculation, or submits an update request to the event processing system. If the case requires modifying the workflow, the workflow adjustment could be considered as a separate process that is started upon this event. As soon as the new process version is available, it will be activated.

## 6 Conclusions

This paper provided an impression of the components and requirements needed for ED-BPM which is elaborated by providing a reference model and reference architecture. After analyzing the presented model and architecture it followed the identification that current business process execution languages need extensions to provide the required capabilities as disclosed in an exemplified financial use case for event-driven behavior. This example serves on the one hand to prove the concept of the (Re-) Active Internet of the Future. On the other hand the BPEL enhancements as an example of an execution language provide a first step into the direction of the (Re-) Active Internet, whereas many other issues have to be solved in processing these events. Past, current and future movements within the according standards and their possible influences show a broad interest both on the modeling and the execution side. The ED-BPM reference model, the enhanced NEXOF-RA, the necessary involved extensions of business process modeling notations and business process execution languages serve as the basis for integrating complex events providing the ability for influencing and dynamically changing business processes. Yet the similarities and diversities of different application domains need to be identified and investigated in the future.

## References

- [1] European Future Internet Portal, <http://www.future-internet.eu>
- [2] The Future Of Internet, <http://www.fi-bleed.eu>
- [3] Li, M.-S., Crave, S., Müller, J.P., Willmott, S.: The Internet of Services: Vision, Scope and Issues. In: eChallenges e-2008 Paper No. 143; eChallenges e-2008 Paper No. 143 (2008), <http://ssrn.com/abstract=1293722>
- [4] Mulyar, N.A., Schonenberg, M.H., Mans, R.S., Russell, N.C., van der Aalst, W.M.P.: Towards a taxonomy of process flexibility (extended version). BPM Center Report No. BPM-07-11, Brisbane/Eindhoven (2007)
- [5] Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Professional, Reading (2002)
- [6] Ammon, R.v., Emmersberger, C., Springer, F., C., W.: Event-Driven Business Process Management and its Practical Application Taking the Example of DHL. In: Future Internet Symposium, Vienna (2008), [http://icep-fis08.fzi.de/papers/iCEP08\\_8.pdf](http://icep-fis08.fzi.de/papers/iCEP08_8.pdf)
- [7] Ammon, R. v., Emmersberger, C., Springer, F.: "Event-Driven Business Process Management" - Eine neue Technologie und erste Projekte am Beispiel der DHL, OBJEKTSpektrum 06/2008 SIGS-DATACOM (2008), [http://www.sigs.de/publications/os/2008/06/ammon\\_OS\\_06\\_08.pdf](http://www.sigs.de/publications/os/2008/06/ammon_OS_06_08.pdf)
- [8] Ammon, R.v.: Event-Driven Business Process Management. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems. Springer, Heidelberg (2008)
- [9] Ammon, R.: Domain Specific Reference Models for Event Patterns – for Faster Developing of Business Activity Monitoring Applications. In: VIPSI 2007, Lake Bled, Slovenia, October 8-11 (2007)
- [10] Albek, E., Bax, E., Billock, G., Chandy, K.M., Swett, I.: An Event Processing Language (EPL) for Building Sense and Respond Applications. In: Proceedings of the 19th IEEE international Parallel and Distributed Processing Symposium (Ipdp's'05) - Workshop 2, IPDPS, April 04 - 08, vol. 03. IEEE Computer Society, Washington (2005)
- [11] Brandl, H.-M., Guschakowski, D.: Complex Event Processing in the context of Business Activity Monitoring. An evaluation of different approaches and tools taking the example of the Next Generation easyCredit. Diploma thesis, Preworkshop DEBS 2007 (2007), [http://www.citt-online.de/downloads/Diplomarbeit\\_BaGu\\_Final.pdf](http://www.citt-online.de/downloads/Diplomarbeit_BaGu_Final.pdf)
- [12] Networked European Software & Services Initiative, NESSI Brochure (2005), [http://www.nessi-europe.eu/Nessi/Portals/0/Nessi-Repository/Publications/Flyers/2005\\_09\\_NESSI\\_Brochure.pdf](http://www.nessi-europe.eu/Nessi/Portals/0/Nessi-Repository/Publications/Flyers/2005_09_NESSI_Brochure.pdf) (Retrieved February 15, 2009)
- [13] Corte, P., Desideri, D.: NEXOF RA, Definition of an architectural framework & principles (2008), [http://www.nexof-ra.eu/sites/default/files/D7.2\\_Definition\\_of\\_an\\_architectural\\_framework\\_\\_principles.doc](http://www.nexof-ra.eu/sites/default/files/D7.2_Definition_of_an_architectural_framework__principles.doc) (Retrieved February 24, 2009)
- [14] NEXOF RA, NEXOF Reference Architecture (2008), <http://www.nexof-ra.eu/?q=node/1> (Retrieved February 23, 2009)
- [15] Ammon, R.v., Emmersberger, C., Ertlmaier, T., Etzion, O., Paulus, T., Springer, F.: Existing and Future Standards for Event-Driven Business Process Management. In: DEBS 2009, Nashville (2009)

- [16] Heinel, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M.: A comprehensive approach to flexibility in workflow management systems. In: WACC 1999: Proceedings of the international joint conference on Work activities coordination and collaboration, pp. 79–88. ACM, New York (1999)
- [17] Regev, G., Wegmann, A.: A regulation-based view on business process and supporting system flexibility. In: Workshop on Business Process Modeling, Design and Support (BPMDS 2005), Proceedings of CAiSE 2005 Workshops, Porto, pp. 35–42 (2005)
- [18] Reijers, H.A.: Workflow flexibility: The forlorn promise. In: 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), pp. 271–272. IEEE Computer Society, Manchester (2006)
- [19] Daoudi, F., Nurcan, S.: A benchmarking framework for methods to design flexible business processes. *Software Process Improvement and Practice*, Trier 12, 51–63 (2007)
- [20] Snowdon, R.A., Warboys, B.C., Greenwood, R.M., Holland, C.P., Kawalek, J.P., Shaw, D.R.: On the architecture and form of flexible process support. *Software Process Improvement and Practice*, Trier 12, 21–34 (2007)
- [21] Regev, G., Bider, I., Wegmann, A.: Defining business process flexibility with the help of invariants. *Software Process Improvement and Practice*, Trier 12, 65–79 (2007)
- [22] Regev, G., Soffer, P., Schmidt, R.: Taxonomy of flexibility in business processes. In: Proceedings of the 7th Workshop on Business Process Modeling, Development and Support (BPMDS 2006), Luxembourg (2006)
- [23] Regev, G., Wegmann, A.: Business process flexibility: Weick’s organizational theory to the rescue. In: Proceedings of the 7th Workshop on Business Process Modeling, Development and Support (BPMDS 2006), Luxembourg (2006)
- [24] Bider, I.: Masking flexibility behind rigidity: Notes on how much flexibility people are willing to cope with. In: Proceedings of the CAiSE 2005 Workshop, Porto, pp. 7–18 (2005)
- [25] Soffer, P.: On the Notion of Flexibility in Business Processes. In: Proceedings of the CAiSE 2005 Workshop, Porto, pp. 35–42 (2005)
- [26] S-Cube, <http://www.s-cube-network.eu/>
- [27] Kazhamiak, R.: Adaptation and Monitoring in S-Cube: Global Vision and Roadmap. In: Proceedings of the Mona+ Workshop, ServiceWave 2008, Madrid, pp. 67–76 (2008), <http://www.s-cube-network.eu/news/s-cube-mona-proceedings-published>
- [28] Ammon, R.v., Etzion, O., Paschke, A., Stojanovic, N.: Event-Driven Business Process Management. In: ED-BPM Workshop, ServiceWave 2008, Madrid (2008), <http://www.nessi-europe.com/Nessi/AdminP/ArchivedPages/ServiceWave2008WorkshopBPM/tabid/468/Default.aspx> (Retrieved September 20, 2009)
- [29] 1st European Conference on Software Services and SOKU technologies, <http://www.eu-ecss.eu/contents/conference/exhibitors-ssoku09/view>
- [30] Liu, D., Pedrinaci, C., Domingue, J.: Semantic Enabled Complex Event Language for Business Process Monitoring. In: Proceedings of the 4<sup>th</sup> SBMN-Workshop, collocated with ESWC 2009, Crete, Greece, May 31 - June 4 (2009)
- [31] Ammon, R.v., et al.: 1st European Conference on Software Services and SOKU technologies, SSOKU 2009, Brussels (2009), [http://www.citt-online.de/downloads/Flyer\\_SSOKU\\_BankingFraud.ppt](http://www.citt-online.de/downloads/Flyer_SSOKU_BankingFraud.ppt)
- [32] Widder, A., Ammon, R.v., Schaeffer, P., Wolff, C.: In: Proceedings of the DEBS 2007, Rome. ACM International Conference Proceeding Series, vol. 233 (2007)
- [33] Widder, A., Ammon, R.v., Hagemann, G., Schönefeld, D.: An Approach for Automatic Fraud Detection in the Insurance Domain. In: AAAI 2009 Spring Symposia / Intelligent Event Processing, Stanford, March 23-25 (2009)

- [34] Paulus, T., Zacharias, R., Scheider, H., Wolff, C.: Fraud Management and Notification Event Architecture for Retail (NEAR) Standard and First Experiences with Point of Sales/Point of Services at Wincor Nixdorf. In: Service Wave 2008, Madrid (2008)
- [35] Paulus, T.: Tutorial on Event Processing Use Cases, ED-BPM in the Retail Domain - Taking the Example of Fraud Management. In: 3<sup>rd</sup> ACM International Conference on Distributed Event-Based Systems 2009, Nashville (2009)
- [36] OMG, Business Process Management Initiative, <http://www.bpmn.org>
- [37] Allweyer, T.: BPMN – Business Process Modeling Notation. Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand GmbH (2009) ISBN 978-3-8370-7004-0
- [38] Decker, G., Grosskopf, A., Barros, A.: A Graphical Notation for Modeling Complex Events in Business Processes. In: 11th IEEE International, Enterprise Distributed Object Computing Conference, EDOC 2007, Annapolis, October 15-19, p. 27 (2007)
- [39] Object Management Group, Event Metamodel and Profile (EMP) Request For Proposal (2008), <http://doc.omg.org/ad/2008-9-15> (Retrieved February 24, 2009)
- [40] Fleischmann, A.: Distributed Systems – Software Design and Implementation. Springer, Berlin (1994)
- [41] Schmidt, W., Fleischmann, A., Gilbert, O.: Subjektorientiertes Geschäftsprozessmanagement. Praxis der Wirtschaftsinformatik, HMD, Heft 299, dpunkt.verlag, p. 52, Heidelberg (2009)
- [42] 2<sup>nd</sup> ED-BPM Workshop at the Service Wave 2009, November 24-27, Stockholm (2009), <http://www.icsoc.org/>
- [43] Gartner, S.J.: Getting Painted in a Corner by Structured Business Processes (2009), [http://blogs.gartner.com/jim\\_sinur/2009/08/06/getting-painted-in-a-corner-by-structured-business-processes/](http://blogs.gartner.com/jim_sinur/2009/08/06/getting-painted-in-a-corner-by-structured-business-processes/) (Retrieved September 20, 2009)
- [44] OASIS, Web Services Business Process Execution Language (WSBPEL) Technical Committee, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)
- [45] OASIS, WS-BPEL Extension for People (BPEL4People) Technical Committee, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=bpel4people](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bpel4people)
- [46] Decker, G., Kopp, O., Leymann, F., Pfitzner, K., Weske, M.: Modeling Service Choreographies using BPMN and BPEL4Chor. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 79–93. Springer, Heidelberg (2008)
- [47] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Berlin (2007) ISBN 978-3540735212
- [48] Barros, A., Decker, G., Dumas, M.: Multi-staged and Multi-viewpoint Service Choreography Modeling. Technical Report 4668, Queensland University of Technology, Brisbane, Australia (2006)
- [49] XPD L Support and Resources, <http://www.wfmc.org/xpdl.html>
- [50] Abelson, H., Sussman, G., Sussman, J.: Structure and Interpretation of Computer Programs, 2nd edn. The MIT Press, Cambridge (1996)
- [51] OASIS, Web Services Business Process Execution Language Version 2.0, Committee Draft (January 25, 2007) <http://docs.oasis-open.org/wsbpel/2.0/> (Retrieved February 14, 2009)
- [52] OASIS, Web Services Business Process Execution Language Version 2.0 Primer (May 9, 2007) <http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.pdf> (Retrieved February 14, 2009)

# Towards Business-Oriented Monitoring and Adaptation of Distributed Service-Based Applications from a Process Owner's Viewpoint

Krešimir Vidačković, Nico Weiner, Holger Kett, and Thomas Renner

Fraunhofer IAO, Competence Center Electronic Business,  
Nobelstr. 12, 70569 Stuttgart, Germany  
{kresimir.vidackovic,nico.weiner,holger.kett,  
thomas.renner}@iao.fraunhofer.de

**Abstract.** Dynamically changing economic environments require distributed Service-Based Applications (SBAs) to be highly flexible and reactive, so that the utilization of monitoring and adaptation functionalities becomes imperative. While approaches for monitoring functional and non-functional properties from the operational environment have gained a certain degree of maturity, there is still a lack of reflecting business-related aspects. This paper introduces the vision of a generic monitoring and adaptation framework focusing on the interactions between different abstraction layers of distributed SBAs. Starting from the business model perspective, strategic decisions are specified by the business model design in order to constitute the scope for possible operational adaptations at the business process, service and resource layers. Additionally, the monitoring of technical and business-related aspects affects not only the operational layers, but also the business model layer in the long-term.

**Keywords:** Service-Based Applications, Cross-Layer Monitoring, Business Model Design.

## 1 Motivation

With distributed Service-Based Applications (SBAs) encountering highly dynamic environments, process owners, who integrate external and internal services into their business processes, must be aware of both, the current status of the entire business process and the environmental changes forcing them to react in an adequate way. This leads to the necessity for utilizing suitable monitoring and adaptation functionalities in order to obtain a flexible and reactive SBA.

Several research approaches have addressed this topic in the last few years, e.g. [1, 2, 3, 4] for monitoring and [5, 6, 7] for adaptation, and the current movement is to combine existing frameworks in order to benefit from their particular strengths [8]. However, even if they are reaching a high maturity, our feeling is that considering only the operational environment does not go far enough. We believe that the next step should be an extension to the business environment.



In one of our use cases with industrial partners from the insurance domain, the claims settlement process of an insurance company is automated and composed by several internal and external services. These are on the one hand human tasks with a connection to the IT infrastructure and on the other hand web services from different providers. Most of the integrated services are substitutable, so that dynamic adaptations can lead to optimizations. The customer satisfaction as well as financial aspects play an important role in this scenario and therefore need to be reflected in particular.

From the viewpoint of the process owner, in our scenario the insurance company, an SBA is built on four abstraction layers: business model, business process, service and resource layer. Resources, like human resources or the IT infrastructure including runtime environments, are allocated to deliver internal services. The latter are situated together with external services and compositions of both of them on the service layer. The business process layer holds abstract representations of the company's core processes which partly or entirely consist of the services or service compositions from the layer below. Focusing merely on these three operational layers, what most of the current monitoring and adaptation approaches do, lacks the important strategic perspective of the business model. However, especially when most of the services within the business process are substitutable, as it is the case in our scenario, an overall optimization can only be accomplished, if the business model perspective is considered as well.

Taking this into account, our idea is to develop a generic framework where monitoring and adaptation of distributed SBAs is accomplished in a cross-layer manner with interactions between the operational layers of the SBA and its business model. Our approach has its seeds in the research project Theus/TEXO funded by the German Federal Ministry of Economy and Technology (<http://theseus-programm.de/en-us/theseus-application-scenarios/texo>).

## 2 Business Model Design

The term "business model" has been used in several different meanings in the last couple of years. On the one hand, practitioners use the term to describe one or more key components of a real company, without really defining or explaining the meaning of the term. On the other hand, deductive research shapes the scientific landscape on business models since the turn of the millennium and the new economy. In research, the term "business modeling" is sometimes used to describe the business application environment within a company [9] or even to describe the intra-company business processes. Actually, most researchers describe a company's business model as an ontology composed of abstract components like value proposition, partners and customers, financial aspects, internal processes and corresponding relations [10, 11, 12]. For traditional business model research, it is crucial to understand the business logic, not to create a new one. More recent research streams called "business model innovation" or "business model dynamics" consider both, understanding the business logic and utilizing this view to create new opportunities in the market.

To define a business model, we adopt the generic definition of Osterwalder and Pigneur which is used by many other authors as well [13]. According to them, a business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific company. Thus, a business model is neither (only) a process model nor (only) a set of key components of a company. An important part is the formalization of relationships between the components and how environmental and internal changes cause an adaptation of those components. Environmental changes for instance include business strategies (e.g. competitive positioning), market changes (e.g. new competitors, new technologies) and regulations (e.g. new laws). Internal changes include e.g. process and service adaptations and pricing model changes. We assume the business model as a mediator layer between the strategic perspectives and the process view of a company [14]. To further describe this simplified view on a company's business logic, we derive domains shaping a business model [11]:

- The finance domain includes costs, revenues, profits and pricing strategies. It is directly influenced by the realized business processes to create a value.
- It leads to a service and a product respectively addressing specific customers. This domain defines all the values related to the company's environment.
- The people domain describes the company's partners, customers and the roles they represent.
- The assets domain and the workflow domain describe the value creation and which resources are used by the various business processes.
- The rules domain contains organizational aspects like the company's policy.

Additionally, the strategy of a company influences its business model. Most authors in business model research regard the strategy as an external influence [12, 14]. The strategy usually contains e.g. a market (or competitor) strategy



**Fig. 1.** Business model elements following Kett et al. [11]

and a customer strategy. Figure 1 depicts a derived high level concept of the business model elements [11].

### 3 Business-Oriented Monitoring and Adaptation

After clarification of the business model layer as the strategic perspective and the underlying business process, service and resource layers as the operational perspective, the question arises how to bridge the gap between these worlds. Following the idea of a combined business model and process monitoring and adaptation, we introduce the vision of a generic framework comprising these layers in a top-down approach as illustrated in figure 2.

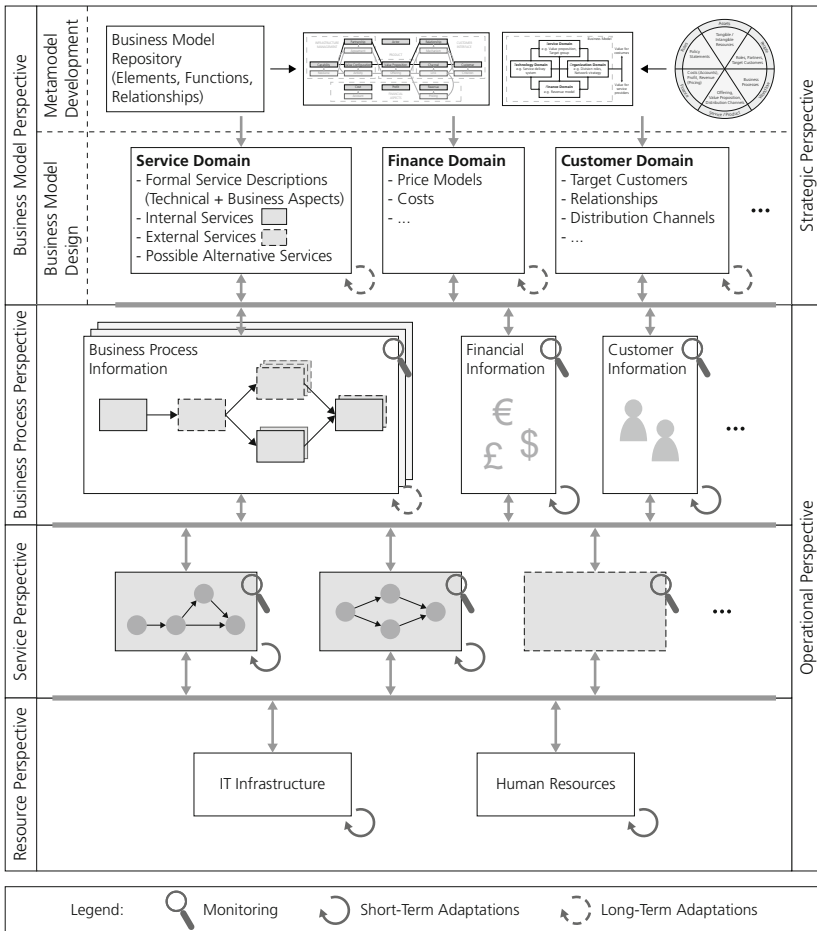


Fig. 2. Generic, cross-layer monitoring and adaptation framework

Monitoring and adaptation functionalities should be applied on all of these abstraction layers with several interactions between them which is explained in more detail in the next sections. The starting point is the business model perspective where strategic decisions are specified by the business model design.

### 3.1 Business Model Perspective

Although business model ontologies have gained acceptance to represent business models over the last years, almost no software tool exists to support the graphical modeling of a service-centric business model which is directly related to the underlying business processes and incorporates process performance measures. Related approaches in terms of business modeling include business process modeling (e.g. IDEF, ARIS etc.) and meta-modeling approaches of a business (e.g. Porter's Value Chain etc.). Performance aspects are realized by tools like the Balanced Scorecard by Kaplan and Norton and Business Activity Monitoring approaches. There is only a few research which combines these approaches in order to provide additional opportunities like business model adaption. Samavi et al. for example describe an approach which relates the strategic goals of a company with the operational alignment of the business model [14].

More important, there is a lack of tools providing the feasibility to execute automatic adaptations of business processes on the basis of an event-based monitoring with the help of a pre-defined business model instance. Such tools would extract the scope for an adaptation out of the business model elements. For example, for pricing adaptations of the business process and its integrated services, the finance domain would provide a possible set of pricing strategies to be used as the adaptation logic. Partner adaptations would extract the set of possible partners involved in a process out of the people domain. This idea requires the modeling of a business model via the different domains we introduced in the preceding section. There is no specific process knowledge required in this stage. Those tools are intended to be applied by managers and business analysts to review and steer the market strategy and to get a detailed understanding of the company's environment (customers, competitors). Another important application can be venture capitalists, in order to understand and evaluate the business logic of a new start-up and to understand critical success factors [15]. A well-known approach is introduced by Gordijn utilizing value streams [16]. The disadvantage here is the lack of integrating measurements and indicators about the status of the business model. Therefore, the model is not dynamic and cannot be used permanently. Actually, this methodology does not aim at supporting an on-going business, but specializes on brainstorming activities for the evaluation of business model ideas.

For a tool-supported business model design, we are going to develop a more detailed ontology based on [10] and [11]. This will result in a metamodel development stage shown in figure 2 on the top layer. The metamodel consists of business model components, relations and measurement indicators in order to provide a generic toolset for construction of a business model. The business

model design layer uses any metamodel component from this business model repository to provide a design environment for business model visualization.

The business model perspective is subsequently connected to the underlying business process perspective constituting scopes for adaptation functionalities and showing information on business performance. These scopes correspond to the modeled service domain, the finance domain, the customer domain etc.

As an example from the service domain, it is possible to model a set of alternative partners for the execution of an external service which is included in the business process. Partner-related information for instance can comprise a quality rating or former partner relationships. For a manager or business analyst, this information is crucial in order to decide which partner to choose for an external service within the business process. The next step is to link all the chosen partners and their services to the related business process activities and to decide which of these partners and their services are used initially for process execution. The other linked partners and their services constitute alternatives and currently stay "on hold". This information is stored in a formal service description containing business and technical aspects.

An example from the finance domain is the modeling of different price models for the offering of the business process to the target customer. The latter as well as e.g. customer relationships and possible distribution channels are modeled in the customer domain of the business model design.

Adaptations on the layers below are only possible within these scopes, i.e. if exchanging an external service by another one or switching to another price model would lead to an optimization, this can only be accomplished with respect to the priorly defined business model.

### 3.2 Business Process, Service and Resource Perspective

From the operational perspective, the strategic decisions from the business model design represent restrictions for potential adaptations. Monitoring on the business process layer is performed not only on functional and non-functional information gathered from the business process itself, but also on business information as e.g. financial or customer-related aspects. Examples include the monitoring of costs and revenues in case of time- or usage-dependent pricing of external services as well as of customer satisfaction and feedback.

The collected monitoring information is used by the process owner on the one hand for visualization purposes and on the other hand for automated short-term adaptations which can be conducted on the following layers:

- Business process layer: Adaptations regarding business-related information, e.g. switching to another price model in the finance domain or changing the advertising campaign in the customer domain. If such an adaptation leads to an optimized business process based on a specific criterion, e.g. costs or customer satisfaction, there is no need for an adaptation on the service layer, otherwise it could be required as well.

- Service layer: Adaptations of the service composition, i.e. recomposition or substitution of an external or internal service according to the alternatives specified in the service domain of the business model which therefore represents a restriction for the adaptation functionality. This could also imply that an internal service is replaced by an external service or vice versa. If this is not enough for optimization, adaptations on the resource layer could be executed as well.
- Resource layer: Adaptations of the resources, e.g. automatic reallocation of IT infrastructure or reorganisation of human tasks in internal services.

Furthermore, it is also possible that the monitored information leads to long-term or manual adaptations which are situated on the following layers:

- Business model layer: Adaptations of the business model from a strategic point of view, e.g. searching for new partners to deliver external services, renewing the price model, redefining the target customer or distribution channel.
- Business process layer: Redesign of the business process, i.e. fundamental modifications of the business process or business process activities.

## 4 Conclusion

As discussed in the last chapter, our vision of a generic monitoring and adaptation framework for SBAs from the process owner's viewpoint is a cross-layer approach taking into account not only the operational perspective, but also the strategic perspective through the consideration of the business model. As there are inevitable interdependencies between the different layers, all of them need to be addressed by the framework (neglecting the resource perspective), and their interactions are an integral part of our vision.

For the monitoring and adaptation of the operational environment, the existing state-of-the-art approaches are suitable, as they have already gained a high degree of maturity. However, they have to be extended with the following capabilities in order to meet our requirements:

- Monitoring of business-related aspects, particularly financial and customer information, i.e. usage of additional information sources.
- Better integration of monitoring functionalities on different abstraction layers.

In addition, more research needs to be conducted in the following areas:

- Formal service description which comprises technical as well as business-related aspects and the alternatives given for each specified service.
- Modeling of business model components in such a manner that the information can be extracted in an automated way in order to be usable for specifying the scope for adaptation functionalities.
- Support for the process owner in modeling the monitoring and the adaptation model on the different abstraction layers with an appropriate design tool.

In our opinion, the research community in the area of monitoring and adaptation of SBAs should move more towards the business aspects and integrate them into their approaches. From an industrial viewpoint, this is a crucial step for a comprehensive business process optimization.

## 5 Roadmap

In order to turn our vision of a generic, cross-layer monitoring and adaptation framework into reality, several research activities are required and will be executed in the near future.

Firstly, we are going to define models for the business model design which derive from our metamodel development and can be used as a reference on the business process layer. This has to be done for all of the relevant business model domains, e.g. finance domain, customer domain, etc. In the service domain, this also includes the evaluation of existing formal service descriptions regarding the ability to specify technical and business-related aspects as well as to integrate partners in case of external services and to consider alternative services when needed. First research approaches in this field were already introduced, e.g. in [17] and [18]. If they are not feasible for our requirements, a new formal service description has to be developed for our purpose.

The next step is to evaluate monitoring and adaptation approaches for the operational perspective. The most relevant criterion for complying with our vision is the possibility to be extended on the business environment. Another important issue is the seamless integration of the monitoring functionality into the existing IT infrastructure of the process owner. In this context, we have developed a concept for consuming the monitoring functionality as a service from a specialized monitoring service provider [19].

In addition, a design tool has to be implemented for supporting the process owner in modeling of the business model design as well as the monitoring and adaptation model.

Finally, we are going to abstract from our use case scenario in the insurance domain in order to gain a more generic approach.

## Acknowledgement

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07017". The authors take the responsibility for the contents.

## References

- [1] Baresi, L., Guinea, S.: Towards Dynamic Monitoring of WS-BPEL Processes. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 269–282. Springer, Heidelberg (2005)

- [2] Pistore, M., Traverso, P.: Assumption-Based Composition and Monitoring of Web Services. In: Baresi, L., Nitto, E.D. (eds.) *Test and Analysis of Web Services*, pp. 307–335. Springer, Heidelberg (2007)
- [3] Spanoudakis, G., Mahbub, K.: Non-Intrusive Monitoring of Service Based Systems. *International Journal of Cooperative Information Systems* 15(3), 325–358 (2006)
- [4] Momm, C., Gebhart, M., Abeck, S.: A Model-Driven Approach for Monitoring Business Performance in Web Service Compositions. In: *Proceedings of the Fourth International Conference on Internet and Web Applications and Services (ICIW)*, pp. 343–350. IEEE Computer Society, Washington (2009)
- [5] Colombo, M., Nitto, E.D., Mauri, M.: SCENE: A Service Composition Execution Environment Supporting Dynamic Changes Disciplined Through Rules. In: Dan, A., Lamersdorf, W. (eds.) *ICSOC 2006. LNCS*, vol. 4294, pp. 191–202. Springer, Heidelberg (2006)
- [6] Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A Framework for Executing Adaptive Web-Service Processes. *IEEE Software* 24(6), 39–46 (2007)
- [7] Gehlert, A., Heuer, A.: Towards Goal-Driven Self Optimisation of Service Based Applications. In: Mähönen, P., Pohl, K., Priol, T. (eds.) *ServiceWave 2008. LNCS*, vol. 5377, pp. 13–24. Springer, Heidelberg (2008)
- [8] Baresi, L., Guinea, S., Kazhamiak, R., Pistore, M.: An Integrated Approach for the Run-Time Monitoring of BPEL Orchestrations. In: Mähönen, P., Pohl, K., Priol, T. (eds.) *ServiceWave 2008. LNCS*, vol. 5377, pp. 1–12. Springer, Heidelberg (2008)
- [9] Eriksson, H.-E., Penker, M.: *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons, New York (2000)
- [10] Bouwman, H., De Vos, H., Haaker, T.: *Mobile service innovation and business models*. Springer, Heidelberg (2008)
- [11] Kett, H., Scheithauer, G., Weiner, N., Weisbecker, A.: Integrated Service Engineering (ISE) for Service-Ecosystems: An Interdisciplinary Methodology for the Internet of Services. In: Cunningham, P., Cunningham, M. (eds.) *eChallenges e-2009 Conference Proceedings, IIMC International Information Management Corporation Ltd., Dublin* (2009)
- [12] Osterwalder, A.: *The Business Model Ontology: a proposition in a design science approach*. PhD thesis, Université de Lausanne (2004)
- [13] Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying Business Models: Origins, Present, and Future of the Concept. In: *Communications of the Association for Information Systems*, vol. 16, pp. 1–25 (2005)
- [14] Samavi, R., Yu, E., Topaloglou, T.: Strategic Reasoning about business models: a conceptual modeling approach. *Information Systems and E-Business Management* 7(2), 171–198 (2009)
- [15] Weiner, N., Renner, T., Neuhart, A., Kett, H.: Success Factors for Innovative Internet Business Models - Venture Capital Insights. In: *Proceedings of the 2nd ISPIM Innovation Symposium* (2009)
- [16] Gordijn, J.: E-business value modelling using the e<sup>3</sup>-value ontology. In: Currie, W. (ed.) *Value creation from e-business models*, pp. 98–127. Elsevier, Oxford (2004)



- [17] Scheithauer, G., Winkler, M.: A Service Description Framework for Service Ecosystems. In: Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78., Bamberg University (2008)
- [18] Cardoso, J., Winkler, M., Voigt, K.: A Service Description Language for the Internet of Services. In: Alt, R., Fähnrich, K.-P., Franczyk, B. (eds.) Proceedings of the First International Symposium on Services Science ISSS 2009, pp. 229–240. Logos, Berlin (2009)
- [19] Vidackovic, K., Kett, H., Renner, T.: Service Chain Monitoring for the Internet of Services. In: Cunningham, P., Cunningham, M. (eds.) eChallenges e-2009 Conference Proceedings. IIMC International Information Management Corporation Ltd., Dublin (2009)

# Adaptation of Service-Based Applications Based on Process Quality Factor Analysis

Raman Kazhamiakin<sup>1</sup>, Branimir Wetzstein<sup>2</sup>, Dimka Karastoyanova<sup>2</sup>, Marco Pistore<sup>1</sup>, and Frank Leymann<sup>2</sup>

<sup>1</sup> FBK-Irst, via Sommarive 18, 38100 Trento, Italy  
{raman,pistore}@fbk.eu

<sup>2</sup> Institute of Architecture of Application Systems, University of Stuttgart, Germany  
{karastoyanova,leymann,wetzstein}@iaas.uni-stuttgart.de

**Abstract.** When service-based applications implement business processes, it is important to monitor their performance in terms of Key Performance Indicators (KPIs). If monitoring results show that the KPIs do not reach target values, the influential factors have to be analyzed and corresponding adaptation actions have to be taken. In this paper we present a novel adaptation approach for service-based applications (SBAs) based on a process quality factor analysis. This approach uses decision trees for showing the dependencies of KPIs on process quality factors from different functional levels of an SBA. We extend the monitoring and analysis approach and show how the analysis results may be used to come up with an adaptation strategy leading to an SBA that satisfies KPI values.

## 1 Introduction

In recent years extensive attention has been paid to devising and improving the concepts and infrastructures for service-based applications (SBAs) [1]. SBAs can be viewed in terms of three functional layers, namely (i) business processes, (ii) service compositions that implement these business processes, and (iii) services and service infrastructure. A major concern for enterprises is to ensure the quality of their SBA-based business processes. Thereby, process quality goals are specified in terms of Key Performance Indicators (e.g., order fulfillment time), i.e. key process metrics that contain target values which are to be achieved in a certain period. KPIs of business processes that are implemented in terms of SBAs are typically monitored using business activity monitoring technology. If monitoring results show that KPIs do not meet target values, further *process quality factor analysis* is needed to find out which of the lower level process metrics (e.g., duration of process activities, type and amount of ordered products etc.) or QoS metrics (e.g., availability of IT infrastructure) mostly influence KPI target violations.

After influential factors of KPI violations are identified, the goal is to perform *process adaptation* in order to prevent KPI violations for future process instances or even for the running instances. Thereby, several challenges arise. Firstly, one has to choose appropriate adaptation actions for each influential factor identified (e.g., selection of a faster delivery service in order to decrease deliver time). Secondly, one has to take into account that an adaptation action can have positive effect on one metric

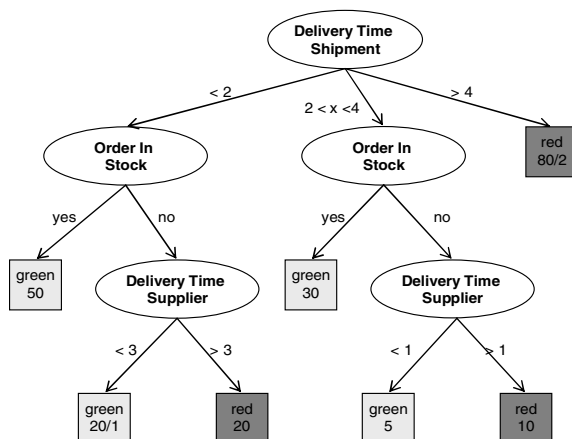
but negative effect on others (e.g., a faster delivery time normally involves higher costs). Thus, when adapting the process one has to choose a set of adaptation actions which improve the influential factors as shown in the analysis and take into account their effects.

In order to deal with this adaptation problem, we extend previous work described in [2] which uses decision trees for process quality factor analysis. Based on this work, in this paper, we show how to extract a set of adaptation requirements from the decision tree and find an adaptation strategy consisting of a set of adaptation actions which takes into account both positive and negative effects of adaptation actions on metrics. We discuss limitations of our approach so far and show several possibilities for extending the approach in future work.

## 2 Background and Motivation

In this section we present the motivation for our approach and a scenario which we use in the following sections for explaining our concepts based on examples. This scenario has already been used in [2] for experimental evaluation of process quality factor analysis. The scenario consists of a purchase order process implemented by a reseller which offers products to its customers and interacts with its suppliers, a banking service, and a shipment service for processing the order. The customer sends a purchase order request with details about the required products and needed amounts to the reseller. The latter checks whether all products are available in the warehouse. If some products are not in stock, they are ordered from suppliers. When all products are in place, the warehouse packages the products and hands them over to the shipment service, which delivers the order to the customer, and finally notifies the reseller about the shipment. For measuring the performance of its business process, the reseller defines a set of Key Performance Indicators (KPIs). A typical KPI for the reseller in our scenario is order fulfillment lead time [3], which measures the number of days from order receipt to the delivery of the ordered products at the customer. A KPI is a key metric (with either technical or business meaning) with target values which are to be achieved in a certain analysis period (e.g., order fulfillment lead time < 5 days). After specifying a set of KPIs with target values, they have to be measured based on executed process instances. If the measurement shows an unsatisfying result, i.e. the KPI targets are violated, the reseller wants to improve its process, for instance, by using process adaptation.

Due to the fact that KPIs are complex characteristics that rely upon a wide range of factors originating from different functional levels, adaptation of underlying SBAs is not a straightforward approach. In our scenario the KPI may be influenced by many factors (which have to be measured both on process level (process performance metrics) and service infrastructure level (QoS metrics): duration of sub-processes and activities, response time and availability of used services, ordered products and their properties such as number of ordered items, product type and size, cost of delivery service, availability of IT infrastructure etc. All those factors and a combination of those can lead to late delivery. Thus, the first step needed is to perform a process quality factor analysis and find out the influential factors for KPI violations.



**Fig. 1.** An Example Dependency Tree

Our approach to quality factor analysis is based on machine learning techniques, more specifically decision tree algorithms [4]. The approach, its assumptions and reasons for using machine learning techniques, have been described in detail in [2]. In the following we will give only a general overview based on an example. The result of this analysis is a decision tree as shown in Fig. 1, called a *dependency tree* as it shows the main quality factors the KPI *depends* on. The tree is generated automatically for a KPI selected by the user. The leaves of the tree show the classification of the KPI, i.e. whether it is satisfied (“green”) or violated (“red”) in relation to its target values, and the number of process instances which led to this path. Note that the classification (satisfied or violated) could be extended towards more than two nominal values or even numerical value ranges (regression trees); this is part of our future work. The other nodes of the tree are the main influential factors (metrics) and the branches contain conditions on those metrics.

In order to improve those factors, different adaptation actions may be considered, for example, replacing a service either dynamically or using a predefined set of services; renegotiating the Service Level Agreements (SLAs) with the corresponding service provider; outsourcing a subprocess or replacing it with a service from an external provider. On infrastructure level, possible adaptations are replacement of IT components with faster ones, clustering for improving availability, upgrading hardware components etc. For a particular situation, different adaptation actions and their combinations may be necessary for improving the same KPI; consistency and non-contradicting actions with respect to the KPI (and perhaps other KPIs) needs to be ensured. This is because an adaptation action can positively affect one influential factor but negatively others. Assume, for example, the selection of a new better performing service which leads to a better response time but negatively affects the cost metric. We call the collection of the adaptation actions that, being enacted in combination, achieve the desired outcome an adaptation strategy.

### 3 Overview of the Approach

In this work we present an approach that allows for adapting service-based applications in order to prevent the violation of KPIs. The overall process is represented in Fig. 2. This approach consists of the following four phases:

- *Quality modeling for analysis and adaptation.* At design time the metrics model and the adaptation actions model are created. In the metrics model, the user specifies the application KPIs, and the quality metrics representing the potential influential factors of KPIs. In the adaptation actions model, the user specifies the available adaptation actions per metric and the effect of those actions on application metrics specified in the metrics model. In particular, this model allows for defining whether an action contributes positively or negatively to a certain quality factor, i.e., whether it improves the value of a metric.

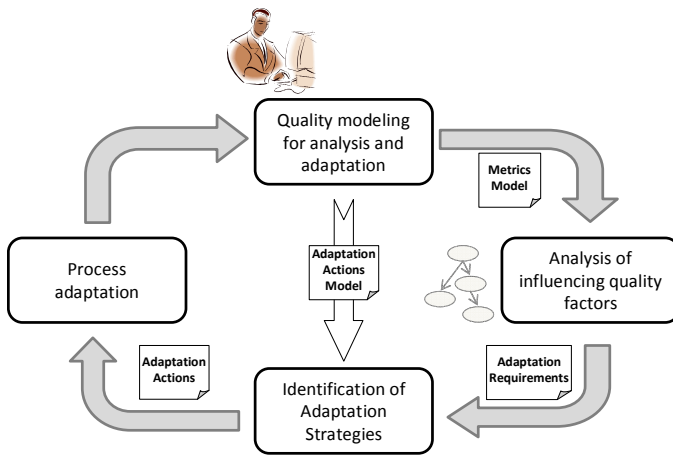


Fig. 2. Quality factor analysis and adaptation

- *Analysis of influential quality factors.* In the second phase, based on the metrics model, the monitoring of KPIs and potential influential quality metrics is performed across the instances of the application; the information is continuously aggregated and updated. Then the metrics related to previous executions of a given application are analyzed in order to identify the reasons, i.e., the influential factors, which lead to the undesired values of the specified KPIs. A dependency tree is generated. In the next step, one identifies those tree paths of application metrics (and their value ranges) that correspond to the “bad” values of the KPI and thus fail the underlying business goal. The result of the analysis characterizes thus those factors of the application that should be improved, i.e., that are subject of adaptation, and how they should be improved (their values) (Section 4.2).
- *Identification and selection of an adaptation strategy.* In the next phase the approach aims to combine, and enact concrete adaptation actions that address the identified requirements as part of a coherent adaptation strategy (Section 4.3). This phase relies on the adaptation action model, where the effect of those

actions on different application metrics is described. It takes into account that an adaptation action contributes positively or negatively to a certain quality factor, i.e., whether it improves the value of a metric. After identification of a set of alternative adaptation strategies, one strategy is selected based on certain criteria.

- *Process adaptation.* The selected adaptation strategy is used for adaptation of the process model or process instance by executing all contained adaptation actions. After adaptation, the existing KPIs and metric definitions might have to be adapted thus closing the cycle.

## 4 Identification of Adaptation Requirements and Strategies

Once the influential factors are highlighted, it is necessary to identify those elements of the application that should be improved, i.e., to identify *adaptation requirements*. Based on those requirements and on the model of adaptation actions associated to the quality properties of the application, possible *adaptation strategies* are identified and triggered.

### 4.1 Model of Adaptation Actions

In order to adapt different elements of the application at all the levels of the application stack, i.e., to enable a holistic adaptation strategy, it is necessary to provide a generalized model of possible adaptation actions. This model should relate different adaptation mechanisms to the quality properties of the application, which are subject of adaptation. In other words, the model should characterize the available adaptation actions to the model of metrics described above. More precisely, the definition of adaptation actions in our approach consists of the following elements:

- *Adaptation mechanism.* This part of the definition characterizes the machinery or a technique used to realize the specified adaptation action. For example, the adaptation action “substitute a service with another service” may be realized by a composed mechanism, which consists of service discovery and binding [5]. Other actions may include (but are not limited to): re-negotiation of quality parameters of the services used in the service composition, re-composition of the underlying service composition or a part of the process, replacement of a subprocess with another subprocess or with a single service (process outsourcing, [6]), or infrastructural reconfiguration.
- *Adaptation effect.* To relate the adaptation action to the system we characterize the former in terms of the effects the action causes on one or another application quality metric. We say that the action has a *positive effect* on the metric if the value of the latter is improved as a *result of the application* of the action. We say that the action has a *negative effect* on the metric if the value of the latter is worsened as a result of the application of the action. Otherwise, we say that the *action does not affect* the metric.

We remark that for the presented approach we abstract away the details on how the adaptation actions are *realized and enacted*. The core part of the model of the adaptation action is how the action relates to the application metrics.

An adaptation action  $a$  is defined as a pair  $\langle M^+, M^- \rangle$  where

- $M^+ \subseteq M$  is a set of metrics, on which the action has a direct positive effect;
- $M^- \subseteq M$  is a set of metrics, on which the action has a direct negative effect.

## 4.2 Identification of Adaptation Requirements

After the dependency tree is generated (as discussed in Section 2), the next step is to identify the adaptation requirements for the application in order to improve the performance of the application. This activity relies on the analysis of the dependency trees of the relevant KPIs of the application. Intuitively, this analysis may be described as follows.

- As a first step it is necessary to understand, which of the violations of the KPI (i.e., which of the “red” blocks) should be prevented. For example, it may be the case that all the possible violations should be avoided. In this case it is necessary to find an adaptation strategy (consisting of possibly several adaptation actions) preventing all of those situations. It is also possible to prevent violations only in selected situations, e.g., the most frequent ones. In this case, the other violation cases are ignored and excluded from the tree. We remark that this decision may be done by the business analysts or even automatically, based on some predefined criteria (e.g., for the cases where number of violations exceeds 10%).
- Second, it is necessary to associate the violations with the influential factors that might help avoiding the violations. This is done by identifying all the metrics in the nodes (and their sub-ranges) on the path from the “red” node to the root of the tree. If some of these metrics is improved such that there are no violations, then the adaptation will be successful. In the above dependency tree in order to improve for the central “red” node, it is enough to improve the metric “Delivery Time Supplier” to the value of  $< 1$ . On the other hand, in order to improve for the rightmost “red” node it is not enough to improve the value of the “Delivery Time Shipper” to the value of  $< 4$ . Indeed, the other violations are still possible and the other metrics should be improved as well. If a running instance is to be adapted and some of the metric values are already known for this instance, then obviously some paths of the tree leading to “red” nodes might be irrelevant for that instance and can be excluded from further consideration.
- The final step takes into account the need to consider all the selected “red” nodes together in order to merge the appropriate actions into a complete set of adaptation requirements.

In order to realize this approach, we rely on the algorithms provide by the decision procedure for the Satisfiability Modulo Theories (SMT [7]). In this problem, the goal is to find solutions for a set of logical constraints (formulas) with respect to combinations of background theories, such as the theory of real or integer numbers, Boolean arithmetic, and even complex data structures. Below we show how the problem of finding adaptation requirements may be expressed in terms of SMT problem, and how the adaptation requirements may be then extracted.

Our goal is to avoid all the paths in the tree that lead to the “red” leaf nodes. That is, the combination of the “metric-range” pairs on the path should not occur. This

combination may be represented as a conjunction of expressions over those metrics, i.e., for a path with  $n$  nodes we built an expression  $(\mu_1, r_1) \wedge \dots \wedge (\mu_n, r_n)$ , where  $(\mu_i, r_i)$  represents an expression over the  $i^{\text{th}}$  metric on the path. For instance, for the central “red” path in the example tree the expression would be as follows:

$$(2 < \text{“Del. Time Shipment”} < 4) \wedge (\text{“Order in Stock”} = \text{yes}) \wedge (\text{“Del. Time Supplier”} > 1)$$

If in case of running instance adaptation, some of these metrics values are known, then the expression can be simplified. If, e.g., “Order in Stock” is false, then the expression is already known to be false and thus this “red” path is already avoided for this instance. If “Order in Stock” is true, then it can simply be removed from the expression, thus simplifying later analysis.

In order to avoid those paths, our goal is to make all those expressions false, i.e., for  $m$  paths, we have to find possible assignments of metric values such that the following formula becomes true<sup>1</sup>:

$$((\mu_{1i}, \neg r_{1i}) \vee \dots \vee (\mu_{ni}, \neg r_{ni})) \wedge \dots \wedge ((\mu_{1m}, \neg r_{1m}) \vee \dots \vee (\mu_{nm}, \neg r_{nm}))$$

It is easy to see that if the formula is satisfied, then neither “red” node is reachable. The result of the analysis is represented as a set of alternatives, each of which contains the list of metrics that should be adapted and their expected ranges. In order to carry out the analysis task we use the MathSAT tool [8], which implements the SMT decision procedure.

More precisely, we define the resulting adaptation requirements as  $R = \{A_1, \dots, A_n\}$ , where  $A_i = \{(\mu_{1i}, r_{1i}), \dots, (\mu_{mi}, r_{mi})\}$  is a set of metric-range pairs that should be achieved in order to address the adaptation needs.

We present the approach using the dependency tree depicted in Fig. 1 (we assume that the goal in the example is to avoid any of possible violations). For the metrics “Delivery Time Shipment” (Sh), “Delivery Time Supplier” (Su), and “Order in Stock” (O), and for the three paths to “red” nodes we construct the following three constraints:

$$- \text{Sh} < 2 \wedge \text{O=no} \wedge \text{Su} > 3; \text{Sh} > 2 \wedge \text{Sh} < 4 \wedge \text{O=no} \wedge \text{Su} > 1; \text{Sh} > 4$$

Based on these clauses, we need to satisfy the following formula:

$$- (\text{Sh} > 2 \vee \text{O=yes} \vee \text{Su} < 3) \wedge (\text{Sh} < 2 \vee \text{Sh} > 4 \vee \text{O=yes} \vee \text{Su} < 1) \wedge (\text{Sh} < 4).$$

The result of the analysis provided by the tool represents the following alternatives:

$$- (2 < \text{Sh} < 4) \text{ and } (\text{Su} < 1), (\text{Sh} < 2) \text{ and } (\text{Su} < 3), \text{ and } (\text{O=yes}) \text{ and } (\text{Sh} < 4)$$

That is, to avoid violations of the KPI it is necessary to improve the metric “Delivery Time Supplier” to the value  $< 1$  and the metric “Delivery Time Shipment” to the value from 2 to 4, or alternatively improve the metric “Delivery Time Supplier” to the value  $< 3$  and the metric “Delivery Time Shipment” to the value  $< 2$ , or “Order in Stock” to become true and the “Delivery Time Shipment” to the value in range  $< 4$ . Note that as the “Order in Stock” metric is not adaptable, the third alternative is not relevant for adaptation of process models (future process instances); however it could be used for adaptation of running process instances where  $\text{O=yes}$ .

<sup>1</sup> Here the notation  $\neg r_{ij}$  stands for complement of the specified range.



### 4.3 Identification of Adaptation Strategies

After the adaptation requirements are identified, the next step is to associate possible adaptation strategies which should lead to KPI fulfillment, i.e. the sets of adaptation actions that adapt the corresponding influential factors. As described in Section 4.1, for adaptable metrics a set of possible adaptation actions has been specified. The first step is thus to come up with alternative adaptation strategies, and in a second step to select one of those strategies in an optimal way.

```

1  let  $S = \emptyset$  // set of resulting strategies
2  for each  $A_i \in R$ 
3     $S = S \cup \text{strategies}(A_i)$ 
4  function strategies( $A$ )
5  let  $S_A = \{\emptyset\}$  // set of strategies for  $A$ , initially contains an empty set
6  for each  $(\mu, r) \in A$ 
7    let  $S' = S_A$  // temporary set of partial strategies built on previous steps
8     $S_A = \emptyset$ 
9    let  $act = \{a \mid \mu \in M^+(a) \wedge \text{forall } (\mu', r') \in A, \neg(\mu' \in M^-(a))\}$ 
10   if  $act = \emptyset$  return  $\emptyset$  // the whole alternative cannot be achieved
11   for each  $a \in act$  // build a Cartesian product of actions
12     for each  $s \in S'$ 
13        $S_A = S_A \cup \{s \cup \{a\}\}$ 
14   return  $S_A$ 

```

**Fig. 3.** Strategy selection algorithm

The algorithm for identifying adaptation strategies is represented in Fig. 3. The set of strategies contains the strategies for all the alternatives. For every alternative the following procedure is applied (lines 4-10). For each of the metric to be adapted, we select the set of actions that improve it without negatively affecting other metrics to be adapted (line 9). If this set is empty for some metric, the alternative cannot be satisfied and an empty result is returned. Otherwise, a Cartesian product of those actions with the actions for other metrics is created (lines 11-13). The resulting set of strategies is returned. For the sake of simplicity we omit here formal proofs of the algorithm correctness.

It is easy to see that any of the strategies extracted in this way will satisfy the identified adaptation requirements. However, the effect of different adaptation strategies on the SBA is not the same. This is because adaptation strategies depending on contained adaptation actions differ in their negative effects on certain applications metrics.

To order the strategies we adopt a heuristic, in which the strategy with less negative effects is more preferable. All the strategies are then ordered according to this number: the lower this number is the more the adaptation strategy is preferred. Note that even if the two actions in the strategy negatively affect the same metric, the effect is counted twice as it may have stronger impact. However, other approaches and heuristics for the selection of an optimal adaptation strategy may be thought of. Some of them are discussed in the conclusions section.

## 5 Related Work

The field of QoS-aware adaptable SBAs has only recently been given attention, which is also reflected in the scarce amount of related literature. There are no approaches, to the best of our knowledge, that enable adaptation of SBAs based on quality characteristics yet in an integrated manner across all layers, based on monitoring and analysis of KPIs and the corresponding influential factors.

There are several existing works in the context of QoS-aware service compositions [9, 11] which describe how to create service compositions which conform to global and local QoS constraints taking into account process structure when aggregating QoS values of atomic services. These approaches can be used for QoS-based adaptation by replanning the service composition during monitoring [12]. Our approach is different in that we not only take into account QoS but also quality characteristics from other SBA layers and perform analysis based on their dependencies. We do not (yet) exploit information on process structure during dependency analysis, as the approach described in [10], but use decision tree algorithms instead.

Closely related to our approach is iBOM [13] which is a platform for monitoring and analysis of business processes based on machine learning techniques. It focuses on similar analysis mechanisms as in our approach, but does not deal with adaptation of SBAs by extracting adaptation requirements from the decision trees and automatically deriving adaptation strategies, but uses simulation and what-if analysis techniques instead.

Work on service composition adaptation is available and the existing approaches that do not focus on QoS-awareness of SBAs have been classified. The available approaches are mechanisms for service composition adaptation can similarly be borrowed in the approach presented in this paper as adaptation mechanisms on the service composition level [5, 14].

## 6 Conclusions and Future Work

In this paper we have presented a novel adaptation approach for SBAs based on quality factor analysis. We have extended previous work on quality factor analysis by showing how the resulting dependency tree can be used for adaptation purposes. In particular we have shown how to model adaptation actions and associate them with quality metrics, how to extract adaptation requirements from the dependency tree and come up with an adaptation strategy.

Our future work involves extending the approach in several ways. First, it is possible to define global SBA constraints as a metric that should not be negatively affected by any of the adaptation actions. If some action may violate such a constraint, it should be excluded. Second, if it is possible to capture the effect of the adaptation action onto the metric with a higher precision (e.g., instead of simple positive/negative contribution give a numerical value or even specify the effect of the action on the metric value), then the analysis should give precedence to the actions with better effect. Finally, it is possible also to exploit the relation between the metric and the number of KPI violations. This would allow also for ordering the requirements: the more violations are associated with the metric value, the more important it is. The adaptation actions, therefore, should be selected accordingly.

**Acknowledgements.** The research leading to these results has received funding from the European Community's 7th Framework Programme under the Network of Excellence S-Cube Grant Agreement no. 215483.

## References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. *IEEE Computer* 11 (2007)
2. Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar, S., Leymann, F.: Monitoring and Analyzing Influential Factors of Business Process Performance. In: *Proceedings of EDOC 2009, Auckland, New Zealand* (2009)
3. Council, S.: *Supply Chain Operations Reference Model Version 7.0* (2005)
4. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
5. Karastoyanova, D., Houspanossian, A., Cilia, M., Leymann, F., Buchmann, A.: Extending BPEL for Run Time Adaptability. In: *Proceedings of EDOC 2005, Enschede, The Netherlands* (2005)
6. Danylyevych, O., Karastoyanova, D., Leymann, F.: Optimal Stratification of Transactions. In: *Proceedings of ICIW 2009, Venice, Italy* (2009)
7. Tinelli, C.: A DPLL-based Calculus for Ground Satisfiability Modulo Theories. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) *JELIA 2002. LNCS (LNAI)*, vol. 2424, pp. 308–319. Springer, Heidelberg (2002)
8. Bozzano, M., Bruttomesso, R., Cimatti, A., Junttila, T., van Rossum, P., Schulz, S., Sebastiani, R.: An Incremental and Layered Procedure for the Satisfiability of Linear Arithmetic Logic. In: Halbwachs, N., Zuck, L.D. (eds.) *TACAS 2005. LNCS*, vol. 3440, pp. 317–333. Springer, Heidelberg (2005)
9. Zeng, L., Benatallah, B., Dumas, M., Kalagnamam, J., Chang, H.: QoS-aware Middleware for Web Services Composition. *IEEE Trans. on Software Engineering* 30(5) (May 2004)
10. Bodestaff, L., Wombacher, A., Reichert, M., Jaeger, M.C.: Monitoring Dependencies for SLAs: The MoDe4SLA Approach. In: *Proceedings of SCC 2008, Washington, DC, USA* (2008)
11. Jaeger, M.C., Muhl, G., Golze, S.: QoS-aware Composition of Web Services: An evaluation of selection algorithms. In: *Proceedings of COOPIS 2005, Cyprus* (2005)
12. Canfora, G., di Penta, M., Esposito, R., Villani, M.L.: QoS-Aware Replanning of Composite Web Services. In: *Proceedings of ICWS 2005, Orlando, USA* (2005)
13. Castellanos, M., Casati, F., Shan, M.C., Dayal, U.: iBOM: A Platform for Intelligent Business Operation Management. In: *Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*, pp. 1084–1095 (2005)
14. Karastoyanova, D., Leymann, F., Buchmann, A.: An Approach to Parameterizing Web Service Flows. In: Benatallah, B., Casati, F., Traverso, P. (eds.) *ICSOC 2005. LNCS*, vol. 3826, pp. 533–538. Springer, Heidelberg (2005)

# Delivering Multimedia in Autonomic Networking Environments

Vassilios Kaldanis<sup>1</sup>, Ranganai Chaparadza<sup>2</sup>, Giannis Katsaros<sup>1</sup>,  
and George Karantonis<sup>1</sup>

<sup>1</sup> VELTI S.A. Mobile Marketing and Advertising,  
42 Kifissias Ave, 15125 Maroussi, Athens, Greece,  
{vkaldanis, gkatsaros, gkar}@velti.com

<sup>2</sup> Fraunhofer FOKUS, Institute for Open Communication Systems,  
31 Kaiserin-Augusta-Allee, D-10589, Berlin, Germany  
Ranganai.Chaparadza@fokus.fraunhofer.de

**Abstract.** This paper aims to investigate the most important aspects, features and requirements around the delivery of Multimedia services in cross-layer architectures that can be exploited for the development of autonomic service management middleware on the top of autonomic networking environments. Fundamental autonomic features like self-management, self-healing and self-adaptation are closely (re) considered and analyzed through the prism of network agnostic media adaptation and other application-layer management mechanisms with the aim to identify the expected impact of autonomies in end user service deployment experience. The work describes on a high level the specific features and requirements of a proposed autonomic service management framework for the support of steaming media applications in fixed and mobile autonomic accessed networking environments. The work is part of the EFIPSANS project one of the largest research project in IPv6 enabled autonomic networks and *Extensions to IPv6 (IPv6++)* towards the Self-Managing Future Internet.

**Keywords:** Autonomic Networking, Service Management, Self-Management, Self-Adaptation, Med.

## 1 Introduction

*Autonomicity* realized through *control-loop structures* operating within network nodes/devices and the network as a whole, is an enabler for advanced and enriched self-manageability of network devices and networks. We define an *autonomic behaviour* as follows: It is a *behaviour* or *action* that may consist of a set of sub-behaviours or sub-actions, triggered by what is called a *Decision-Making Element* (DME) or Decision Element (DE) in an attempt to achieve the goal defined by how the DME manages a Managed Entity (ies) or ME(s) under its control. The autonomic behaviour is considered as a behaviour of the DE or DME (combined DE/DME), triggered as a result of reception of information from its information suppliers such as its associated Managed Entity (ies), in an attempt to regulate or reconfigure the behaviour of the Managed Entity (ies). An example of an autonomic behaviour is: *self-description* and

*self-advertisement, self-healing, self-configuration*, all triggered by an *Decision-Making Element*. Therefore, it is important to note that an autonomic behaviour binds to a *Decision-Making Element*, and possibly (though not necessarily) to information supply parts of the control-loop implemented by the *Decision-Making Element* together with the Managed Entity (ies) under the control of the *Decision-Making Element*. A *Decision-Making Element* can be introduced as standalone entity in a node or as part of a functional entity such as a protocol or application.

This paper attempts through a high-level approach, to identify the different aspects and perspectives of multimedia delivery, both in fixed/wireless and mobile access environments that can be exploited in the design of an autonomic self-managed and self-adapted system. To this end it proposes an autonomic mechanism for the efficient self-management of multimedia services and applications taking advantage at the highest possible level the existing underlying autonomic networking environment. Important issues across application to transport layer like media and network adaptation, resource utilization, and congestion control are under focus in cross-layered architectures as appear today. Our purpose is to provide an innovative framework that will effectively and autonomously manage well known protocols like SIP/SDP, RTP/RTCP, and RTSP, while at the same time being tightly coupled with an also autonomous networking environment.

End users deploying pervasive services on the top an IPv6/IPv6++ autonomic and heterogeneous network expect to meet certain levels of QoS and QoE or performance conditions. Our approach aims at dealing with all required underlying network management and control issues and complexities in order to deliver a reliable and efficiently self-manageable transport layer that will transparently support a wide range of end user applications. Isolation of end users from any form of network management is the most important requirement that will allow the deployment of pervasive multimedia services in the form of context aware applications.

The field of autonomics is expected to highly influence the effective wedding of the two major fields of service and network management together into a self tuned and self-managing network. The proposed mechanisms aim at a first stage to address all required autonomic network management issues around QoS and resource availability as well as the lower layer aspects like connectivity, routing etc as well. The framework also provides an advanced performance monitoring mechanism capable to measure a number of autonomic oriented metrics and parameters both in network and service/application part. The running application on the top of the autonomic node highly relies on the collected performance data to feedback the corresponding autonomic mechanisms in order to:

- Self-adapt to the changing network conditions
- Self-manage the current application session (media)
- Self-heal from severe network conditions (e.g. congestion)

Finally the study provides a unified methodology for the evaluation and assessment of the impact of autonomic behaviors on application performance, providing the mechanism for collecting the end-to-end performance data of applications and the transfer of these performance metrics to the autonomic node in order to adapt the application. The GANA based autonomic service management framework along with the former performance estimation strategy consist a complete framework for the evaluation of autonomic communications systems.

For comparison purposes in [7] a similar work is also presented as an early attempt to address automated management of services. There the proposed architecture ensures the delivery of services according to specific service level agreements (SLAs) between customers and service providers. Our approach takes a rather more realistic view maintaining the distinction between the service and the network layers capturing the required interfaces and interactions among the proposed autonomic entities and the individual underlying protocols as managed entities.

The work presented in this paper is structured as follows: in chapter 2 the particular aspects and parameters of the multimedia delivery mechanisms in cross-layer architectures are presented focusing on the media management and adaptation approaches today. In chapter 3 a brief analysis of the Generic Autonomic Network Architecture (GANA) is presented with special attention on the higher layer components directly involved with the service management processes. In chapter 4 the desired autonomic features are interpreted in the language of multimedia application and service management as presented in chapter 2. Then a detailed analysis of the GANA compatible Service Management framework is provided along with requirements description.

## 2 Aspects of Multimedia Delivery in Cross-Layered Environments

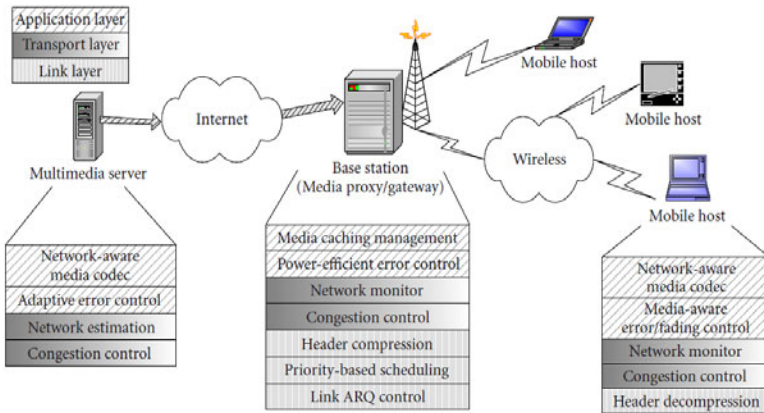
Delivering multimedia over the Internet is a quite challenging task that requires specific requirements to be met in many different parts of the network, the most important being the transmission, the transport and the network access. Successful delivery typically imposes strict QoS requirements on bandwidth, delay and delay jitter and only best-effort service can be supported assuming constantly varying network conditions. Such variation becomes more unpredictable and rapid in the case of wireless and mobile transmission where mobility further affects network conditions by moving between servicing areas with limited resource availability and connectivity—making delivery more difficult. In order to improve perceived multimedia quality and guarantee delivery by end users over wired and wireless/mobile Internet QoS provision is performed across different layers ranging from application, transport to data link layer following the so-called cross-layered approach. Different layers impacts differently the delivered multimedia quality and therefore they impose different approaches for improving delivery depending on the case.

In [2] a quite enlightening diagram of a typical cross-layer architecture and involved layers in different components for multimedia delivery has been borrowed from [2] and is provided in the following figure (fig 1).

The three key components in a wireless Internet architecture for multimedia delivery are: the multimedia server, the Base Station (BS) and the mobile hosts. In the multimedia server component the application layer is responsible for the session management using the key functionalities: **network aware media adaptation** and the **adaptive error control**.

The network aware media adaptation functionality determines the media codec type that is used depending the detected network conditions. It has the ability to dynamically change the coding rate and other coding parameters adapting in this way to the varying network conditions (bandwidth, loss, delay etc). Furthermore scalable coding techniques are also introduced to realized this type of media adaptation the most important of

which is layered coding (e.g. H.263 and MPEG-4). The full details of media coding functionality although here gets out of scope however it is of major importance that the application server gets constant feedback about the current network conditions on the reception end and based on that it selects the appropriate media coding scheme to transport the multimedia content to the end user dynamically during the session.



**Fig. 1.** Cross-layer architecture for multimedia delivery over wireless Internet (source [2])

In the case of adaptive error control it is mainly used to shield delay sensitive applications like voice in the case of problematic wireless transmission where efficient channel coding techniques are required (e.g. ARQ and FEC) to protect data integrity.

To complete the puzzle, in the multimedia server component the transport layer or multimedia transmission control layer is responsible to efficiently deliver the service to the end user at the best possible quality. To accomplish this role it is important to estimate the status of all underlying networks so that multimedia can adapt accordingly. This is achieved through the deployment of specific streaming control transport protocols for the monitoring and control of the media streaming process like the IETF real-time transport control protocol (RTP/RTCP), real-time streaming protocol (RTSP), session initiation protocol (SIP), session description protocol (SDP), and streaming control transport protocol (SCTP). The degree at which these protocols are able to achieve a desirable media streaming quality relies on the accuracy of the estimation of the network conditions.

Estimation of network conditions in practice is performed through detection of current available bandwidth and application of efficient congestion control. A proper congestion control scheme should maximize the bandwidth utilization and at the same time avoid overusing network resource which may cause network collapse. Using a TCP friendly streaming protocol is able to apply two kinds of protocol based congestion control: the window-based and the model-based. The window-based congestion control performs additive increase and multiplicative decrease (AIMD) rate adjustment which is similar to TCP.

As seen in Fig. 1, network estimation information is measured created (structured into metrics) and updated in Media Proxy/Gateway and End Nodes (mobile host)

while it is transferred and continuously maintained at the application server. This scheme is required in order to provide the application server with all required information for create establish and manage an application session with an end node through a network of intermediate nodes. All former analysis has been presented here in order to define the environment that autonomous service management components will be specified as seen next.

### 3 The Service Management Layer in GANA

The next figure (fig 2) illustrates the structure of a GANA node its network and service layer DEs and how these DEs are managing its assigned MEs. For this study the DEs of prime interest are: the Service-Management-DE, Monitoring-DE, the Mobility-Management-DE, the QoS-Management-DE, the Forwarding-Management-DE and the Routing-Management-DE.

Each of these DEs, manages specific protocols and mechanisms assigned to it according to the aspects abstracted by the DE. The Routing-Management-DE manages the routing protocols and mechanisms of the node; the Forwarding-Management-DE manages the corresponding protocols for Layer-3/Layer-2.5/Layer Forwarding as well as Layer-3/Layer-2 Switching supported by the node (e.g. IPv6, MPLS, Ethernet). The Monitoring-DE manages those monitoring protocols and mechanisms that can be orchestrated for the benefit all the functions of the node including applications. The QoS-Management-DE manages the QoS related issues such as the schedulers and queue parameters on the interfaces and links connected to the node. The Mobility-Management-DE manages the mobility related mechanisms and protocols of the node.

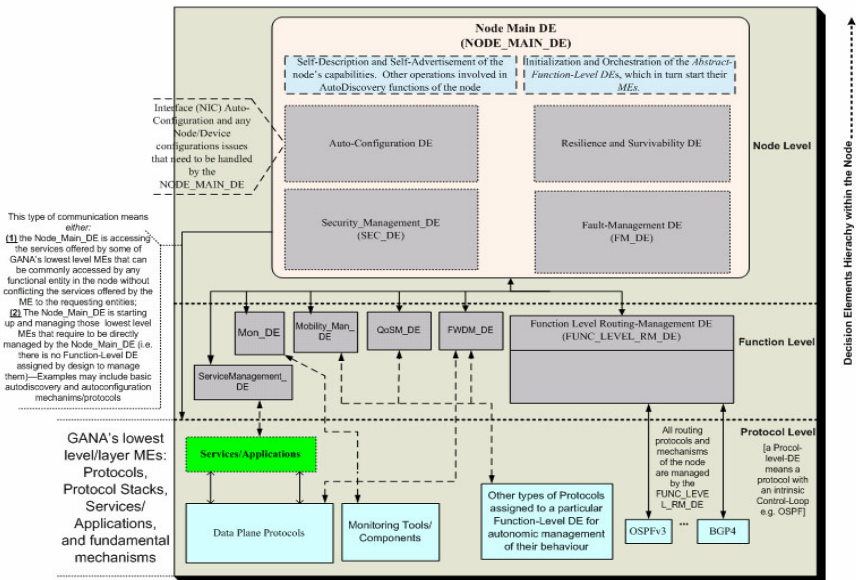


Fig. 2. The GANA Node with the Service Management Layer



Finally the Node-Main-DE controls the overall behaviour of the node consisting of some sub-DEs (e.g. Security-Management-DE), which are considered part of the Node-Main-DE which require exclusive access to the lower DEs down to the lowest level of protocols and protocols stacks of the GANA node for the purposes of managing those aspects that are shared by all entities of the node or benefit them all.

## 4 Autonomic Features under the Prism of Multimedia Delivery

In order to efficiently make applications and services benefit from the autonomic features of the lower layers GANA introduces the need for interactions between network functions specific DEs and the Service-Management-DE in order to allow network functions specific DEs to know how to configure and manage their MEs, towards meeting the requirements imposed by Services and Applications, such as network resilience and QoS guarantees, etc. This creates the notion of a network-layer that is application and service-aware, including transport and network-layer awareness to survivability requirements of services and applications, which enables the network to react to the needs of applications and services. The survivability requirements of an application define the time constraints within which the application should be notified by the underlying transport and network layers of incidents requiring it to adapt to the challenging adverse conditions of the network.

In GANA, self-adaptation is implemented by the Forwarding-Management-DE or FWDM\_DE (see the FWDM\_DE on the Fig.2). All the DEs implement some kind of self-adaptation behaviors specific to the DE, which ensure that the node fulfils the optimal goal of its overall functionality and the performance of the applications and services running on the node. In [5] and [6] the reader can find an analysis of how self-adaptation on routing is achieved through the Routing-Management-DE and adaptive behaviors based the Monitoring-DE respectively. A representative paradigm of self-adaptation can be also found in [4] as implemented by the FWDM\_DE for better understanding.

Based on the former analysis of the multimedia delivery aspects and involved functionality an interpretation of the described adaptation mechanisms into autonomous features in autonomic terms as follows:

- **Media Codec Self-Adaptation:** The autonomic node AND the autonomic server should be capable to self monitor the underlying application session conditions and connection environment, exchange information between them about network conditions estimation and decide about the most suitable coding mechanism that will guarantee the particular QoS level(s) at the end node and servicing network.
- **Service/Application Self-Management:** The autonomic node after getting continuous monitoring information about its state, the connected access network conditions and the application quality at the end user it decides about the necessary parameter change(s) to be requested from other autonomic components (e.g. QoS components) in order to protect or maintain the agreed application/service quality.

- **Service/Application Self-Healing:** The autonomic node in connection with other autonomic DEs and the autonomic application server will decide about the policy/strategy to be applied in the case of application collapse due to specific reason(s) identified and known to the autonomic node (e.g. congestion, channel fading, packet loss).

## 5 GANA-Based Service Management for Streaming Media

### 5.1 Requirements

The Service Management autonomic component called Service Management Decision Element (SM\_DE) has the following roles:

- Guarantee the required network resources (through QoS requests) the protocol-level session management entities like SIP/SDP, RTP/RTCP, RTSP, etc, considered as Managed Entities (MEs) in the GANA architectural Reference Model.
- Monitor the application level performance metrics through communication with the Service-related Managed Entities (e.g. streaming packets delay & loss through RTCP)
- Guarantee the best possible service quality and experience at the end user interface
- Continuously balance and maintain service stability between application-level quality and network resource availability

From the functional perspective the SM\_DE is expected to manage all underlying lower protocol level entities under the general term Session-associated Managed Entities (MEs) responsible for service discovery, registration, initiation, control, etc, performing the following tasks:

- Request the service initiation (in the case of SIP, SDP) to launch the desired service after contacting the remote application server (e.g. INVITE, ACK)
- In the case of multimedia communication it activates and manages the RTP/RTCP protocols in order to efficiently control the formerly launched session.
- After successful session launch, it continuously monitors its progress and performance by exchanging information about perceived service quality (streaming packet loss and delay) between end points.
- Contact QoS Management DE in the same layer (e.g. Function-Level in GANA) in order to negotiate service specific QoS levels
- Continuously receive monitoring information from the Monitoring DE about underlying network performance in order to maintain a combined view of network and transport conditions the active session runs with.

### 5.2 Proposed Framework for Service Management

The Service Management DE or SM\_DE is the core component for GANA-based (fig. 3) for autonomic service management that is responsible for:

- Manage the session and application level mechanisms at the lower protocol level (e.g. RTP, RTCP) in the case that running application(s) require more media resources (e.g. better video quality, higher data rate) by contacting the remote application server.
- Request more network resources (e.g. increase QoS, stronger channel coding due to errors in wireless transmission) in the case of bandwidth greedy application by contacting directly other DEs at the same layer (e.g. QoS\_DE)
- Continuously monitor the network performance metrics (delay, loss, jitter) in close connection with Monitoring\_DE and cross check the observed network conditions with the application level metrics (e.g. congestion, delay, loss) in order to act proactively about an upcoming application collapse event (it may consult application server through RTCP to reduce streaming packet rate due to the detection of bandwidth fluctuations at the end user)

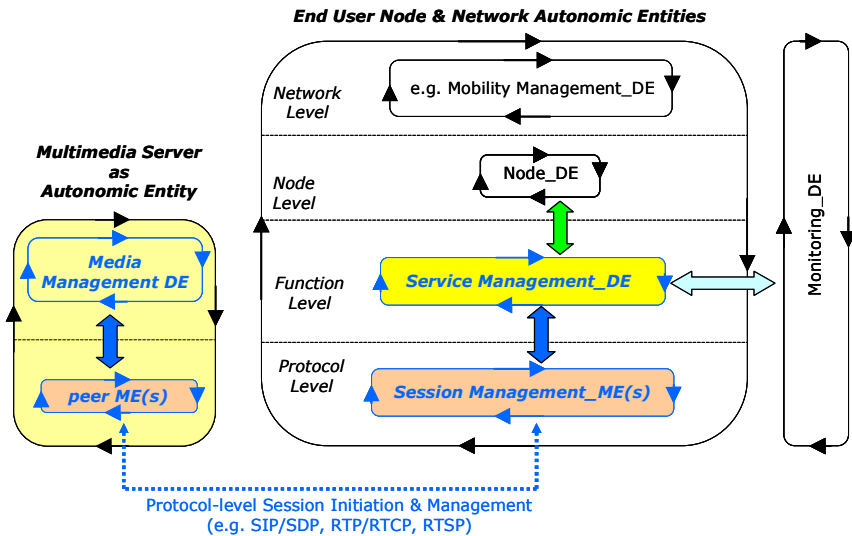


Fig. 3. Service Management ecosystem in GANA

At the protocol level the Session Management Managed Entity (Session Management ME) in one of the known transport level protocols for service management like SIP/SDP, RTP/RTCP, etc as described in chapter 2. This ME is managed directly by the SM\_DE above and do not take decisions about the session before it contact SM\_DE. Obviously in this way the SM\_DE is the coordinating entity that upon getting a clear view of existing conditions on the network and application parts decides about the required modifications in system parameters regarding performance and end user application quality.

## 6 Conclusions and Future Work

In this paper the high-level requirements for multimedia delivery in an autonomic network exploiting the benefits of cross-layer techniques is presented. The main focus was on the emerged evolvable holistic architectural based on GANA Reference Model for autonomicity and self-management within node and network architectures and their proposed extension for services and application layer. In order to efficiently make applications and services benefit from the autonomic features of the GANA lower layers a number of interactions between network level Decision Elements (DEs) and the Service Management DE has been introduced in order to allow network function specific DEs to know how to configure and manage their Managed Entities (MEs), towards meeting the requirements imposed by Services and Applications, such as network resilience and QoS guarantees, etc. Furthermore self-adaptation of DEs itself is significantly improved ensuring that the node fulfils the optimal goal of its overall functionality while the performance of node's running services and applications as well.

## Acknowledgement

This work has been partially supported by EC FP7 EFIPSANS project (INFSO-ICT-215549).

## References

1. EU funded FP7-EFIPSANS Project, <http://efipsans.org/>
2. Zhang, Q., Yang, F., Zhu, W.: Cross-Layer QoS Support for Multimedia Delivery over Wireless Internet. *EURASIP Journal on Applied Signal Processing* 2005(2), 207–215 (2005)
3. Chaparadza, R., Papavassiliou, S., Kastrinogiannis, T., Vigoureux, M., Dotaro, E., Davy, A., Quinn, K., Wódczak, M., Toth, A., Liakopoulos, A., Wilson, M.: Creating a viable Evolution Path towards Self-Managing Future Internet via a Standardizable Reference Model for Autonomic Network Engineering. In: *FIA Prague 2009 Conference*, Published in the *Future Internet Book* produced by FIA (2009)
4. Chaparadza, R., Prakash, A.: Self-configuring and Self-Adaptive Forwarding in the GANA based Self-Managing Future Internet Architecture. Submitted to the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), November 23 or 24 (2009), co-organized with the ICSOC/ServiceWave 2009 Conference, Stockholm, Sweden
5. Rétvari, G., Németh, F., Chaparadza, R., Szabó, R.: OSPF for Implementing Self-adaptive Routing in Autonomic Networks: a Case Study. In: *Proceedings of the the 4th IEEE International Workshop on Modelling Autonomic Communication Environments (MACE 2009)*, Venice, Italy, October 26-27 (2009)
6. Zafeiropoulos, A., Liakopoulos, A., Davy, A., Chaparadza, R.: Monitoring within an Autonomic Network: A GANA based Network Monitoring Framework. Submitted to the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), November 23 or 24 (2009); co-organized with the ICSOC/ServiceWave 2009 Conference, Stockholm, Sweden
7. Cheng, Y., et al.: A generic architecture for autonomic service and network management. *Computer Communications Journal*, Copyright 2006 Elsevier B.V. All rights reserved (2006), doi:10.1016/j.comcom.2006.06.017

# An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring\*

Dragan Ivanović<sup>1</sup>, Manuel Carro<sup>1</sup>, and Manuel Hermenegildo<sup>1,2</sup>

<sup>1</sup> School of Computer Science, T. University of Madrid (UPM)

<sup>2</sup> IMDEA Software, Spain

idragan@clip.dia.fi.upm.es, {mcarro,herme}@fi.upm.es

**Abstract.** Several activities in service oriented computing can benefit from knowing ahead of time future properties of a given service composition. In this paper we focus on how statically inferred computational cost functions on input data, which represent safe upper and lower bounds, can be used to predict some QoS-related values at runtime. In our approach, BPEL processes are translated into an intermediate language which is in turn converted into a logic program. Cost and resource analysis tools are applied to infer functions which, depending on the contents of some initial incoming message, return safe upper and lower bounds of some resource usage measure. Actual and predicted time characteristics are used to perform predictive monitoring. A validation is performed through simulation.

**Keywords:** Service Orchestrations, Resource Analysis, Data-Awareness, Monitoring.

## 1 Introduction

Service Oriented Computing (SOC) is a well-established paradigm which aims at expressing and exploiting the computational possibilities of remotely interacting loosely coupled systems that expose themselves using service interfaces, while the implementation is completely hidden. Several services can be *put together* to accomplish more complex tasks through *service compositions*, written using some general-purpose programming language or a specifically designed language [1,2,3], and are usually exposed as full-fledged services.

---

\* The research leading to these results has received funding from the European Community's Seventh Framework Programme under the Network of Excellence S-Cube - Grant Agreement n° 215483. Manuel Carro and Manuel Hermenegildo have also been supported by projects FET IST-231620 *HATS*, Spanish FIT-340005-2007-14 *ES\_PASS* (within EU ITEA2 06042-ESPASS) and 2008-05624/TIN *DOVES* projects, and CAM project S-2009/TIC/1465 *PROMETIDOS*.

SOC systems are expected to be active during long periods of time and operate across geographical and administrative boundaries. This requires monitoring and adaptation capabilities: monitoring compares the actual and expected behavior of the system (e.g., its QoS), and may trigger an adaptation if needed. Comparing the actual and the expected QoS of a composition—even assuming the composition is static—is far from trivial. *Predictive* monitoring aims at detecting deviations ahead of time by e.g. forecasting the future behavior. This is more complex but also more interesting and useful, as it can perform *prevention* instead of *healing*. Clearly, greater accuracy in calculating the expected QoS leads to better predictions.

Two factors, at least, have to be considered when estimating QoS behavior: the structure of the composition itself, i.e., what it does with incoming requests and which other services it invokes and how, and the variations on the environment, such as network links going down or external services not meeting the expected deadlines.

Of these two sources of information, the latter has been extensively studied [4,5,6,7], while the former has been less deeply explored. The actual data a service manages have been recognized as relevant [8,9], and actual message contents can greatly influence the runtime behavior of a composition (Section 2), but it has not been adequately addressed so far: prediction techniques which do not take run-time parameters into account are potentially inaccurate.

In this paper we will focus on applying a methodology, based on previous experience on automatic complexity analysis [10,15,16], which can generate correct approximations of complexity functions via translation to an intermediate language (Sections 3). These functions use (abstractions of) incoming messages in order to derive safe upper and lower bounds which depend on the input data and which are potentially more accurate than data-unaware approximations. In Section 4 we show how these functions can be used by monitoring to make better decisions and in Section 5 we make an experimental evaluation.

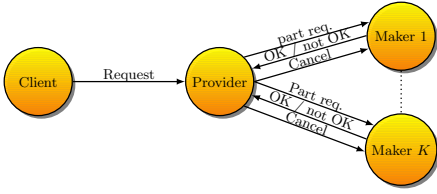
Correct data-aware cost functions can be useful for any situation where a more informed QoS estimation is an advantage. In particular, QoS-driven service composition [11,12,13] can use them to select better service providers for an expected kind of request, and adaptation mechanisms can also benefit from that knowledge [14].

## 2 A Motivating Example

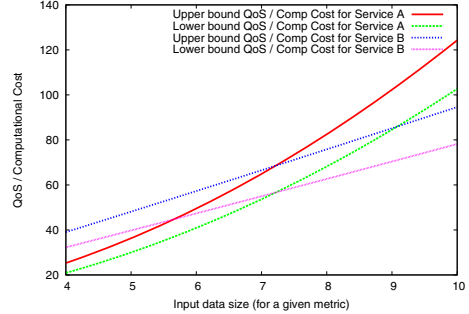
We illustrate with a simple example how actual data can be taken into account when generating QoS expressions for service compositions.

**Example 1.** *In a hotel reservation system (Fig. 1) a Client contacts a Booking Agency to request  $N$  hotel rooms. The Booking Agency uses a composed service that either books  $N$  rooms from a pool of  $K$  hotels, one at a time, or replies*

<sup>1</sup> The adaptation technique presented in [14] uses the same technique presented in this paper to derive cost bound functions. However, here we focus on the different problem of its application to predictive monitoring.



**Fig. 1.** Simplified hotel reservation system



**Fig. 2.** Upper, lower bounds for two services

that no rooms are available. If not enough rooms are available after scanning all the hotels, it cancels any previously made reservation. A hotel without available rooms is excluded from further search. One message is used to for each room query, confirmation / rejection, and cancellation.

The whole process cannot be done in a single transaction, because the reservation systems of different hotels are disconnected; therefore it has to be instrumented at the level of composition. We will assume that we are interested in the number of messages sent / received. There are several reasons for this: message exchanges can carry a sizable overhead, thus affecting actual execution time, or it is possible that the hotel reservation / booking service take a toll on every message they answer.

Assuming  $K \geq N$ , the *smallest* number of exchanged messages for a successful reservation  $2N$  ( $N$  requests and replies to the same hotel) while the *greatest* number of messages is  $2K + 3(N - 1)$  for the worst case of an unsuccessful reservation ( $N - 1$  successful reservations, plus one last unsuccessful reservation which triggers their cancellation). Between these extremes, the maximum for a successful reservation would be  $2K + 2(N - 1)$  messages. The complexity analysis depends both on the structure of the composition and on the values of  $N$  and  $K$ , which are composition parameters, since it is more likely that the hotels are listed in a separate registry, than hardwired into the composition code.

Compared with probabilistic approaches, the following differences can be pointed out:

- If data is not taken into account, the impact of loops and conditionals can be only statistically estimated. *Guarantees* cannot easily be provided, as the value for any QoS attribute will be constant regardless of the actual values for  $K$  and  $N$ .
- Safe (upper and lower) bounds cannot usually be obtained, as probabilistic formulations usually rely on some kind of average.
- In QoS-aware matchmaking / rebinding, comparing different service compositions ignores the functional dependencies of QoS on the data.

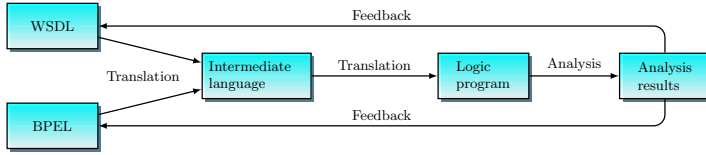


Fig. 3. The overall process

Table 1. Abstract orchestration elements

<i>Declarations and definitions</i>			
<i>Complex type definition</i>	<code>:-struct(QName, Members).</code>		
<i>Port type definition</i>	<code>:-port(QName, Operations).</code>		
<i>External service</i>	<code>:-service(PortName, Operation, (TrustedProperties)).</code>		
<i>Service definition</i>	<code>service(Port, Operation, InMsg, OutMsg) :- Activity.</code>		
<i>Activities</i>			
<i>Variable assignment</i>	<code>Var &lt;- Expr</code>	<i>Service invocation</i>	<code>invoke(Port, Op, OutMsg, InMsg).</code>
<i>Reply and exit</i>	<code>reply(OutMsg)</code>	<i>Sequence</i>	<code>Activity<sub>1</sub>, Activity<sub>2</sub></code>
<i>Conditional execution</i>	<code>if(Cond, ActThen, ActElse)</code>	<i>While loop</i>	<code>while(Cond, Activity)</code>
<i>Scope</i>	<code>scope(VarDecl, ActivityList)</code>	<i>Scope fault handler</i>	<code>handler(FaultName, Activity)</code>
<i>Parallel flow</i>	<code>flow(LinkDecl, Activities)</code>	<i>Activity in a flow</i>	<code>float(Attributes, Activity)</code>

Figure 2 portrays upper and lower bounds of two compositions for some QoS as a function of a single input parameter. For some ranges of data input one composition is preferable over the other, while in the central zone we cannot decide.

### 3 Resource Analysis for Orchestration

Due to space constraints, we present here an abridged version of the resource analysis technique in [14]. Our approach is based on translating process definitions into a language for which automatic computational cost analysis tools are available (see Fig. 3).

#### 3.1 Overview of the Translation

Our orchestration language is a subset of BPEL 2.0 with WSDL meta-information. These are translated into an intermediate language which is then translated into the *Ciao* logic programming language [17]. The resulting logic program is then analyzed by the *CiaoPP* tool [18], which is able to infer upper and lower bounds for computational costs [15], among other analyses.

A BPEL process definition is translated into a service definition which associates a port/operation with the orchestration body. Processes forming a service network are translated into predicates that call each other to mimic service invocations. The intermediate language can describe XML schema-derived data types for messages, service ports, as well as relevant properties of external services of interest for the analysis when such services cannot be analyzed. Supported operations include generic constructs (assignment, sequence, loop...), and specific workflow constructs, such as flows, scopes, and invocations.



The translation does not follow strictly the operational semantics of the orchestration language: it just captures enough of it to ensure that the analyzers will infer correct information while minimizing precision loss.

Our analysis is currently restricted to orchestrations that start after receiving an initial message and finish by returning a reply or a fault notification. Stateful service callbacks using correlations are not supported. We support however a variant of the `scope` construct, which introduces local variables and fault / compensation handlers. We do not fully support compensation handlers, which in BPEL “undo” the effects of a scope using snapshots of variables recorded at successful completion of the scope. Except for recording snapshots, compensation handlers can be treated as pseudo-subroutines on a scope level, and inlined at their invocation place.

### 3.2 A Sketch of the Translation

The simple types in XML schemata are abstracted as three disjoint types: `numbers`, `strings` (translated into `atoms`), and `booleans`. Complex XML types are translated into predicates specifying how the type is built. The accepted expression language is a subset of XPath, so that navigation is statically decidable and components of XML structures can be passed as separate arguments when necessary to improve analyzer accuracy.

A process that implements operation  $o$  on port  $p$  is translated into a clause:

$$s_{p.o}(X, Y) \leftarrow T([A], \eta, Y)$$

where  $X$  and  $Y$  correspond to the initial message and the final reply, and  $\eta$  is an environment that maps orchestration variables to logical variables.  $T$  is the translation operator, and  $[A]$  is the process body.  $T$  is defined for both simple and structured activities, and may generate auxiliary predicates if needed. Table 2 sketches the translation of some activities.

A translation example is presented in Fig. 4. Subfigure (a) is a BPEL fragment of an orchestration that cancels the list of reservations and reports a fault, (b) is the corresponding intermediate form, and (c) is the translation into a logic program.

The resource analysis finds out how many times external service invocations will be performed during process execution, from which deducing the number of messages exchanged is easy. The upper and lower bounds estimates (as functions of input data) for the complete orchestration are displayed in Table 3. Results are given for both the fault-free execution case, and the more general case with possibility of faults.

**Table 2.** Translation of different activities

$A$	Translation of $T([A R], \eta, V)$	$A$	Translation of $T([A R], \eta, V)$
<code>reply(v)</code>	$V = \text{reply}(\eta(v))$ ( <i>End orchestration</i> )	$A_j, A_k$	$T([A_j, A_k R], \eta, V)$ ( <i>Sequence</i> )
<code>throw(f)</code>	$V = \text{fault}(f)$ ( <i>No fault handler</i> ) $T([H], \eta, V)$ ( <i>Insert fault handler</i> )	$v \leftarrow e$	$a(\eta, Y) \leftarrow E(e, \eta, X), T(R, \eta[X/v], Y)$ ( <i>E evaluates e in env. <math>\eta</math></i> )
<code>if(c, A', A'')</code>	$a(\eta, Y) \leftarrow C(c, \eta), !, T([A' R], \eta, Y)$ $a(\eta, Y) \leftarrow T([A'' R], \eta, Y)$	<code>while(c, A')</code>	$a(\eta, Y) \leftarrow C(c, \eta), !, T([A' A], \eta, Y)$ $a(\eta, Y) \leftarrow T(R, \eta, Y)$

```

<sequence>
  <while name='a_13'>
    <condition>${i}>0</condition>
    <scope>
      <assign name='a_14'>
        <copy><from>${i} - 1</from>
        <to variable='i'></copy></assign>
      <assign name='a_15'><copy>
        <from>${resp.body/factory:part[${i}]
          </from><to variable='p'></copy>
      </assign>
      <invoke name='a_16'
        portType='factory:sales'
        operation='cancelReservation'
        inputVariable='p'
        outputVariable='r'>
    </scope>
  </while>
  <throw
    faultName='factory:unableToComplete'>
</sequence>

```

(a) A BPEL code fragment

```

while('${i}>0', ( % a_13
  '$i' <- '$i-1', % a_14
  '$p' <- '$resp.body/factory:part[${i}]', % a_15
  invoke(factory:sales, % a_16
    cancelReservation, '$p', '$r')),
  throw( factory:unableToCompleteRequest)

```

(b) The intermediate representation.

```

% (${i}, $p, $resp.body/factory:part, $r, Y)
a_13(A,B,C,D,E):- A > 0, !, a_14(A,B,C,D,E).
a_13(A,B,C,D,E):- E =
  fault('factory->unableToComplete').
a_14(A,B,C,D,E):- F is A-1, a_15(F,B,C,D,E).
a_15(A,B,C,D,E):- nth(A,C,F), a_16(A,F,C,D,E).
a_16(A,B,C,D,E):-
  'factory->sales->cancelReservation'(B,F),
  (F=fault(G) -> E=fault(G)
  ;F=reply(H) -> a_13(A,B,C,H,E)).

```

(c) Translation into logic program.

Fig. 4. Translation example

Table 3. Resource analysis results for the group reservation service

Resource ( $n \geq 0$ : input arg. value)	With fault handling		Without fault handling	
	lower bound	upper bound	lower bound	upper bound
Basic activities	2	$7 \times n$	$5 \times n + 2$	$5 \times n + 2$
Single reservations	0	$n$	$n$	$n$
Cancellations	0	$n - 1$	0	0

## 4 Cost Functions for Monitoring

In this section we will describe how the expected value of some QoS characteristics can be derived from the value of resource consumption functions and the (expected) value of some environment characteristics, and we will also show how the availability of cost functions can be used to perform predictive monitoring.

### 4.1 QoS Metrics and Cost Functions

The precise cost function which is needed to express some QoS characteristic depends on the QoS metric itself. For example, if bandwidth consumption is involved in the measure of some QoS, then the number of messages and size of each message is relevant, but the number of executed activities is not directly relevant (although possibly related). However, a cost function cannot in general convey by itself all the information necessary to represent a QoS function: some data which come from the environment is needed. Therefore, and for some QoS metrics, an interval of lower and upper bounds depending on the input data can be expressed as

$$QoS_{\langle L,U \rangle}(n) = \langle cost_L(n) \oplus env_L, cost_U(n) \oplus env_U \rangle \quad (1)$$

where the tuple components are the expected lower and upper bounds for the quality of service,  $cost_X(n)$  is a function representing a lower / upper bound on resource consumption,  $env_X$  represents the minimum and maximum influence of the environment on the QoS attribute at hand, and  $\oplus$  is an operation which combines together cost functions and environment conditions.

For example, in the case of the execution time of a single process,  $cost_X(n)$  can be the number of activities executed,  $\langle env_L, env_U \rangle$  the minimum / maximum time a single activity can take and  $\oplus$  would be just multiplication. Since  $\langle cost_L(n), cost_U(n) \rangle$  are lower and upper bounds, if  $\langle env_L, env_U \rangle$  are also safe lower and upper bounds, the calculated QoS bounds will be safe lower and upper bounds of the actual (runtime) QoS values.

This generic scheme can admit variations: for example, a more accurate approximation can be constructed by assigning different weights to activities so that  $env_X$  is an array with a component for the execution time for every type of activity,  $cost_X(n)$  is an array counting how many times every type of activity is executed, and  $\oplus$  is the vector dot product.

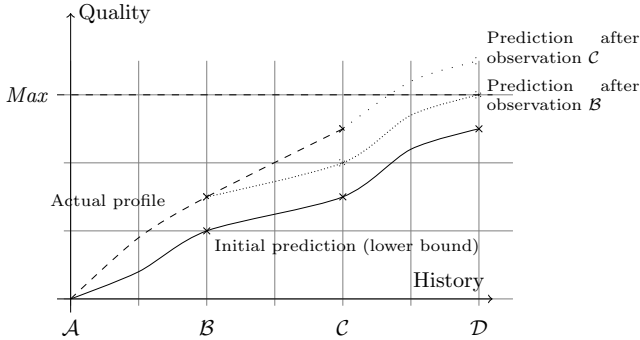
## 4.2 QoS and Cost Functions during Composition Execution

In the general case, the cost function of a composition is made up of several parts related to different blocks of the composition structure. As an example, the upper bound of an `if-then-else` activity is the upper bound of the condition plus the maximum of the upper bounds of the `then` and `else` branches.

We can associate to every point in the composition a measure of how much resources “remain to be spent”. In the `if-then-else` example, once the `if` part is over, what remains is the maximum of the upper bounds of the `then` and the `else` parts. This value depends on the point in the service composition it is measured and also on the values of the data at that moment. In a loop, where the same activity is executed several times, less “mileage” is left until the end of the execution after every iteration, even in the same point of the composition. The difference comes from the different state of the variables, and this is one of the reasons why taking data into account is beneficial.

There is, therefore, a notion of “remaining” QoS: for example, from the activities still to be executed and the expected time of every activity, the time remaining for the completion of the composition can be derived. Measuring this remaining time is relevant from a monitoring point of view. Assuming that we have a faithful predictor of the QoS remaining until the end of the execution (such as that given by resource consumption functions and environment conditions), then deviations of the environmental characteristics can be used to predict more accurately what will be the QoS at some future point by dynamically combining the cost / resource consumption functions with the actual environment conditions.

Figure 5 exemplifies such a situation. Let us assume we are interested in some QoS metric of a composition, whose value must not exceed  $Max$ . The cost functions and environment characteristics representing safe upper and lower bounds can be used here: if the **upper** bound is smaller than  $Max$ , then we have the guarantee that we do not violate the QoS boundary. If the **lower** bound is



**Fig. 5.** Actual and predicted QoS throughout history

larger than  $Max$ , then we have the guarantee that we will violate the expected QoS. If none of these hold, then we cannot say anything for sure.

We designate four points ( $A$ ,  $B$ ,  $C$ , and  $D$ ) in the execution of some composition and we will focus on how monitoring at these points can be predictively done with the use of cost functions. In Figure 5 the solid line represents the QoS initially predicted with the statically inferred cost functions and the expected environment conditions, while the dashed line represents the observed QoS.

At point  $B$ , the actual quality has deviated with respect to the predicted one. Since the composition has not changed, and thus neither have the cost functions, we can conclude that the deviation can only be due to a change in the environment behavior (e.g., additional load on a server or a faulty network). An updated prediction for the future can be done by using the environment influence observed so far and the existing cost function. This new prediction curve (densely dotted) still ends, at point  $D$ , within the limits of the acceptable range  $Max$ . However, at point  $C$  a new observation gives yet higher values for the QoS attribute. Yet another function and associated plot curve (sparsely dotted) can be constructed which predicts that the execution will violate the expected QoS. Therefore, at point  $C$  we have detected a problem before it appears and we can raise an alarm and maybe trigger an adaptation procedure. In order for this technique to work in complex service compositions with loops, different response times depending on invocations, etc. it is necessary to take data into account from the beginning.

## 5 Experimental Evaluation

To validate the applicability of the proposed approach to predictive monitoring, we conducted a series of experiments which simulate the behavior of a service composition which may violate some QoS attribute (time, in this case) and we try to detect this situation ahead of time using analytically derived complexity bounds and the observed environment conditions. In our scenario, service

composition  $A$  is initiated with an input message, whose size (using some appropriate metric) is represented as an integer  $n$  (ranging, in this case, between 1 and 50). Upon message reception, service  $A$  iteratively invokes a partner service  $B$  and waits for a reply. The number of iterations is bounded by upper and lower bounds  $\langle E_{AU}(n), E_{AL}(n) \rangle$  which grow linearly and which range between 100 and 500 and between 50 and 250, respectively, as  $n$  goes from 1 to 50.

Each invocation of  $B$  uses some time to transmit the invocation and its reply. Time is modeled as an environmental factor with bounds  $\langle u_{AL}, u_{AU} \rangle$  that may change over time. Executing  $B$  involves a number of operations (or steps) independent from  $n$  with bounds  $\langle E_{BL}, E_{BU} \rangle = \langle 8, 16 \rangle$ . Each step takes some time between the time-varying bounds  $\langle u_{BL}, u_{BT} \rangle$ . The initial values for the environmental factors (measured in milliseconds) are  $u_{AL} = 7.5$ ,  $u_{AH} = 20$ ,  $u_{BL} = 3.75$ , and  $u_{BH} = 10$ . The experiment is run under several regimes that differ on how environmental factors evolve.

The experiment assumes that the service composition  $A$  has to finish in at most  $T_{\max} = 25,000$  ms. Violations to this requirement are detected by monitoring with the help of the complexity functions and the environment bounds, as previously described. The monitor periodically builds a *running* upper / lower bound estimate of the remaining execution time based on the elapsed time and the complexity / environmental factor bounds, respectively. The monitor issues a warning (**Warn**) when the upper bound of the estimate for the end-time of the task exceeds  $T_{\max}$ , to flag the risk of the time constraint violation, and an alarm (**Alarm**) if the lower bound estimate exceeds  $T_{\max}$  to indicate that a violation of the time constraint is imminent under the current conditions. **Alarm** prevails over **Warn**.

We performed one hundred simulations for every value of  $n$  by randomly choosing, for each  $n$ , a concrete complexity value between the bounds for  $A$  and  $B$ . We measure the effectiveness of the approach by empirically assessing the frequency of violation given a warning/alarm status. Figure 6 shows alarm / warning / violation profiles for two environmental regimes. The one on the left simulates a system reconfiguration where environmental characteristics remain at their initial values until time  $T_{\max}/3$ , when their bounds suddenly double (i.e., delays increase). The second regime (right) simulates a gradual degradation of

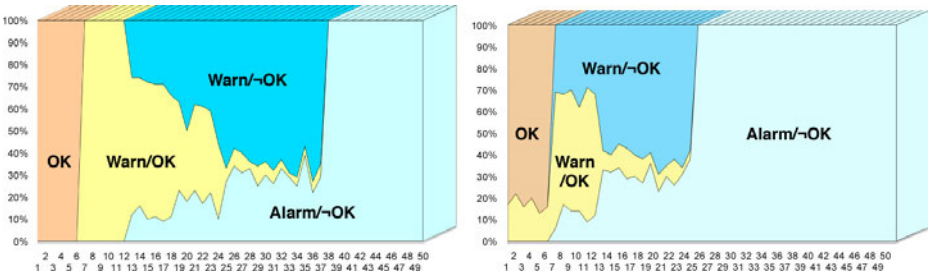


Fig. 6. Ratio of true and false positives for two environmental regimes

the system, where environmental bounds linearly increase over the period  $T_{\max}$  by a factor of four.

Under the first regime, composition executions for small values of  $n$  take little time to complete, so they comply with the time limit (marked by OK) and no alerts are raised. For slightly larger input sizes (e.g.  $n = 9$ ), executions still comply with the time limit, but warnings are raised (Warn/OK), since the monitor's estimate of the upper bound running time exceeds  $T_{\max}$ . As  $n$  increases, the number of false warning positives decreases in favor of the true warning positives (Warn/−OK), because the average running time increases and thus the possibility of execution being affected by sudden deterioration of the environment factors. In the same region (around  $n = 20$ ) some warnings start to be promoted to alarms, as the lower bound time estimates increasingly start to overshoot  $T_{\max}$ . These cases are marked with Alarm/−OK, and they are all true positives, since the system degrades monotonically (things never get better). Further increases in  $n$  (to around 30) lead to rapid disappearance of the false warning positives. After  $n = 38$ , all executions fall into Alarm/−OK, because the monitor is always able to detect ahead of time that the lower execution time bound overshoots the time limit.

In the second regime in Figure 6 featuring a gradual degradation, the upper execution time bound overshoots  $T_{\max}$  in some cases even for very small input sizes (e.g.  $n = 3$ ). A warning is then raised although no actual violations happen (executions are OK). As  $n$  increases, a pattern similar to that in the first regime is followed. For large values of  $n$  (but before the point in which it happened in the previous regime) all alarms rightly correspond to the −OK case.

## 6 Concluding Remarks

We have sketched a resource analysis for service orchestrations based on a translation to an intermediate programming language for which complexity analyzers are available. The translation process approximates the behavior of the original process network in such a way that the analysis results (the cost functions) are valid for the original network. We have presented a mechanism to use these functions, together with environmental characteristics, to predict the future behavior of the system even when the environment deviates from its expected behavior. We have applied them to perform predictive monitoring and the approach has been validated with a simulation which detects when the complexity bounds and the actual (simulated) execution cross the deadline and extracts statistical data regarding the accuracy of the predictions.

## References

1. Jordan, D., et al.: Web Services Business Process Execution Language Version 2.0. Technical report, IBM, Microsoft, et. al (2007)
2. Zaha, J.M., Barros, A.P., Dumas, M., ter Hofstede, A.H.M.: Let's Dance: A Language for Service Behavior Modeling. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 145–162. Springer, Heidelberg (2006)

3. van der Aalst, W., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: *The Role of Business Processes in Service Oriented Architectures*. Dagstuhl Seminar Proceedings, vol. 06291 (2006)
4. Mukherjee, D., Jalote, P., Nanda, M.G.: Determining QoS of WS-BPEL Compositions. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) *ICSOC 2008*. LNCS, vol. 5364, pp. 378–393. Springer, Heidelberg (2008)
5. Wu, J., Yang, F.: A Model-Driven Approach for QoS Prediction of BPEL Processes. In: *ICSOC Workshops*, pp. 131–140 (2006)
6. Buccafurri, F., Meo, P.D., Fugini, M.G., Furnari, R., Goy, A., Lax, G., Lops, P., Modafferi, S., Pernici, B., Redavid, D., Semeraro, G., Ursino, D.: Analysis of QoS in Cooperative Services for Real Time Applications. *Data Knowledge Engineering* 67(3), 463–484 (2008)
7. Fugini, M.G., Pernici, B., Ramoni, F.: Quality Analysis of Composed Services through Fault Injection. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) *BPM Workshops 2007*. LNCS, vol. 4928, pp. 245–256. Springer, Heidelberg (2008)
8. Cardoso, J.: About the Data-Flow Complexity of Web Processes. In: *6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility*, pp. 67–74 (2005)
9. Cardoso, J.: Complexity analysis of BPEL web processes. *Software Process: Improvement and Practice* 12(1), 35–49 (2007)
10. Debray, S.K., Lin, N.W.: Cost Analysis of Logic Programs. *ACM Transactions on Programming Languages and Systems* 15(5), 826–875 (1993)
11. Canfora, G., Penta, M.D., Esposito, R., Villani, M.: An Approach for QoS-Aware Service Composition Based on Genetic Algorithms. In: *Proceedings of the 2005 conference on Genetic and Evolutionary Computation*, pp. 1069–1075. ACM, New York (2005)
12. Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering* 30(5), 311–327 (2004)
13. Chen, Y.P., Li, Z.Z., Jin, Q.X., Wang, C.: Study on QoS Driven Web Services Composition. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) *APWeb 2006*. LNCS, vol. 3841, pp. 702–707. Springer, Heidelberg (2006)
14. Ivanović, D., Carro, M., Hermenegildo, M., López, P., Mera, E.: Towards Data-Aware Cost-Driven Adaptation for Service Orchestrations. Technical Report CLIP5/2009.1, Technical University of Madrid (UPM) (March 2010)
15. Navas, J., Mera, E., López-García, P., Hermenegildo, M.: User-Definable Resource Bounds Analysis for Logic Programs. In: Dahl, V., Niemelä, I. (eds.) *ICLP 2007*. LNCS, vol. 4670, pp. 348–363. Springer, Heidelberg (2007)
16. Méndez-Lojo, M., Navas, J., Hermenegildo, M.: A Flexible (C)LP-Based Approach to the Analysis of Object-Oriented Programs. In: King, A. (ed.) *LOPSTR 2007*. LNCS, vol. 4915, pp. 154–168. Springer, Heidelberg (2008)
17. Hermenegildo, M.V., Bueno, F., Carro, M., López, P., Morales, J., Puebla, G.: An Overview of The Ciao Multiparadigm Language and Program Development Environment and its Design Philosophy. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) *Concurrency, Graphs and Models*. LNCS, vol. 5065, pp. 209–237. Springer, Heidelberg (2008)
18. Hermenegildo, M., Puebla, G., Bueno, F., López-García, P.: Integrated Program Debugging, Verification, and Optimization Using Abstract Interpretation (and The Ciao System Preprocessor). *Science of Computer Programming* 58(1–2), 115–140 (2005)

# Service Customization by Variability Modeling

Michael Stollberg and Marcel Muth

SAP Research

CEC Dresden, Germany

{michael.stollberg,marcel.muth}@sap.com

**Abstract.** The establishment of service orientation in industry determines the need for efficient engineering technologies that properly support the whole life cycle of service provision and consumption. One challenge is adequate support for service consumers for employing complex services in their individual application context, which becomes particularly important for large-scale enterprise technologies where generic services are designed for reuse in several business scenarios. This paper presents an approach for service customization by model-driven variability management. The variable aspects of the services are explicitly described on the basis of a metamodel. Upon this, service consumers can easily create personalized service variants that properly suit their specific context while the consistency for service invocation is maintained.

## 1 Introduction

In the last years, service orientation has become the dominating design principle for modern ICT technologies in industry as well as in the public sector. The aim is to exploit the enormous potential of services for enhancing the interoperability among systems and the reuse of implementations. In consequence, a steadily growing number of available services and service-based applications can be observed. The design, development, usage, and management of such solutions requires sophisticated engineering technologies that support the life cycles of service provision and service consumption in an efficient and integrated manner.

This is subject to the emerging discipline of service engineering. Numerous efforts in academia and industry have developed a wealth of techniques, methodologies, and tool support for this. However, existing solutions focus mainly on support for service providers, i.e. for the design, development, publication, and management of services. The consumption side – i.e. the support for service consumers for finding suitable services and integrating them into the specific target application – is often neglected. Existing technology support for this is mostly limited to low-level technical details, leaving the major part of the analysis and integration task for actually consuming services to manual inspection.

The limitations become obvious when considering the consumption of more complex services that commonly occur in real-world business applications. For example, consider the Enterprise Services that form the basis of SAP's modern service-based enterprise technology. These are designed in a generic manner and



cover several usage options, therewith becoming reusable in various business scenarios. On the other hand, their interfaces and usage conditions are considerably complex. Typically, a customer merely requires a subset or a specific flavor of the provided features. Hence, the Enterprise Services need to be configured and integrated in order to properly fit the customer's needs. This is a non-trivial task that requires both technical knowledge and business expertise. Due to the limited tool support, the customization requires massive human involvement and thus becomes a highly cost-intensive and error-prone task.

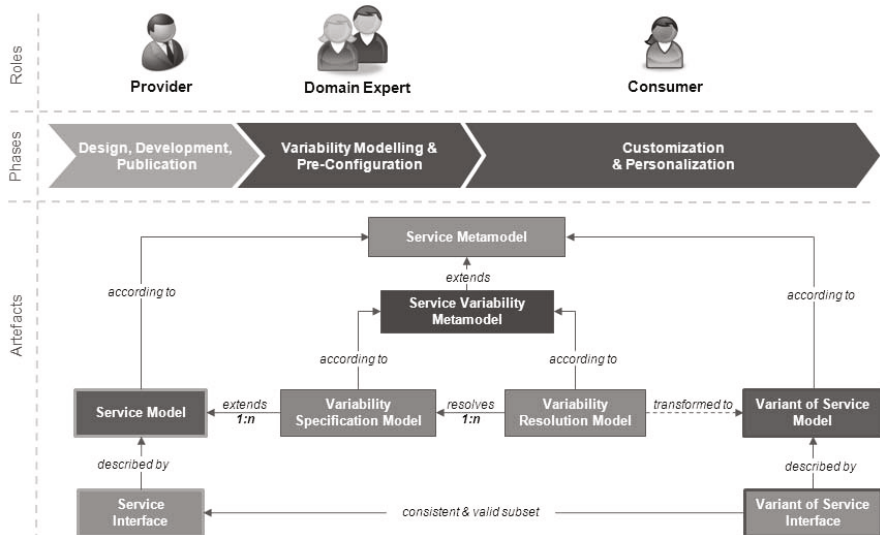
To overcome this, we present an approach for service customization by the creation of simplified variants that only expose those features that are relevant for the usage scenario of an individual consumer. These are defined on the basis of variability specification models that explicitly describe the usage conditions and constraints of the original service. To support model-driven engineering, we define a metamodel for describing the variable aspects of services, i.e. the mandatory and optional operations, properties of message types as well as their dependencies. We define an engineering process for service customization and provide tool support for specifying service variability models as well as for creating service variants via intuitive user interfaces. Finally, a technical interface is generated from the service variant model. This usually is significantly less complex than the interface of the original service while the consistency for the correct invocation is maintained. Furthermore, the simplified service description can serve as the basis for other service engineering techniques, e.g. for mash-up techniques that are mostly limited to services of limited complexity.

The paper is structured as follows. At first, Section 2 provides a concise overview of the approach and defines the engineering process for service customization. Section 3 specifies the metamodel for service variability modeling, and Section 4 defines the tool-supported procedures for service customization. Section 5 illustrates the techniques and tools for customizing an Enterprise Service, and Section 6 concludes the paper and outlines future work. An extended technical report with a detailed positioning in related work is provided in [5].

## 2 Overview

The following provides an overview of the approach for service customization. We define the central artifacts and roles involved in the process of providing and consuming customized services, outline the technical solution that is presented in detail in this paper, and motivate the need for such technologies.

Figure 1 provides a comprehensive overview, identifying the involved roles, phases, and relevant artifacts for the engineering process of providing, preparing, and consuming customized services. Abstracting from the concrete formation in industrial service engineering, we distinguish three main roles: the *Service Provider* develops and publishes services, the *Domain Expert* prepares them for customization by defining the variability specification model along with pre-configurations for respective user groups, and the *Service Consumer* customizes and personalizes the service in order to fit it into the specific application context.



**Fig. 1.** Service Customization – Roles, Phases, Artifacts

In the first phase, the Service Provider develops a service and publishes it in a repository. In the context of model-driven engineering, the service interface that defines the operations, messages, and endpoints is described by a Service Model on the basis of a metamodel (e.g. a WSDL metamodel or SoaML). In the second phase, the Domain Expert prepares the service for customization. For this, he creates a *Variability Specification Model* that describes the variable aspects of the service (i.e. the mandatory and optional operations, messages, and message types as well as dependencies). This is defined in accordance to the *Service Variability Metamodel* that defines the necessary constructs for modeling the variability of services. The Domain Expert might define multiple variability specification models for a service where each one is pre-configured for a particular application scenario (e.g. for specific industry sectors or geographical usage contexts). In the third phase, the Service Consumer adapts the service to the individual consumption context by defining a *Service Resolution Model*. For this, the variable aspects defined in the Variability Specification Model are resolved by selecting the desired features and by defining concrete values for the parameters that are not changed dynamically during the invocation. Finally, a Service Interface for the variant is generated: this only contains the selected features and represents a valid subset of the original service interface; the explicit variability modeling and the validation of the usage conditions throughout the customization process ensure the correct invocation of the service.

The technical solution for supporting service customization presented in this paper encompasses the specification of the Service Variability Metamodel and tools for supporting the creation of Variability Specification Models by Domain Experts as well as for Variability Resolution by Service Consumers. The overall

idea for enabling service customization by variability modeling is adopted from works on variability management in Software Product Line Engineering (SPLE, [1]), which however deal with different elements and thus employ different models and techniques. In order to ensure the efficiency of the service engineering process, we consider a model-driven approach where the service and variability models are defined on the architecture level (i.e. on the PIM level in terms of MDA [3]). Our prototype implementation works on SoaML [2] as the service metamodel; however, our Service Variability Metamodel is defined orthogonally to the base model, so that it can also be applied to other service metamodels.

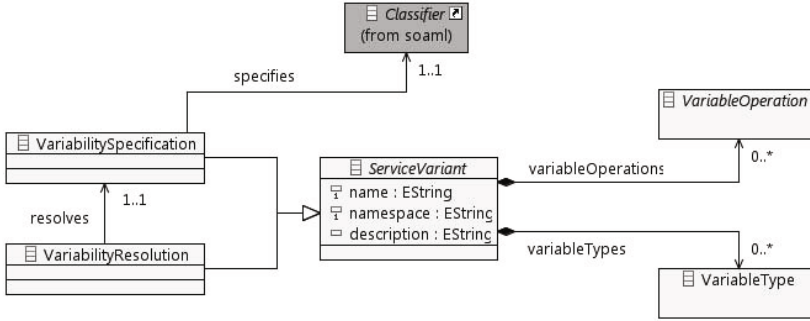
The motivation and business relevance of such a service customization technology results from the growing complexity of services, which particularly arises in the context of business applications. For illustration, consider a business service for creating and managing sales orders. A sales order is a relatively complex data object and, furthermore, its detailed structure is defined differently within the standards for different industries. In order to be reusable in various applications within several industries, a general purpose business service needs to support all options and specific features that are relevant for the targeted usage industries. In consequence, its interface becomes very complex regarding the number operations, the size of input- and output objects, and the conditions and constraints for proper and consistent consumption. A specific consumer typically only requires a subset of the provided features for his individual sales order management. However, due to the complexity, the general purpose service is not easy to understand, and its configuration for the individual needs of the consumer is a time consuming task that usually requires external support.

Our approach enables a step-wise reduction of the complexity and improves the technology support for customization. At first, a Domain Expert can define variants that are pre-configured for specific user groups, e.g. one variant for the automotive industry, one for steel production, and another one for the telecommunication sector. Furthermore, the detailed usage conditions for each variant are described explicitly in terms of a variability specification model. On this basis, a consumer can then define a personalized variant by selecting the desired features. The model-driven approach facilitates the abstraction from technical details as well as the provision of intuitive graphical user interfaces for modeling support, and the variability models ensure that the selections by the consumer are compliant with the usage conditions of the service. The generated technical interface for the consumer's variant is naturally significantly less complex than the one of the original sales-order service while it adheres to its usage conditions, so that a correct and consistent invocation is ensured.

Regarding related work, only a works address service customization on higher levels than technical configuration of which most merely provide methodological guidelines but no automated support. For this, our approach adopts concepts of variability modeling from SPLE (s.a.). Some recent works take a similar approach, but rather focus on architectural aspects than on variability management techniques for services for which our approach presents a complementary technique. We refer to [5] for a detailed positioning in related work.

### 3 Service Variability Metamodel

This section presents the metamodel for describing the variability of services. We first introduce the main elements, and then the constructs for variability modeling on different levels of service descriptions.



**Fig. 2.** Service Variability Metamodel – Main Elements

As shown in Figure 2, a *VariabilitySpecification* and *VariabilityResolution* serve as containers for the variability descriptions of the variable artefacts (operations and datatypes) of a specific service (which here is described by a SoaML Classifier [2]). The *VariabilitySpecification* contains the definition of all the variable aspects which are resolved by instances of a *ResolutionElement*. The variability description of a service are modeled with the help of four mechanisms shown at Figure 3:

1. Declaration of mandatory and optional elements, modeled by the boolean **required** property of *VariableElement* which serves as the the superclass for variability modeling on different levels of service descriptions.
2. Definition of dependencies among elements, modelled by the *Constraint* class with direct support for defining excluding and requiring constraints
3. Selection of desired features for a specific application context, modeled by the boolean **selected** property of *ResolutionElement*, and
4. Definition of fixed values that are changed within the application scenario or default values that are used for invocation when no concrete value is provided for, which is only applicable for *VariableProperty* and *PropertyResolutionElement*.

These central constructs are used to describe and handle the variability of services on various aspects. Currently, the metamodel covers the *operation level* where specialized variable elements for operations and their messages are defined and the *data level* where the variability of messages types can be explicitly defined on the level of properties. Due to space limitations, we refer to the extended technical report for further details on this [5].

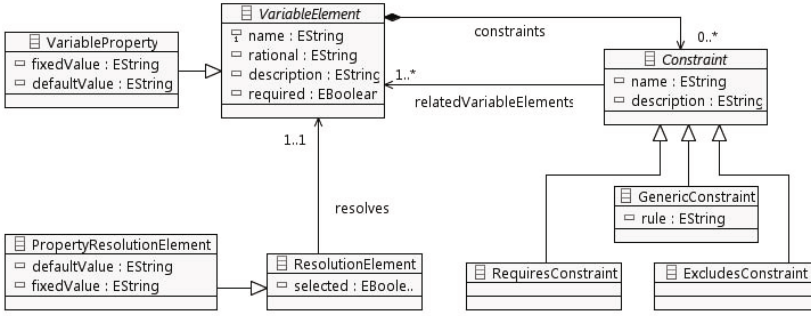


Fig. 3. Service Variability Metamodel – Variability Mechanisms

## 4 Service Customization Procedures and Tool Support

This section explains the techniques for service customization that work on the metamodel presented above, refining the overall engineering process introduced in Section 2. We here focus on the methodological aspects, while presenting the tooling support in the context of the illustrative example below in Section 5.

### 4.1 Service Variability Specification

As outlined above, the first step for enabling the customization of services is the creation of a variability specification model. This is performed by a domain expert in the second phase of the overall engineering process (cf. Figure 1 above). For this, the domain expert analyses the description of the base service, and creates a variability specification along with pre-configurations for the foreseen application context.

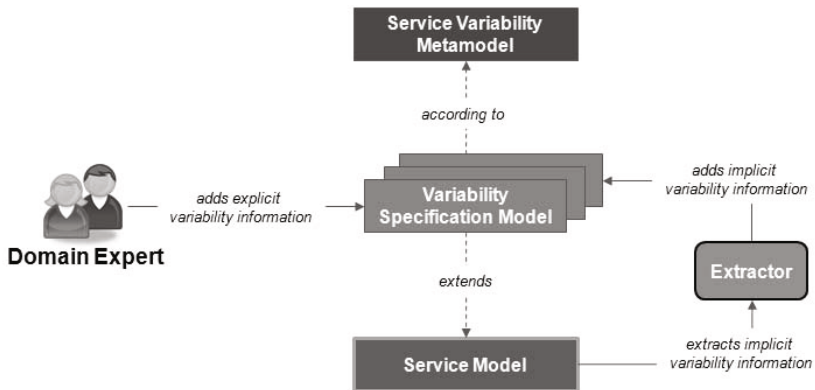


Fig. 4. Variability Specification Modeling

Figure 4 shows the tool-supported procedure for this. At first, the *Extractor* generates a skeleton of the variability specification model from the original service description, reducing the manual modeling effort. Then, the domain expert can refine the skeleton by adding further variability information, using the mechanisms supported by our metamodel. Note that there can be several variability specification models for a service where each one is pre-configured for a specific application scenario. The distinct variability specification models may define different usage conditions and mandatory and optional fields, as e.g. one industry standard requires fields that are irrelevant in another standard.

## 4.2 Service Variability Resolution

In the last phase, the consumer creates a service variant that suits the individual application context. For this, he chooses a suitable variability specification model of the service that has been previously prepared by a domain expert, resolve this by selecting the desired features and setting predefined values. From this the consumer generates a variant of the original service model which describes the interface for invoking the service.

Figure 5 shows the detailed procedure for this, which is supported by a graphical engineering tool that we shall present below. A variant is defined by selecting the desired features and defining default and fixed values for type properties and message parameters that will remain unchanged during the actual service invocation. The tool ensures that the user inputs comply with the conditions defined in the variability specification model. Finally, a variant of the original service model is generated from the resolution model. This is described in terms of a conventional service model, and thus can be used invoking the service.

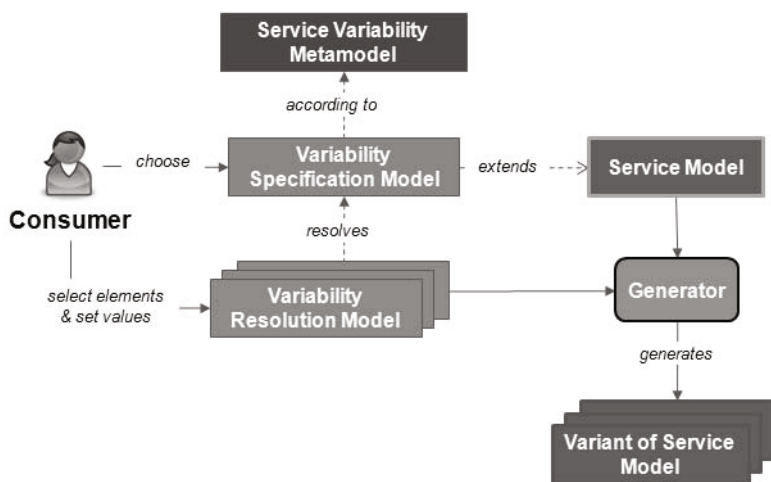


Fig. 5. Service Variability Resolution

## 5 Illustrative Example

This section illustrates the modeling techniques and procedures introduced above for customizing an SAP Enterprise Service, using our prototype which is implemented as a set of plugins based on the Eclipse Modeling Framework (EMF, see [www.eclipse.org/modeling/emf/](http://www.eclipse.org/modeling/emf/)).

### 5.1 Customizing the Goods Movement Enterprise Service

The Enterprise Service “Goods Movement” provides basic business facilities for managing the movement of goods, offering various usage options that work with complex data structures. For illustration, we will create a variant of the service that only contains the operations and necessary message types for the simple creation and reading of goods movement objects. Our prototype uses SoaML as a platform independent modeling language for the basic service descriptions [2]. Sufficient for demonstration purposes, we here work with already simplified data structures for the message types; the actual business objects used within SAP applications contain more than 100 nodes.

### 5.2 Variability Specification Modeling

As explained above, the first step in the service customization process is the creation of the variability specification model. Figure 6 shows our prototype for supporting domain experts in this task. This provides an editing facility for variability specification modeling with a tree-view representation and context-sensitive editing support for defining variability modeling elements, constraints, and bindings to the base model.

For the example, the domain expert defines the following explicit variability information. At first, the two operations for creating goods movement objects are grouped into a *ComplexVariableOperation*: conceptually, they present two

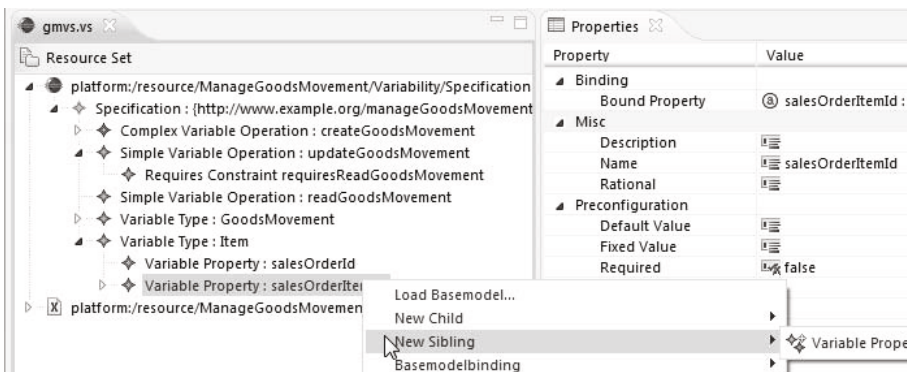


Fig. 6. Variability Specification Tool (Screenshot)

versions of the same operation that differ in the input- and output parameters; however, they are non-exclusive, thus the `multiple`-property is set to `false`. Secondly, the `update`-operation requires the `read`-operation; this is modeled by a `RequiresConstraint` (cf. Figure 2). Thirdly, the mandatory elements and their dependencies on the data level are defined. For instance, the usage of `salesOrderItem` is an attribute of the top-level type `item` and requires the `salesOrderId`. In addition, default values can be defined, e.g. setting country information to 'Germany' for pre-configuring the service variant for German customers.

### 5.3 Variability Resolution Tool

We now turn to the creation of the actual service variant. As explained above in Section 4.2, for this the consumer selects the desired operations and data elements, and defines default or fixed values for static data elements. From this, a conventional service model is generated for the personalized variant. To support consumers in this task, we provide a tool for selecting the desired features via a graphical user interface along with real-time validation of the dependencies and usage conditions defined in the variability specification model.

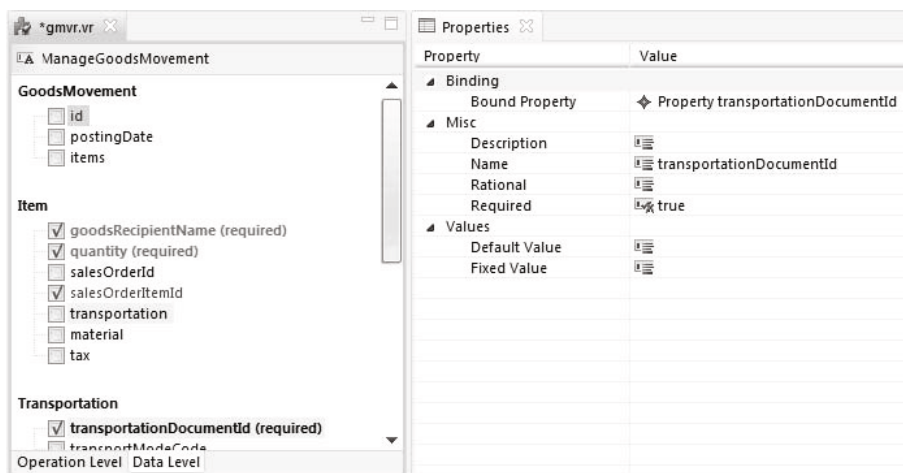


Fig. 7. Variability Resolution Tool (Screenshot)

Figure 7 shows our easy-to-use prototype tool for this. It is organized by tabs for variability resolution on different levels (currently operation and data level), and supports the user by standard metaphors for graphical user interface design [4]: checkboxes for selecting the desired elements, graphical accentuation of required elements, and colored display of constraint violations. The SoaML descriptions for the service variant is generated automatically, and stored in the respective folder of the EMF engineering project.



## 6 Conclusions and Future Work

This paper has presented an approach for service customization by the tool-supported creation of personalized variants on the basis of variability models that explicitly describe the variable aspects and usage conditions of services.

The need for such techniques arises from the growing number of complex services that are designed for reuse in various application contexts and deal with extensive data objects, which can be particularly found in service-oriented business applications. In order to facilitate the efficient and consumer-oriented consumption, we have proposed three-phased engineering process: domain experts prepare the services published by a provider for specific usage contexts by defining variability specification models; these explicitly describe the variable aspects, upon which consumers can easily create personalized variants that adopt the services to the specific context of the individual application scenarios.

In order to support model-driven development, we have defined a metamodel for describing the variability modeling for services on the basis of four mechanisms: the declaration of mandatory and optional elements with their dependencies, furthermore the selection of desired features as well as the definition of default and fixed values for a particular application scenario. We have developed tools for supporting domain experts in the specification of the service variability as well as consumers in the creation of individualized variants, and we have demonstrated them for customizing an Enterprise Service.

The presented technique enables the consumption of complex services in specific application scenarios. Its main benefits are the minimal modeling effort and ensuring a valid service invocation by the explicit variability modeling and thorough resolution validation. We however consider the presented solution as an initial prototype that shall be extended with additional aspects and further developed towards a comprehensive service customization technology in the future.

*Acknowledgements.* This paper is mainly based on works supported by EU funding under the SHAPE project (FP7 - 216408).

## References

1. Bayer, J., Gerard, S., Haugen, Ø., Mansell, J.X., Møller-Pedersen, B., Oldevik, J., Tessier, P., Thibault, J.-P., Widen, T.: Consolidated Product Line Variability Modeling. In: Käkölä, T., Dueñas, J.C. (eds.) *Software Product Lines - Research Issues in Engineering and Management*, pp. 195–241. Springer, Heidelberg (2006)
2. Berre, A. (ed.): *Service oriented architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS)*. Revised submission, OMG, 2008. OMG document: ad/2008-11-01
3. Frankel, D.S.: *Model Driven Architecture. Applying MDA to Enterprise Computing*. John Wiley & Sons, Chichester (2003)
4. Galitz, W.O.: *The Essential Guide to User Interface Design – An Introduction to GUI Design Principles and Techniques*, 3rd edn. Wiley, Chichester (2007)
5. Stollberg, M., Muth, M.: *Service Customization by Variability Modeling*. Extended Technical Report (2010), <http://www.michael-stollberg.de>

# Towards a Quality Model for Choreography<sup>\*</sup>

Michele Mancioffi<sup>1</sup>, Mikhail Perepletchikov<sup>2</sup>,  
Caspar Ryan<sup>2</sup>, Willem-Jan van den Heuvel<sup>1</sup>, and Mike P. Papazoglou<sup>1</sup>

<sup>1</sup> European Research Institute in Services Science (ERISS),  
Tilburg University, The Netherlands  
{m.mancioffi,wjheuvel,mikep}@uvt.nl  
<sup>2</sup> School of Computer Science and IT,  
RMIT University, Australia  
{mikhail.perepletchikov,caspar.ryan}@rmit.edu.au

**Abstract.** Quality attributes of software products like maintainability and reliability have been widely studied in the Software Engineering literature. Their understanding has proven instrumental for developing best practices and tooling support that ultimately result in higher-quality software.

In this paper we investigate external quality attributes (i.e. aspects of quality visible to the end user) of service choreographies. Service choreographies are service compositions that specify completely distributed, message-based interactions among services. Our work is a first step towards the definition of a quality model for service choreographies.

## 1 Introduction

Service Oriented Architecture (SOA) envisions software services that provide functionalities to each other and to consumers. Service compositions aggregate existing services to create value-added ones. Service choreography (choreography, for brevity) is a service composition paradigm that specifies conversations among the composed services, i.e. its participants, from a *global* perspective [1]. Participants in a choreography are independent agents that interact with each other by exchanging messages. In the practice of SOA, choreographies can be specified with modeling notations like Web Service Choreography Description Language (WS-CDL) and Business Process Modeling Notation (BPMN) v1.2.

One aspect that is missing in the state of the art of choreographies is the “big picture” of what are the relevant quality aspects of choreography specifications. The comprehensive specification and evaluation of appropriate quality attributes is a key factor in ensuring adequate quality in the production of any kind of artifacts like physical goods, software source code, etc. To this end, *quality models* have been produced over time for a number of general and specialized areas.

---

<sup>\*</sup> The research leading to these results has received funding from the European Community’s Seventh Framework Programme under the Network of Excellence S-Cube - Grant Agreement n° 215483. This work is also funded by the ARC (Australian Research Council), under Discovery scheme no. DP0988345.

Quality models are frameworks that provide concrete mechanisms for reasoning about quality in a systematic and objective manner [2]. They should be generic enough to cover the needs of different stakeholders, e.g. architects, designers, developers and end-users.

Quality models are composed of *quality attributes* and *quality metrics*. Quality attributes are desirable properties of the artifacts described by the quality model. They represent a checklist of quality concerns, and can be divided into *external* and *internal* attributes. External quality attributes describe how the quality of an artifact is perceived by its end users, e.g. in terms of **Reliability** or **Usability**. Internal quality attributes describe the quality of the artifact's design or implementation, e.g. in terms of **Coupling** and **Cohesion** [3]. Quality metrics are measurement mechanisms that quantitatively evaluate and/or predict internal and external quality attributes. While the external quality attributes of an artifact can be fully measured only after its development is completed, they can be *predicted* by measuring the internal quality attributes. For example, it has been empirically shown that the internal quality attributes of software design have a strong causal impact on external quality attributes in both Procedural [4] and Object-Oriented [5,6] software products. The prediction of the external quality attributes and the consequent improvement early in the Software Development LifeCycle (SDLC) results in a higher-quality software products [7,8].

Our goal is to unlock the benefits of a systematic and comprehensive quality management process for choreography specifications – i.e. descriptions of the structure of choreographies – by working towards a dedicated quality model. To this end, this paper presents our preliminary findings on the external quality attributes of choreographies. Our work is intended to complement the ongoing research work on quality attributes for choreography modeling notations (i.e. the syntaxes used to model choreographies) such as [9]. An overview of choreography and an outline of its aspects that must be considered for the formulation of the quality attributes is presented in Section 2. Section 3 presents an initial suite of external quality attributes specific to choreography. Finally, Section 4 concludes the paper by presenting our closing remarks and future directions.

## 2 Choreographies for Service-Oriented Architecture

This section presents the background knowledge on choreographies that we find relevant for the formulation of choreography quality attributes: their usage in SOA (Section 2.1), the elements they are made of (Section 2.2), and which of their properties are usually verified (Section 2.3). Due to space constraints we reference the interested reader to [10,11] for more information on the development and lifecycle of the choreographies.

### 2.1 Usage of Choreographies

Before treating what choreographies are used for in SOA, it is necessary to underline what choreographies are *not* used for: direct execution on the service

infrastructure. Unlike orchestrations, for example a Business Process Execution Language (BPEL) process deployed on and run by a BPEL engine, choreographies are not deployed on engines to be executed. A choreography is a service composition whose logic is distributed across and shared by its participants. The participants are independent agents (e.g. software services) possibly owned and run by different organizations. At run-time, each participant is responsible for the correct execution of its role in the choreography, i.e. its *expected* messaging behavior, in conjunction with what the other participants are doing. When the participants execute their roles in concert, they *enact* the choreography.

In the practice of SOA, choreographies are used for:

**Specification and communication:** choreographies are used to specify and communicate to the participants the requirements on the messaging interactions that they must fulfill, i.e. their roles.

**Facilitating the implementation of the participants:** during the process of implementing the participants (or the evolution of existing participants to fit a new choreography), the choreography can be used to generate (1) participant skeletons (i.e. partial implementations, see for example [12]) and (2) test cases for the participant implementations [13].

**Monitoring the enactments:** the monitoring facilities (e.g. integrated in the Enterprise Service Bus [14]) verify that the enactments do not violate the choreography.

## 2.2 Choreography Elements

The modeling notations for choreography (called in the remainder “choreography notations” or simply “notations” for brevity) vary greatly in the syntaxes and constructs they provide. Some have only an eXtensible Markup Language (XML) representation, e.g. WS-CDL. Others are graphical notations, e.g. Let’s Dance [15]. Regardless of the syntactical differences among the various choreography notations, the constructs they provide can be categorized as follows:

**Message ordering** constructs define the ordering of the message exchanges that take place among the participants. Message exchanges are usually specified as (1) the identifier of the message, (2) the identifiers of the participant that sends the message (the *sender*) and of one or more participants that receive it (the *recipients*), and possibly (3) time constraints (see for example [16],[17]). No technical information such as participants’ endpoints and interfaces or message types is provided at this level.

**Message types** constructs describe the structure of the messages that are exchanged among the participants, e.g. using XML Schema.

**Grounding** constructs provide the technical details necessary to enact the choreography, e.g. the endpoints of the participants and which transport protocols are supported to exchange messages (e.g. SOAP over HTTP).

Every choreography notation provides message ordering constructs. However, message types and grounding constructs are provided only by some (e.g. WS-CDL).

Another important aspect of a choreography notation is the messaging model it assumes, namely *synchronous* or *asynchronous*. In the synchronous case, the recipients must be ready to accept inbound messages when the sender delivers them. In asynchronous messaging, messages are posted by the sender and stored in queues held by the recipients. Each recipient can then decide if and when to consume the messages stored in its queue. Different types of asynchronous messaging make different assumptions on the size of the queues (finite or unbound), expiration time of the messages (a message is automatically removed from a queue if not consumed within a certain amount of time), preservation of message orders in the queues (whether the messages are stored in the queue in the same order in which they are sent), etc.

### 2.3 Properties of Choreographies

The distribution inherent to choreographies makes them susceptible to defects like deadlocks and the specification of roles that can not be faithfully implemented by the participants. Consequently, much effort has been devoted to the investigation of desirable properties such as deadlock freeness and realizability. In this section we briefly introduce the most relevant of these properties, providing the interested reader with references to more exhaustive material.

**Deadlock Freeness.** A deadlock occurs when the enactment of a choreography reaches a state that (1) is not final and (2) can not be left without violating the message ordering of the choreography. Analyses of choreographies for deadlock freeness usually build on top of model checking techniques (e.g. [18]) or on structural properties of the adopted choreography notation (e.g. [19]).

**Conformance.** A participant conforms to a choreography if its *business protocol* (i.e. the actual messaging behavior of the participant as *perceived* by the others [20,17]) is equivalent to the role played by that participant in the choreography. The actual notion of equivalence varies across different approaches in the literature, ranging from language equivalence – the business protocol and the role can execute all and only the same traces (see for example [21]) – to bisimulation (see e.g. [22]).

**Realizability.** Not all choreographies that can be specified can actually be correctly implemented by their participants. In some cases it is impossible for the participants to provide implementations that conform to the choreography. This problem occurs when a choreography specifies message exchanges between the participants that do not provide them with enough information on the global state of the enactments. The lack of information can lead the participants to unknowingly violate the choreography. In this case, the choreography is said to be not realizable [23,1] (a.k.a. enactable or enforceable).

Realizability is a paramount property of choreographies. Only realizable choreographies can really be enacted correctly in a completely distributed manner. It has been shown that non-realizable choreographies can be made realizable by further adding message exchanges that, during the enactment, make the participants align their perception on the global state [24].

### 3 Choreography Quality Attributes

This section presents our initial formulation of the external quality attributes for choreography. We focus on the quality of two aspects of a choreography: the choreography as a specification and how they way it is specified affects its enactments. Concerning the elements of choreographies (see Section 2.2), we limit ourselves to quality attributes related to the message type and message ordering constructs. We do not take into account the grounding of choreographies because the quality attributes it induces are closely related to the quality of the individual participant implementations (e.g. in terms of their reliability, availability, and trustworthiness [25,26]). Nonetheless, we understand the importance of the grounding in the context of choreography quality and plan to investigate this area in future work.

The external quality attributes we present and they way they are hierarchically organized in main- and sub-attributes are partly inspired by ISO/IEC 9126-1 [2]. Their formulation is based on an extensive literature review about choreographies and other quality models focusing on different, but related areas like web services and Service-Based Applications (SBAs).

Similarly to ISO/IEC 9126-1, all the main attributes include “compliance” sub-attributes, e.g. **Security Compliance**. Compliance sub-attributes capture the capability of the choreography to comply with standards, conventions, regulations, organizational policies and similar prescriptions that concern their main attributes. The particular prescriptions a choreography must comply with depend on the context of the definition of the choreography and of its enactments. For example, **Security Compliance** might capture the adherence of a certain choreography to the data privacy policies of the organizations that participate in it, as well as the compliance with regulations on data encryption and anonymization. All compliance sub-attributes are conceptually the same, and thus their definition under each attribute is omitted in the remainder.

The remainder of this section provides a detailed description of the external quality attributes and sub-attributes separately.

#### 3.1 Functionality

**Functionality** describes the quality of a choreography as a specification, i.e. how appropriate is the choreography to specify and communicate to the participants the requirements in terms of messaging interactions (see also Section 2.1).

**Suitability** is the capability of the choreography to satisfy its stated requirements in terms of message interactions among the participants, and how the message interactions make them achieve the desired goals.

**Accuracy** is the capability of the choreography during its enactments to make its participants achieve the correct or agreed results or effects with the needed degree of closeness. For example, the enactment of an accurate choreography will meet the business goals shared by the participants in an amount of time that is suitable for them (e.g. under one hour for the shipment of an online order).

**Understandability** and **Completeness** are related to the interpretation of the choreography as a specification by its participants. **Understandability** is the capability of the choreography to be univocally understood by its intended audience, e.g. the developers in charge of implementing the participants. **Completeness** is the capability of a choreography not to leave un- or underspecified aspects that, left open to the interpretation of the participants, might result in incompatibilities.

**Reusability** is the capability of a choreography specification to be reused in whole or in part for specifying other messaging interactions. The reuse of choreographies can reduce the cost implementing the participants, for example by facilitating the reuse of source code and test cases.

### 3.2 Security

**Security** is the capability of the choreography to prevent corruption and unauthorized access to information and data during both the implementation of the participants and the enactments.

**Confidentiality** captures the need of participants to (1) exchange privately sensitive data and information during the enactments of the choreography and (2) do not divulge sensitive information in the choreography specification.

**Non-Repudiation** and **Authenticity** are both facets of the issue of ensuring the identities of the participants in a choreography. **Non-Repudiation** is the capability of the choreography of preventing participants from refuting or repudiating messages they have sent or received during the enactments of a choreography (e.g. by using digital signatures and reception acknowledgements). **Authenticity** refers to the guarantees that the choreography specification provides in terms of authentication of messages, i.e. that the message actually comes from its supposed sender and not from another entity.

**Message Integrity** is the capability of the choreography to ensure the integrity of the messages that are exchanged by the participants during the enactments, e.g. by requiring mechanisms for detecting corrupted messages (for example hashing of the messages' payloads) and having them sent again.

Finally, **Message Reliability** deals with the level of reliability set by the choreography for the message exchanges. It should be noted that the actual implementation of the reliability mechanisms is usually delegated to the middleware employed by the participant implementations.

### 3.3 Efficiency

**Efficiency** is the capability of the choreography to specify interactions among the participants so that the enactments efficiently achieve the choreography's goals relative to the amount of resources used. The actual design of a choreography influences the efficiency of its enactments, e.g. in terms of average completion time. Aspects of efficiency of choreographies have been investigated under the umbrella of Quality of Service (QoS) (e.g. [27]). As anticipated at the beginning of the section, we do not consider the grounding of choreographies. On the basis

of the message type and ordering, we have identified two sub-attributes of efficiency of choreographies: **Message Type Efficiency** and **Message Ordering Efficiency**.

**Message Type Efficiency** relates the amount of data and information that are conveyed over message exchanges with the formatting of the messages (i.e. how the message types are specified) and the computing and bandwidth resources that are consumed to perform the message exchanges and process the messages (e.g. composition and compression by the sender, decompression and parsing by the recipients).

**Message Ordering Efficiency** captures how the ordering of the message exchanges in the choreography affects (1) the amount of time necessary to complete its enactments, and (2) the involvement (in terms of amounts of time) of the participants in the enactments.

### 3.4 Implementability

**Implementability** is the capability of the choreography to support and facilitate the implementation of its participants and their interactions during the enactments. The design and specific properties of a choreography influence how its participants are implemented.

**Realizability** captures the fact that the choreography is specified such that its participants can be implemented so to conform with the required messaging behaviors (see Section 2.3).

**Projectability** and **Testability** are related to how the choreography facilitates and supports the use of tools, possibly through Integrated Development Environments (IDEs), to simplify the implementation of its participants. The **Projectability** is the capability of choreography to be used to generate skeletons of the participant implementations (see Section 2.1). **Testability** is the capability of a choreography to be used to generate test cases for the participant implementations (see Section 2.1).

**Deadlock Freeness** refers to the absence of deadlocks in the enactments of the choreography (see Section 2.3).

**Monitorability** of a choreography refers to how its design affects the monitoring of its enactments (see Section 2.1). For example, the fact that each participant has only partial information during the enactment may complicate the monitoring. Participants that need to monitor parts of the choreography enactment might agree to share with each other key monitoring events.

### 3.5 Maintainability

Choreographies are specifications, and specifications are changed over time to reflect how their requirements evolve. **Maintainability** is the capability of the choreography to be changed over time to meet its evolving functional and non-functional requirements, e.g. by introducing or removing participants and message exchanges. We believe that the existing taxonomies of Software Maintenance (e.g. [28]) are largely applicable to choreography specifications.



**Analyzability** is the capability of a choreography to be analyzed using automatic tools like model checkers, simulators, and performance analyzers. The size and complexity of the choreography (e.g. amount of parallelism) affects the feasibility of most analysis methods because of the problem of state explosion. Moreover, the formality and expressiveness of the adopted notation could affect the analyzability of the choreographies modeled with it.

**Changeability** is the capability of the choreography to be modified with minimal effort and predictable results in terms of change propagation, i.e. how changes to the modified parts “ripple through” the choreography and affect the unmodified ones. Similarly to traditional software products, choreographies with a complicated and highly-coupled structure (i.e. message ordering) are likely harder to modify than simpler, more streamlined ones.

**Versionability** is the capability of the choreography to be clearly identified in the scope of version control, i.e. the management of different versions of the choreography that are produced during maintenance. Versionability can be defined at both specification- and enactment level. Specification-level versionability of a choreography is its capability of conveying information about its versioning in terms of (1) which other choreography specification it originates from (i.e. its *baseline*), (2) which changes have been applied to the baseline to lead to this version, and (3) when those changes have been applied, either in terms of their order or (better) the timestamp of their application. Alternatively, this can be achieved using revision control systems like SVN and GIT.

Enactment versionability is the capability of the choreography to have its version identified during its enactments. Participants need to identify the version of the choreography during its enactments in order to avoid issues resulting from the lack of backward- and forward compatibility across versions. A straightforward way of achieving this is embedding the choreography’s version identifier in the messages exchanged by the participants. More elaborate ways might be based on the related work on business process versionability, see for example [29].

It should be noted that, in addition to the proposed maintainability sub-attributes of choreographies, the process of maintaining a choreography also may depend on some of the sub-attributes of **Functionality**, namely **Understandability** and **Reusability**.

## 4 Conclusions and Future Work

In this work we have proposed our initial findings concerning external quality attributes for service choreographies, i.e. their quality aspects as perceived by the end users. Service choreography is a paradigm of service composition that specifies distributed, message-based interactions among multiple participants.

There is no work in the state of the art on quality models for choreographies. This paper takes a first step in that direction. We have overviewed the aspects of service choreography that are relevant for the definition of its quality model: usage in the practice of SOA, constructs, and verification. On the basis of this overview, we have proposed a hierarchy of external quality attributes that cover

five main quality aspects of choreographies: functionality as specifications, security, efficiency, implementability (i.e. the aspects of choreographies that influence how the participants are implemented), and maintainability.

Currently, we are considering different approaches for validating the proposed external quality attributes and sub-attributes. Additionally, we foresee several ways of extending the proposed model of external quality attributes for choreographies, for example the inclusion of attributes related to the grounding of the choreographies, costs and revenues sustained by the participants, and classical non-functional properties of distributed systems like recoverability, fault tolerance, and transactionality.

Finally, it is our intention to investigate the structural properties (or internal quality attributes) of choreographies (e.g. cohesion, coupling, and complexity) and define associated metrics for quantifying these properties in an objective and automated manner. Such metrics can be based on established metrics from the traditional software engineering field and the growing body of work on metrics for business processes.

## References

1. Su, J., Bultan, T., Fu, X., Zhao, X.: Towards a theory of web service choreographies. In: Dumas, M., Heckel, R. (eds.) *WS-FM 2007*. LNCS, vol. 4937, pp. 1–16. Springer, Heidelberg (2008)
2. ISO/EIC 9126-1:2001: Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva, Switzerland (2001)
3. Perepletchikov, M., Ryan, C., Frampton, K., Schmidt, H.W.: Formalising service-oriented design. *JSW* 3(2), 1–14 (2008)
4. Henry, S.M., Selig, C.: Predicting source-code complexity at the design stage. *IEEE Software* 7(2), 36–44 (1990)
5. Briand, L.C., Wüst, J., Daly, J.W., Porter, D.V.: A comprehensive empirical validation of design measures for object-oriented systems. In: *IEEE METRICS*, pp. 246–257. IEEE Computer Society, Los Alamitos (1998)
6. Alshayeb, M., Li, W.: An empirical validation of object-oriented metrics in two different iterative software processes. *IEEE Trans. Software Eng.* 29(11), 1043–1049 (2003)
7. Perepletchikov, M., Ryan, C., Frampton, K.: Cohesion metrics for predicting maintainability of service-oriented software. In: *QSIC*, pp. 328–335. IEEE Computer Society, Los Alamitos (2007)
8. Perepletchikov, M., Ryan, C., Frampton, K., Tari, Z.: Coupling metrics for predicting maintainability in service-oriented designs. In: *ASWEC*, pp. 329–340. IEEE Computer Society, Los Alamitos (2007)
9. Decker, G.: Design and analysis of process choreographies. PhD thesis, Hasso Plattner Institute, University of Potsdam (June 2009)
10. Decker, G., Riegen, M.V.: Scenarios and techniques for choreography design. In: Abramowicz, W. (ed.) *BIS 2007*. LNCS, vol. 4439, pp. 121–132. Springer, Heidelberg (2007)
11. Decker, G., Kopp, O., Barros, A.P.: An introduction to service choreographies (Servicechoreographien - eine Einführung). *it - Information Technology* 50(2), 122–127 (2008)

12. Mendling, J., Hafner, M.: From inter-organizational workflows to process execution: Generating BPEL from WS-CDL. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 506–515. Springer, Heidelberg (2005)
13. Wieczorek, S., Roth, A., Stefanescu, A., Charfi, A.: Precise steps for choreography modeling for SOA validation and verification. In: SOSE, pp. 148–153. IEEE Computer Society, Los Alamitos (2008)
14. Kopp, O., van Lessen, T., Nitzsche, J.: The need for a choreography-aware service bus. In: YR-SOC 2008, pp. 28–34 (June 2008) (Online)
15. Zaha, J.M., Barros, A.P., Dumas, M., ter Hofstede, A.H.M.: Let's Dance: A language for service behavior modeling. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 145–162. Springer, Heidelberg (2006)
16. Kazhamiak, R., Pandya, P.K., Pistore, M.: Timed modelling and analysis in web service compositions. In: ARES, pp. 840–846. IEEE Computer Society, Los Alamitos (2006)
17. Mancioppi, M., Carro, M., van den Heuvel, W.J., Papazoglou, M.P.: Sound multi-party business protocols for service networks. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 302–316. Springer, Heidelberg (2008)
18. Lohmann, N., Kopp, O., Leymann, F., Reisig, W.: Analyzing BPEL4Chor: Verification and participant synthesis. In: Dumas, M., Heckel, R. (eds.) WS-FM 2007. LNCS, vol. 4937, pp. 46–60. Springer, Heidelberg (2008)
19. Qiu, Z., Zhao, X., Cai, C., Yang, H.: Towards the theoretical foundation of choreography. In: WWW, pp. 973–982. ACM, New York (2007)
20. Ponge, J., Benatallah, B., Casati, F., Toumani, F.: Fine-grained compatibility and replaceability analysis of timed web service protocols. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 599–614. Springer, Heidelberg (2007)
21. Baldoni, M., Baroglio, C., Martelli, A., Patti, V., Schifanella, C.: Verifying the conformance of web services to global interaction protocols: A first step. In: Bravetti, M., Kloul, L., Zavattaro, G. (eds.) EPEW/WS-EM 2005. LNCS, vol. 3670, pp. 257–271. Springer, Heidelberg (2005)
22. Busi, N., Gorrieri, R., Guidi, C., Lucchi, R., Zavattaro, G.: Choreography and orchestration conformance for system design. In: Ciancarini, P., Wiklicky, H. (eds.) COORDINATION 2006. LNCS, vol. 4038, pp. 63–81. Springer, Heidelberg (2006)
23. Fu, X., Bultan, T., Su, J.: Realizability of conversation protocols with message contents. *Int. J. Web Service Res.* 2(4), 68–93 (2005)
24. Salaiin, G., Bultan, T.: Realizability of choreographies using process algebra encodings. In: Leuschel, M., Wehrheim, H. (eds.) IFM 2009. LNCS, vol. 5423, pp. 167–182. Springer, Heidelberg (2009)
25. Ran, S.: A model for web services discovery with QoS. *SIGecom Exchanges* 4(1), 1–10 (2003)
26. Gehlert, A., Metzger, A.: Quality reference model for SBA. Contractual Deliverable CD-JRA-1.3.2, S-Cube Consortium (March 2009), <http://www.s-cube-network.eu/results/deliverables>
27. Zhao, X., Cai, C., Yang, H., Qiu, Z.: A QoS view of web service choreography. In: ICEBE, pp. 607–611. IEEE Computer Society, Los Alamitos (2007)
28. Chapin, N., Hale, J.E., Khan, K.M., Ramil, J.F., Tan, W.G.: Types of software evolution and software maintenance. *Journal of Software Maintenance* 13(1), 3–30 (2001)
29. Juric, M.B., Sasa, A., Rozman, I.: WS-BPEL extensions for versioning. *Information and Software Technology* 51(8), 1261–1274 (2009)

# Towards a Conceptual Framework for Legacy to SOA Migration

Maryam Razavian and Patricia Lago

Department of Computer Science, VU University Amsterdam, the Netherlands  
m.razavian@few.vu.nl, patricia@cs.vu.nl

**Abstract.** Migration of legacy assets to SOA embodies a key challenge of service engineering, the rehabilitation of pre-existing enterprise assets into a service based system. As there is still little conceptual characterization of what the legacy to SOA migration process entails it is difficult to understand, compare and assess different approaches. This paper therefore proposes a conceptual framework embracing a holistic illustration of such a migration process. We describe what such migration process entails and what distinct conceptual elements systematically define the process. Based on the constituting conceptual elements, the framework which is considered as a basis for understanding and assessment of different approaches is proposed. Finally, the role of our migration framework in positioning and assessing the existing methods, is discussed.

## 1 Introduction

One of the key features of the service oriented paradigm is to facilitate reuse of business functions provided by legacy systems. The main motivation behind the modernization of legacy systems to SOA is to achieve the advantages offered by SOA and while reusing the embedded functionalities in the legacy systems. Although some characteristics of the SOA paradigm, such as support of loosely coupled services and interoperability, make the service enabling of legacy systems look to be straightforward, it constitutes a key challenge of service design.

Recently, migration of legacy systems to SOA has caught a lot of attention in both research and industry. A vast body of work in this area addresses exposing legacy code as (web) services [1,2], typically, focusing on implementation aspects of migration and usually covering techniques to alter a segment of legacy code to web services. A second family of approaches aims to cover the whole migration process. These approaches are comprised of two main sub processes: top-down service development and bottom-up service extraction. Planning of migration is the main focus of some other group of approaches [3,4]. Basically the feasibility of migration process is assessed on the basis of business drivers and characteristics of the legacy system.

These methods reflect different perspectives on SOA migration. They mainly differ in the way they provide solutions for two challenging problems of what can be migrated (i.e. the legacy elements) and how the migration is performed (i.e. the migration process). However, there is still little conceptual characterization

of what the legacy to SOA migration process entails. As a result, a common understanding of the SOA migration is difficult to achieve. Furthermore, the lack of study on the state of the art makes the understanding, comparison and analysis of existing methods especially difficult. To solve this problem, we need to establish a framework for SOA migration which facilitates achieving a general understanding on SOA migration process.

The question we address in this paper is what this framework entails. As an answer, based on the definition of SOA migration, we propose a framework which embraces a holistic illustration of the SOA migration process, along with the distinct conceptual elements involved in such a process. This framework facilitates the representation of the SOA migration processes in a unified manner and therefore provides the basis for their comparison and analysis. As such, different migration approaches may be mapped to a portion of (or all) the framework.

The remainder of the paper is organized as follows. In Section 2, the proposed SOA migration framework is described. Section 3, discusses the use of the proposed framework. Finally, Section 4 concludes the paper.

## 2 The SOA Migration Framework

The SOA migration framework addresses the question of “what does the migration of legacy systems to SOA entail”. In [5] migration is defined as a modernization technique that moves the system to a new platform while retaining the original system data and functionality. According to Chikofsky [6], reengineering is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form. The commonalities among these two definitions are considerable. In practice, the notions of “legacy migration”, “integration” and “architectural recovery”, which all deal with legacy applications, are considered as approaches to reengineering. Following this line of thought, we consider the problem of migration of legacy systems to SOA a reengineering problem.

According to [7] any type of reengineering consists of three basic reengineering processes: 1) analysis of an existing system, 2) logical transformation, and 3) development of a new system. In the context of architectural recovery, a conceptual “horseshoe” model has been developed by the Software Engineering Institute, which distinguishes different levels of reengineering analysis and provides a foundation for transformations at each level, especially for transformations to the architectural level [7]. Given that migration is considered as a reengineering process and that the horseshoe model is a generally accepted conceptual model for reengineering, we propose an extended form of the horseshoe model as a holistic model of the migration process.

Fig. 1 illustrates our proposed SOA migration framework (so called SOA-MF) for the migration process. Here, a migration process follows a horseshoe model by first recovering the lost abstractions and eliciting the legacy fragments that are suitable for migration to SOA (left side), altering and reshaping the legacy abstractions to service based abstractions (transformations in the middle

area), and finally, renovating the target system based on transformed abstractions as well as new requirements (right side). To adequately illustrate the notion of legacy migration, we should recognize the corresponding key characterizing concepts. Migration processes generally consist of three *sub-processes*, reverse engineering, transformation and forward engineering (thick arrows in Fig. 1). We argue that, the migration process is a transformation of the representations or *artifacts* (parallelograms in Fig. 1), that are carried out by means of a certain *activity* (rounded rectangles in Fig. 1). Activities can be supported by different types of *knowledge* as a resource or as the span of information that they should handle. Finally, the process of moving and mapping among the artifacts within the overall migration task, which graphically resembles a horseshoe, can be performed at different levels of abstraction ranging from “code-level” to “enterprise-level”.

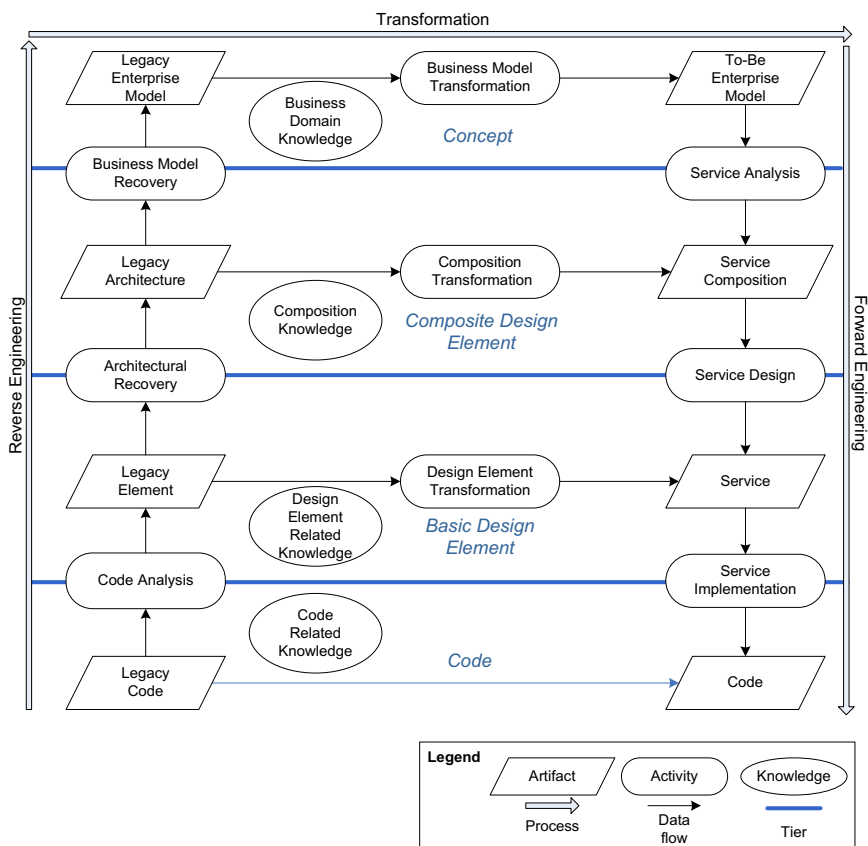


Fig. 1. Overview of SOA Migration Framework

## 2.1 Conceptual Elements of SOA-MF

In the following, we provide a systematic presentation of the main building blocks of the migration framework, so called conceptual elements. Below each conceptual element is presented along with its role in the migration process.

### I. Process

According to Fayad, software processes define what needs to be done in a software development effort and how it is done [8]. Similarly, the migration process could be defined as the set of tasks carried out during migration. As mentioned, the migration process is also divided into three sub processes including reverse engineering, transformation and forward engineering. These sub-processes are carried out through the four levels of abstraction (discussed in Section 2.3). **Reverse engineering** is the process of analyzing the existing system to identify the system's structure, functionality and behavior and to create representation of the system in another form or at a higher level of abstraction [6]. It should be noted that reverse engineering is the matter of examination and not a process of change. More precisely, "meaningful higher level abstractions" of the existing system are identified based on bodies of knowledge addressing, for instance, the domain, technology and architecture. From the migration point of view, the main goal of the reverse engineering process is to reach an understanding of the legacy system to the extent to identify the best candidates among the existing legacy elements for migration to SOA. Here, legacy elements are inherently the "meaningful higher level abstraction", recovered by means of reverse engineering techniques. In other words, the output of reverse engineering process is a number of legacy assets (in different levels of abstraction) extracted by means of the reverse engineering techniques and are suitable for the migration purpose. From the life-cycle coverage perspective, this process could start from existing implementation and continue with extracting the design entities, recovering the architecture and recapturing abstractions in requirements or business models.

**Transformation** is the process of restructuring one representation form to another at the same level of abstraction [6]. This transformation could be in the form of reshaping design elements, restructuring the architecture and/or altering business models and business strategies. It should be noted that each of these transformations belongs to a specific level of abstraction. In other words, based on the level of abstraction of the initial and the target representations, different types of transformations are carried out. From the SOA migration perspective, transformation process embraces migration of legacy assets to service based assets and is performed through a set of activities associated to different levels of abstraction. Accordingly, transformation in a particular migration approach could be performed within just one level or across number of them. Section 2.2 provides more description of different types of transformation activities. It is worth mentioning that transformation does not necessarily mean modification because of new requirements. Based on [6], transformation is altering "how", without altering "what".

During **forward engineering**, the subject system is renovated on the basis of the new requirements and goals offered by the target service based environment as well as the target artifacts produced during the transformation process. Specifically, by considering the new requirements, goals, etc., and also the business model, service composition and/or services produced during the transformation process the service-based system is implemented in a top down manner.

## II. Artifact

According to Conradi et al. [9], artifacts are the products of a process. Here, an artifact represents any product or “raw material” (i.e. models, architecture, piece of code) extracted, transformed or developed during each of reverse engineering, transformation or forward engineering processes.

## III. Activity

An activity is an atomic or composite production step of a process which aims to generate or modify a given set of artifacts [9]. Activities indicate regarding steps of what must be done during each of reverse engineering, transformation and forward engineering processes.

## IV. Knowledge

The Oxford Dictionary defines knowledge as “specific information, facts or intelligence about something”. Here, we define knowledge as a term used to describe the whole spectrum of content for the following concepts concerning the migration process: data, models, procedures, techniques, principles, and context. These concepts are the set of information about software systems and business domain which shape the migration process (given that they provide inputs to the migration activities). Different approaches are distinguished based on types of knowledge they exploit during the migration process. In [7] the set of levels of information about software system from the source code level to the architectural level is proposed. We follow the same view and classify the bodies of knowledge exploited within migration based on their associated level of abstraction. More precisely, where knowledge originates from (the level of abstraction), indicates the type of knowledge. As a consequence, the corresponding categories of knowledge are as follows: code-related knowledge, design element-related knowledge, composition knowledge and business domain knowledge. For example, code grammars and models are categorized as code-related knowledge and are used within the reverse engineering sub-process. Cohesion is considered as a design element-related knowledge since it addresses a principle about a single element (i.e. module, component or service). Architectural patterns and styles are of type composition knowledge and, finally, business rules, risks, benefits and plans are categorized as a business domain type of knowledge.



## 2.2 SOA-MF Description: Process Phases

Migration processes can be built in many different ways, i.e. it is not possible to identify one universal migration process. However, a basic general migration process constituting the skeleton of migration process at its most complete form can be defined. SOA-MF is devised based on the same scheme (skeleton of a complete migration process), that migration approaches cover a portion (or all) of.

So far, we have discussed our definition of migration process and the conceptual elements providing basis to describe the process. Now we describe SOA-MF addressing the migration of legacy systems to SOA. The framework illustrates the migration process together with details of the artifacts included, activities carried out and types of knowledge exploited within each of migration sub-processes. The graphical representations of the conceptual elements are depicted in Fig. 1. The sub-processes, activities, artifacts and knowledge elements are depicted by thick arrows, rounded rectangles and parallelograms respectively. In this section, the role of each activity in the whole migration process and the associated input and output artifacts and the knowledge exploited by the activity are described. The conceptual elements comprising the framework are in italic.

### I. Reverse Engineering

In its most basic form, reverse engineering starts from analyzing the legacy code within the *code analysis* activity. This activity aims to extract the legacy elements identified as candidates for transformation to services. Code analysis techniques such as graph-based analysis, lexical analysis, code querying, etc., are considered instances of the code analysis activity. With regard to this activity, the input artifact is the legacy code while the output consists of set of legacy elements (which could be in the form of components, modules, segments of code, etc.). The extraction of legacy elements from code is influenced by involvement of *code related knowledge* (such as code grammar and model) as well as bodies of knowledge addressing higher level concepts (such as *business domain knowledge*).

So far, within the reverse engineering sub process, the extracted legacy elements are inherently design entities recaptured by means of reverse engineering techniques. However, we could go one step further and recapture the meaningful compositions of these legacy elements within the *architectural recovery* activity. Here, the *composition knowledge* such as architectural patterns and architectural styles are involved in identification of the architectural elements and their associated relationships.

Finally, the *legacy enterprise model* is extracted during the *business model recovery* activity. The inputs to this activity are the legacy architecture and the existing *business domain knowledge* such as business rules, business processes, etc.

### II. Transformation

As mentioned, transformation encompasses process of restructuring one representation form to another at the same level of abstraction. The activities of

*design element transformation*, *composition transformation* and *business model transformation*, respectively, realize the tasks of reshaping design elements, restructuring the architecture and altering business models and business strategies. The bridge part of the SOA-MF represents these transformations.

*Design element transformation* activity is typically performed to move the encapsulation of the legacy elements (extracted during the reverse engineering process) to services. Most of the wrapping techniques fall in this category of transformations. The input artifact to this activity is the legacy element (i.e. module, component or segment of a code) and the output artifact is basically a service. Types of knowledge which are inputs to design element transformation are: code related and design element knowledge.

*Composition transformation* activity embodies transformation of the legacy architecture (input artifact) to service compositions (output artifact) in terms of changing the allocation of functionality, their topology, etc. In other words, components and connectors are transformed to a service composition embracing services and relationships among them. Pattern based architectural transformation techniques fall in this category of transformations. Commonly, this activity exploits *composition knowledge* and *design element knowledge* as inputs to perform the transformation. For instance, architectural patterns, service composition patterns and service inventory patterns (i.e. *composition knowledge*) are used within the composition transformation activity.

During *business model transformation* activity the existing business model is transformed to a to-be business model based on new requirements as well as opportunities offered by service based systems. Here, existing business rules, business processes and strategies which are partially embedded in the legacy enterprise model are transformed to new ones to form the basis for development of service based system. The input artifact to this activity is *legacy enterprise model*, whereas the to-be enterprise model forms the output. The business model transformation activity is assisted by the *business domain knowledge* such as business rules, risks, benefits and plans.

### III. Forward Engineering

In its most complete form, the forward engineering process starts from the *to-be enterprise model*. During *service analysis*, based on the *to-be enterprise model* a set of candidate service compositions which conceptualize the business processes are identified. Afterwards, the candidate *service compositions* are consolidated with service compositions identified during *composition transformation* activity. This activity is succeeded by *service design* during which the renovated services are designed based on the consolidated candidate service compositions. Similar to service compositions, candidate services are merged with the services identified during *design element transformation* activity (of transformation process). Finally, during *service implementation* the service design is transformed to code.

### 2.3 Tiers

Transformation process embraces reshaping of an artifact of the existing legacy system to another artifact in the service oriented system. Considering four levels of abstraction including code, basic design elements, composite design element and concept, we argue that usually the as-is artifact (in reverse engineering process), the to-be artifact (in the forward engineering process) and their associated transformation activity all reside in the same level of abstraction. In that case, if we consider the as-is and to-be artifacts as well as the transformation among them as a tier, we could characterize and classify a migration approach based on the tiers supported. Fig. 1 depicts the tiers distinguished from each other by solid lines. It should be noted that, in a sample migration process, different set of tiers could exist, which are not necessarily adjacent. Consider a migration method which covers the concept and design element tiers. This implies that the business model transformation and design element transformation activities are included in transformation process whereas no composition transformation is carried out. From another point of view, a transformation in higher level of abstraction may not entail the transformation in lower levels.

## 3 On the Role of SOA-MF

As mentioned previously, the main goal of the SOA-MF is understanding through classification and comparison of existing SOA migration methods. We argue that, SOA-MF is an intuitive graphical representation, which provides pieces of information to illustrate and characterize each existing migration process. More precisely, each migration process could be described based on the processes it supports, artifacts included, activities carried out, types of knowledge exploited and finally tiers they reside in. In other words, if we consider the SOA-MF model as a diagram, each migration process constitutes a portion of this diagram including the covered conceptual elements. Existing migration approaches use different terms and expressions for inherently similar tasks and concepts regarding migration. A general understanding of the migration methods could be reached by mapping and positioning the methods and their associated tasks and artifacts into a common framework. In the same vein, the SOA-MF model facilitates understanding and classification of existing migration methods and possibly provides the basis for analyzing their limitations and pitfalls. This is realized by identifying each method's associated portion of the diagram and comparing them based on their supported conceptual elements and their position related to the SOA-MF model (main diagram).

We have studied a number of SOA migration approaches, and mapped and positioned them on SOA-MF. The mapping of two of these approaches (SMART [3] and Sneed's [2]) on SOA-MF are presented here to clarify what the mappings imply and how it facilitates the understanding and comparison of the approaches.

Fig. 2, reflects the following features of SMART: from lifecycle-coverage perspective, the reverse engineering and the forward engineering sub-processes are not covered. As a result, recapturing the abstractions of existing legacy system

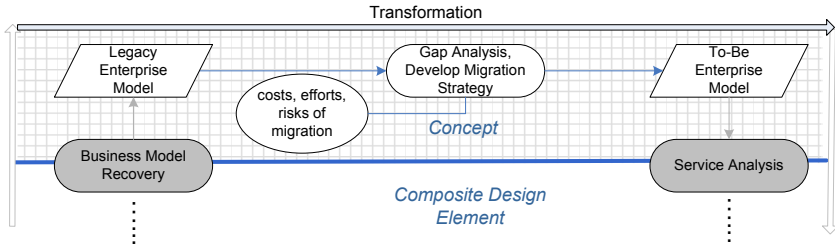


Fig. 2. Mapping of SMART on SOA-MF

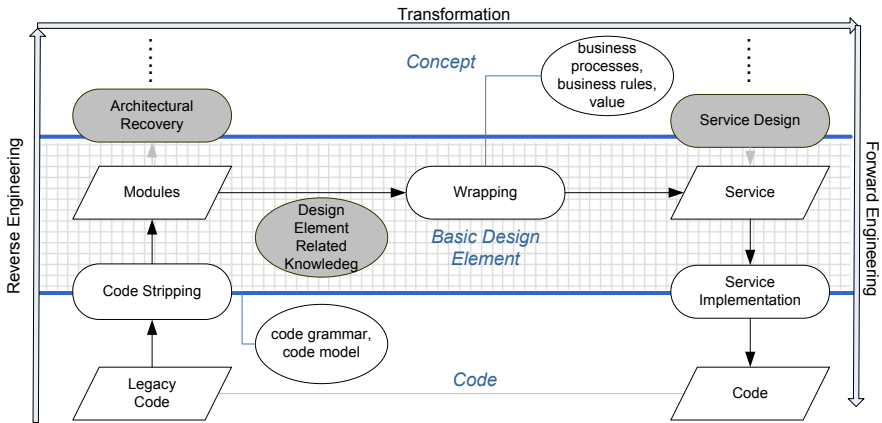


Fig. 3. Mapping of Sneed's approach on SOA-MF

are not addressed. In addition, development of the service based system is not covered. Transformation is carried out at concept level and embraces altering legacy enterprise model of the existing system to the to-be enterprise model using business domain level knowledge.

The following features of Sneed's migration approach can be extracted from the associated mapping on SOA-MF. The horseshoe like representation of this approach on SOA-MF illustrates that all three sub-processes of reverse engineering, transformation and forward engineering are carried out. The transformation occurs at basic design element level and migration is limited to altering modules to services, however, business domain knowledge facilitates the migration.

To sum up, SMART provides high level solutions for migration problem, while ignoring the technical details of legacy element extraction and service development. Whereas, Sneed's approach presents a migration process at lower level through focusing on technical details.

## 4 Conclusion

It is hard to understand and classify existing approaches in an emerging and still fuzzy research field like SOA migration. In the past few years, several SOA migration approaches have been introduced, each focusing on a specific perspective on SOA migration, and using own concepts (activities, artifacts, etc.) to represent migration.

This paper has presented a SOA migration framework (SOA-MF). SOA-MF facilitates to characterize and isolate the properties of migration approaches in terms of processes it supports, artifacts included, activities carried out, and types of knowledge exploited. A unified representation of different approaches is achieved by mapping them on SOA-MF, which also provides the basis for their comparison and analysis.

During experimentation with earlier versions of SOA-MF, we observed that the notion of tier plays an important role in positioning and classifying the various migration approaches. The tiers of SOA-MF covered by a specific SOA migration approach can explain the following aspects: the associated level of abstraction in which the transformation occurs and the transformations that entail lower level ones. A relevant classification of existing approaches can be achieved by considering tiers they cover in SOA-MF. For instance, SMART is dedicated to the concept tier category (in which just the concept tier is covered); while Sneed's method is regarded as a basic design element tier approach; and the process presented in [10] is dedicated to the category of migration processes covering all tiers.

As our future work, we will use SOA-MF in industrial organizations to elicit how industry performs SOA migration in practice and see if we can recognize generic patterns. Moreover, to evaluate and refine SOA-MF, we are also bringing it to perform a systematic literature review on existing SOA migration processes.

## Acknowledgment

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research and Development (Jacquard) program on Software Engineering Research via contract 638.001.206 SAPIENSA: Service-enAbling PreexIsting ENterprIse Assets; and the European Community's Seventh Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

## References

1. Aversano, L., Canfora, G., Cimitile, A., de Lucia, A.: Migrating legacy systems to the web: an experience report. In: CSMR 2001: Proc. of the 5th European Conference on Software Maintenance and Reengineering, Washington, DC, USA, p. 148. IEEE Computer Society, Los Alamitos (2001)
2. Sneed, H.M.: Integrating legacy software into a service oriented architecture. In: CSMR 2006: Proc. of the Conference on Software Maintenance and Reengineering, Washington, DC, USA, pp. 3–14. IEEE Computer Society, Los Alamitos (2006)

3. Lewis, G., Morris, E., Smith, D., O'Brien, L.: Service-oriented migration and reuse technique (smart). In: STEP 2005: Proc. of the 13th IEEE International Workshop on Software Technology and Engineering Practice, Washington, DC, USA, pp. 222–229. IEEE Computer Society, Los Alamitos (2005)
4. Umar, A., Zordan, A.: Reengineering for service oriented architectures: A strategic decision model for integration versus migration. *Journal of Systems and Software* 82(3), 448–462 (2009)
5. Bisbal, J., Lawless, D., Wu, B., Grimson, J.: Legacy information systems: Issues and directions. *IEEE Software* 16, 103–111 (1999)
6. Chikofsky, E.J., Cross II, J.H.: Reverse engineering and design recovery: A taxonomy. *IEEE Software* 7(1), 13–17 (1990)
7. Kazman, R., Woods, S.G., Carrière, S.J.: Requirements for integrating software architecture and reengineering models: CORUM II, 154 (1998)
8. Fayad, M.E.: Software development process: a necessary evil. *Commun. ACM* 40(9), 101–103 (1997)
9. Conradi, R., Fernström, C., Fuggetta, A.: A conceptual framework for evolving software processes. *SIGSOFT Softw. Eng. Notes* 18(4), 26–35 (1993)
10. Andreas Winter, J.Z.: Model-based migration to service-oriented architecture. In: The International Workshop on SOA Maintenance Evolution, SOAM 2007 (2007)

# MINERVA: Model drIveN and sERvice oRIented Framework for the Continuous Business Process improvement and relAted Tools

Andrea Delgado<sup>1</sup>, Francisco Ruiz<sup>2</sup>, Ignacio García-Rodríguez de Guzmán<sup>2</sup>,  
and Mario Piattini<sup>2</sup>

<sup>1</sup>Computer Science Institute, Faculty of Engineering,  
University of the Republica Julio Herrera y Reissig 565,  
CP 11300, Montevideo, Uruguay

<sup>2</sup>Technologies and IS Depto., Faculty of Computer Science,  
University of Castilla-La Mancha, Paseo de la Universidad No.4,  
CP 13071, Ciudad Real, España  
adelgado@fing.edu.uy,

{francisco.ruizg, ignacio.grodriguez, mario.piattini}@uclm.es

**Abstract.** The importance and benefits of Business Process Management (BPM) for organizations are nowadays broadly recognized, as not only the business area but also the information technology one are embracing and adopting the paradigm. The implementation of business processes as services helps in reducing the gap between these two areas, easing the communication and understanding of business needs. Although there is a general agreement on the benefits of the joint application of these two paradigms, some issues still need to be addressed; being a key one the automatic generation of services from business process models. In this article, we present MINERVA framework which applies Model Driven Development (MDD) and Service Oriented Computing (SOC) paradigms to business processes for the continuous business process improvement in organizations, supporting the different stages defined in the business process life cycle from modeling to evaluation of its execution.

**Keywords:** business process, Business Process Management (BPM), Service Oriented Computing (SOC), Model Driven Development (MDD), improvement.

## 1 Introduction

The progressive adoption of the Business Process Management (BPM) [1][2] paradigm by organizations, defined as the activities that organizations do to optimize or adapt their business processes to the new organizational needs, puts the spotlight on the business process lifecycle as defined in [3][4][5], and on tools and technologies to support each stage. A business process is defined as a set of activities performed in coordination in an organizational environment to reach a business objective [3]. BPM Systems (BPMS) are generic software systems driven by explicit representations of business processes to coordinate their execution [3]. The implementation of business processes as services helps in reducing the gap between business and Information

Technology (IT) areas, easing the communication and understanding of business needs. It also promotes the independence between the definition and modeling of business processes, and their implementation into a specific technology, allowing changes in each one with minimal impact on the other.

Service Oriented Computing (SOC) refers to software development based on services to support distribute low cost interoperable evolving and massive applications [6]. A service provides an implementation which provides business logic and data, a service contract which specifies the operations and pre and post conditions, and an interface to expose the functionality [7]. Service Oriented Architecture (SOA) [7][8] is a software architecture style which constitutes an specific realization of SOC, implemented in general by Web Services (WS) [9]. Model Driven Development (MDD) bases the software development on models, using as first order artifacts metamodels, models and languages which allow transformations between them [10][11]. Model Driven Architecture (MDA) [12] is a standard realization of MDD by the Object Management Group (OMG) [13], where the main characteristic is model transformation defined as the process of converting one model into another model of the same system.

MINERVA stands for "Model drIveN & sErvice oRiented framework for the continuous business process improvEment & relAted tools", and aims to provide a framework comprising methodologies, concepts and tools for the automated development of service oriented solutions from business processes in organizations, combining the application of SOC and MDD paradigms to business process. It is defined to support the business process lifecycle as defined in [3], covering the four phases of: Design and Analysis, Configuration, Enactment and Evaluation. The rest of the document is organized as follows: section 1 presents the MINERVA proposal describing its general elements, in section 2 the dimensions of MINERVA are detailed showing its main elements; in section 3 the related work is mentioned and finally in section 4 the conclusions and future work are discussed.

## 2 MINERVA Proposal

MINERVA (Model drIveN & sErvice oRiented framework for the continuous business process improvEment & relAted tools) constitutes a framework for the business process improvement based on the business process lifecycle [3] that is defined in three dimensions: conceptual, methodological [14] and tool support. It also takes into account the Business Process Maturity Model (BPMM) [15] OMG standard, and measures for the design [4] and execution of business processes [16], which will drive the improvement effort in the business process lifecycle. Starting from the modeling of business processes in the Business Process Modeling Notation (BPMN) [17], MINERVA automatically obtains from this model a services model expressed in Service Oriented Architecture Modeling Language (SoaML) [18] by means of Query/Views/Transformations (QVT) [19] transformations, to the execution of processes expressed in BPEL [20] or XPDLL [21] in a suitable process engine. Fig.1 shows the general framework for the derivation of services from business process.



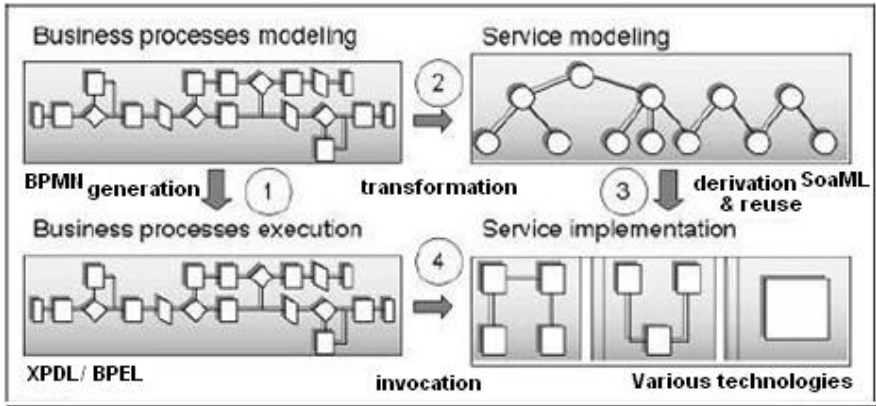


Fig. 1. Business processes and services relationships in MINERVA

Fig. 1 illustrates the steps for the automatic derivation of services from business processes, which is integrated in a service oriented methodology to guide the development of services from business processes. The proposal begins with the modeling of business processes in BPMN, shown on the top left corner in figure 1, which constitutes the input to the defined development process. From this model two complementary models are obtained: (1) automatic generation of the process specification in some executable/interpretable language like BPEL/XPDL, and (2) (almost) automatic generation of the services design model to realize the process in SoaML profile. From the services design model the implementation model is automatically obtained in step (3) for the selected technological platforms. Finally, both implementations are related to connect the business process execution: from business processes in BPEL/XPDL running in a process engine, components implementing the obtained services are invoked. To support the business process continuous improvement, it is necessary to add elements to the framework, as we need to collect data in the form of measures about relevant aspects, which can come from two sources: business process models and execution. We will integrate both types of measures: to verify models from which to obtain the services prior to the generation, and to analyze execution applying techniques such as Process Mining[22]. A continuous improvement process is being defined to support the improvement cycle to be followed to obtain the defined business goals of the organization.

### 3 Dimensions of MINERVA

The framework integrates elements into three dimensions: conceptual, methodological and tool support, comprising different elements that constitute the basis for the proposal, which are described in the following.

### 3.1 Conceptual Dimension

The conceptual dimension aims to define concepts, terminology and relations between them to be used all over the framework. An ontology defines the relevant elements (concepts, relationships) for the domain under study, providing meaning to the vocabulary and formalizing restrictions on its use [23]. The main objectives for the inclusion of an ontology in the MINERVA proposal are: to define, organize and reuse knowledge about concepts involved in the management of business processes and their life cycle, as well as their design and implementation based on services and the relationship between them. It also establishes the basis for defining transformations from business process to service metamodels as in Business Process Metamodel Definition (BPMD) [24] and SoaML.

Based on the business processes lifecycle [3], we have identified five groups for the main conceptual elements required in BPM. Therefore, sub-ontologies are defined for: modeling, simulation, execution, measurement and evaluation of business processes. To support service orientation, there are two main conceptual groups which define the sub-ontologies for service-oriented modeling and execution. The Service Oriented Modeling sub-ontology (SOMsO) corresponds to the Business Process Modeling sub-ontology (BPMsO) meaning that the elements from the first one trace to elements from the second one. On the other hand, the Business Process Execution (BPEsO) sub-ontology “use” the Service Oriented Execution sub-ontology (SOEsO), where an execution of a business process in a business process engine will invoke the execution of corresponding services. The Business Process Measuring sub-ontology (BPMEsO) integrates measures for business process models and execution [16] adding elements adapted from the Software Measurement Ontology (SMO) [25]. The Business Process Evaluation sub-ontology (BPEVsO) uses measuring and execution sub-ontologies, defining other elements like Process Mining for execution logs analysis. The Business Process Simulation sub-ontology (BPSsO) defines elements to simulate and understand various characteristics of models prior to their execution.

So far, we have an initial definition of the BPMsO and the SOMsO which we briefly describe in the following, the complete definition can be seen in [26]. For business process modeling, the main references are the BPMN and BPDM OMG standards from which their defined concepts were taken. In service orientation there are many sources, from which were evaluated: SoaML [18] from OMG, Web Services Architecture (WSA) [9] from W3C, Service Oriented Architecture Reference Model (SOA RM) [27] from OASIS, Service Oriented Architecture Ontology (SOA O) [28] from Open Group. As shown in Fig. 2, the BPMN known grouping for BP elements is used: flow objects, connector objects, swimlanes (pool/lanes) and artifacts. As a key element of the service model the Service concept is identified, which is composed of an Implementation providing the required functionalities, a Contract that specifies the Operations provided and an Interface that offers the functionality. Provider and Consumer Agents exchange the defined Messages. The main relations between the BPMsO and the SOMsO refer to the correspondence between its key elements, for example activities (simple, sub-process) correspond to services, and pools correspond to participants, among others.

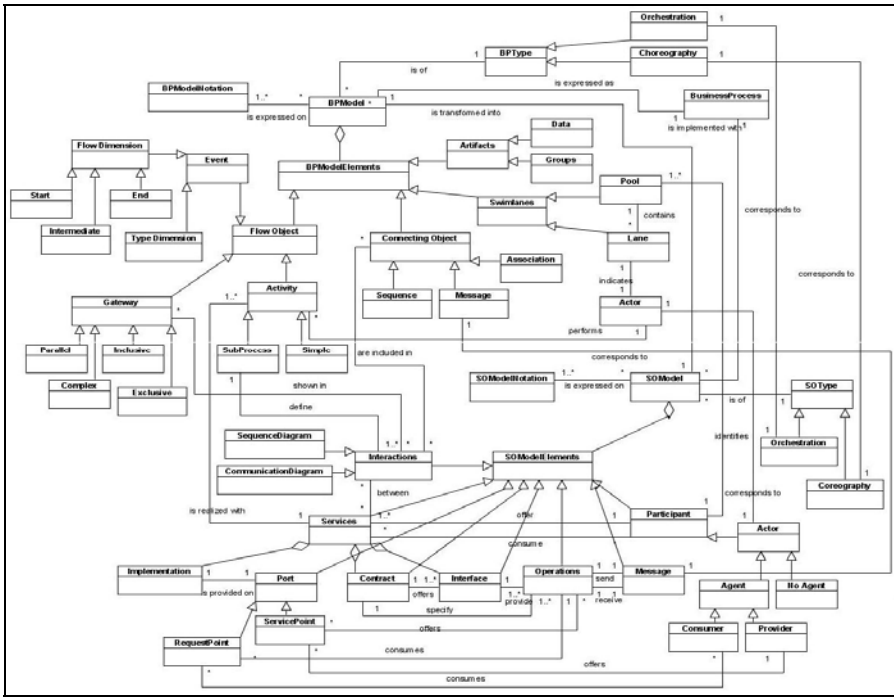


Fig. 2. Business Processes and Service Oriented Modeling sub-ontologies

### 3.2 Methodological Dimension

In this dimension the methodological approaches are integrated into MINERVA. For the definition of the methodology for the continuous business process improvement we are currently evaluating the continuous improvement process PMCompetisoft integrated into COMPETISOFT [29], to adapt it. For the Service Oriented development a previous defined methodology is integrated which defined a core set of disciplines, activities, deliverables and roles to be added to the development process used in the organization, which we briefly present in the following but can be seen in detail in [14]. In Fig. 3 the Business Modeling, Design and Implementation Disciplines, its activities and execution flow are shown as a BPMN process model. The methodology defines several related activities to develop service oriented systems from business processes, starting with business process modeling where the use of process patterns [30] is recommended. Conceptually the methodology can be added as a plug-in to the development software process used in the organization, adding the new elements to guide the service oriented development from business processes. To make this integration effective, we are working on its implementation with the Software Process Engineering Metamodel (SPEM) [31] using the Eclipse Process Framework Composer (EPF Composer) [32]. It also defines input and output deliverables (i.e. Services document) and responsible and participant roles for the defined activities, as well as a detailed description of objectives and tasks to be carried out when performing the activity, which are summarized in table 1.

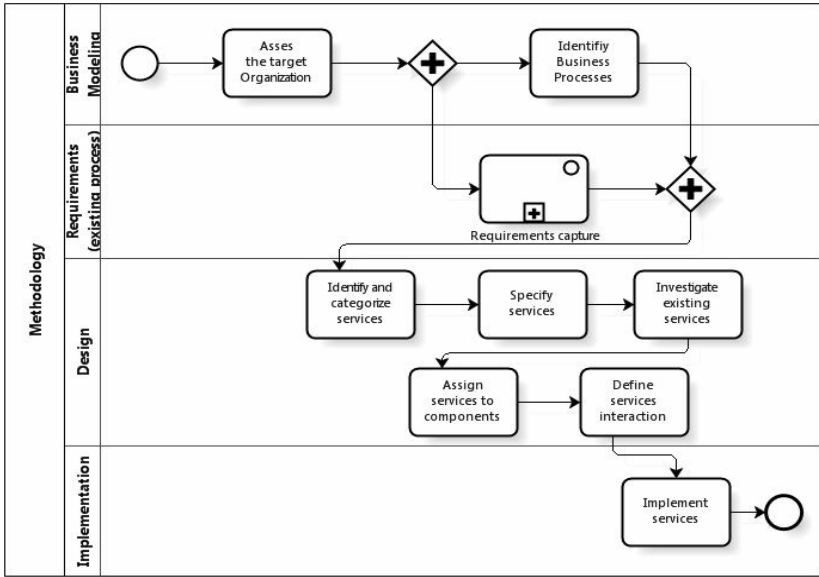


Fig. 3. Service Oriented development Methodology as BPMN process

Table 1. Elements defined in the methodology related to the presented activities

Activity	Objective	Inputs/Outputs	Roles
Assess the target Organization (MN1)	Obtain the organization map, its processes and technologies	I:Meetings with clients/ O:Evaluation of the Target Organization Document	Analyst, Architect
Identify Business Processes (MN2)	Model business processes, flow, involved roles, associated functionalities	I: Evaluation of the Target Organization Document, minutes of meetings with clients, O:BP Document	Architect, Analyst
Identify and categorize services (D1)	Define and classify the services to carry out the business processes and functionalities	I: SW Architecture, BP, Requirements and Services Documents, O: Services Document,SWArchitecture	Architect, Analyst, Developer
Specify services (D2)	Define contract of services, operations, parameters, etc.	I/O:Services Document	Architect, Analyst
Investigate existing services (D3)	Reuse services, components, implemented functionalities	I/O: Services Document and Catalogue	Architect, Analyst
Assign services to components (D4)	Define service implementation	I: Services and Design Documents, O: Services and Implementation Docs.	Architect, Developer
Define services interaction (D5)	Define sequences of invocation of services to carry out the business processes	I: BP, Services and Requirements Documents, O:Services and Implementation Docs.	Architect, Analyst, Developer
Implement services (I1)	Build the services as they were designed	I: Services, Design and Implementation Docs., O: Implemented service	Developer

### 3.3 Tool Dimension

The tool dimension gives support to the defined methodologies and automation of transformations between the different models used in MINERVA. The integrated environment, which is currently under definition, proposes the use of different tools for each phase of the business process lifecycle to support the defined concepts in all framework dimensions. For the business process modeling we are currently evaluating Bizagi [33] and Visual Paradigm [34] tools, among others, which provide BPMN. One of the main outputs of this phase is the business process model in a suitable format as XMI [35], so it could be later imported by other tools, like tools for simulation and validation of models or software development. For the software development support for business processes in the Configuration phase, the use of IDE Eclipse [36] is proposed, with a plug-in to load the business process model made, i.e. the Visual Paradigm plug-in [34] or the SOA Tools Platform Project (STP) [37].

To transform the business process models into service oriented UML artifacts, applying model driven development principles, these have to be marked with some information, i.e. the type of activity like “service task” with input and output specified messages. So, to obtain, for example, a service class model from a business process model, QVT transformations are defined from BPMN to the SoaML using the Eclipse plug-in MediniQVT [38], supported by the defined ontologies. After applying the QVT transformations, a PIM will be obtained in the SoaML profile, from which to generate the PSM to the desired technology platform. At the moment of generating the specification of the business processes in BPEL/XPDL, some marks have to be used too, to allow the generated components to be invocated from the process execution flow. Then, from the execution of the process, log files will be obtained containing the defined information to collect, to be analyzed in the Evaluation phase using the ProM [39] tool by means of the different plug-ins it provides.

## 4 Related Work

Some of the most relevant works in the area comprising the integration of MDD and SOC paradigms to business processes are: In [40] models, metamodels and transformations between them are defined to obtain a service composition model which expresses the interaction of services to perform business processes. In [41] they integrate a business value model from which to derive software artifacts. [42] proposes a business rules driven approach for the development of adaptive collaborative service oriented business processes, defining three dimensional views: collaboration aspects, business and technical requirements. In [43] a service oriented design approach is proposed, to relate services modeled in different levels of abstraction: business and application, with techniques based on ISDL: profiles to relate models and model conformance. In [44] models and metamodels for services are defined to relate them to business processes and the underlying architecture, focusing the derivation of services on three architectures: brokerless, centralized and decentralized broker. [45] defines patterns to guide the definition, transformation and implementation of technical processes using software services from business processes. [46] proposes a model driven approach to relate business process with software services in a target (distribute) three layered architecture. The business

process are modeled in UML deriving an analysis model which is mapped to the design model defined in the existing target architecture, and then to the implementation model. [47] also proposes a model driven approach for collaborative service oriented architecture, to transform BPMN into UML and BPEL models. It defines a collaborative SOA metamodel composed of three views: service, information and process; the business process model is transformed to each one.

Other proposals that integrates only one of the MDD or SOC paradigms to business processes are: [48] defines a set of transformation patterns to transform a business process into a technical process based on existing services provided by internal systems, defining levels for the quality of the transformation. Also in [49] a pattern based technique is used in a layered architecture defined in a framework for EI architectures, for service identification and transformation from business models to service architecture, organizing patterns (process, domain, SOA) in catalogues. In [50] a four level architecture and a design process to develop B2B applications are defined, selecting the services to realize business process from an existing repository or catalog. In [51] a three level conceptual framework is defined to relate business process with implemented services, with five layers including a service mediating one where a Service Invocation Coordinator (SIC) implements service invocation. In [52] UML artifacts like use cases, activity and collaboration diagrams are automatically obtained from BPMN business processes. [53] defines phases, activities and artifacts for services development from business processes, and implementation as WS. [54] proposes modeling of business process realization by services diagrams, integrating a Business Services Model (BSM) mediator between requirements and implementation.

## 5 Conclusions and Future Work

We have presented the ideas and work behind the definition of the MINERVA framework for the continuous business processes improvement, based on the application of the SOC and the MDD paradigms to business processes. The implementation of business processes by software services helps in closing the gap between business and IT areas. Our proposal includes the integration of business people into the business process modeling stage in the development process. A service oriented methodology based on business processes is integrated, and all the stages are supported by a variety of tools. An ontology is being constructed that allows defining and relating concepts from business processes to service orientation. The metamodels from which we are defining the QVT transformations, use those concepts allowing us to understand the elements we want to obtain for services models from business process models. The automatic generation of services from business processes will serve as a basis for the improvement of the implementation and execution of service based business processes. We are also working in the definition of the continuous improvement process based on business process lifecycle, which will guide the activities to perform for assessing the modeling and execution of business process to find improvement opportunities. For helping in doing the assessment, modeling and execution measures will also be integrated, basing its elements, relations and use in the defined ontology.

**Acknowledgments.** This work has been partially funded by the Agencia Nacional de Investigación e Innovación (ANII) from Uruguay, ALTAMIRA project (Junta de Comunidades de Castilla-La Mancha, Fondo Social Europeo, PII2I09-0106-2463) and INGENIO project (Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, PAC08-0154-9262).

## References

1. Business Process Management Initiative, <http://www.bpmi.org/>
2. Smith, H., Fingar, P.: Business Process Management: The third wave. Meghan-Kieffer (2003) 978-0929652337
3. Weske, M.: BPM Concepts, Languages, Architectures. Springer, Heidelberg (2007) 978-3-540-73521-2
4. Mendling, J.: Metrics for process models. Springer, Heidelberg (2008) 978-3-540-89223-6
5. van der Aalst, W.M.P., ter Hofstede, A., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
6. Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenge. IEEE Computer Society, Los Alamitos (2007)
7. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA, Service Oriented Architecture: Best Practices, 1st edn. Prentice Hall, Englewood Cliffs (2005) 978-0131465756
8. Erl, T.: SOA: Concepts, Technology, and Design. Prentice-Hall, Englewood Cliffs (2005) 978-0131858589
9. Web Services Architecture (WSA), W3C, <http://www.w3.org/TR/ws-arch/>
10. Mellor, S., Clark, A., Futagami, T.: Model Driven Development - Guest editors introduction. IEEE Computer Society, Los Alamitos (September/October 2003)
11. Stahl, T., Volter, M., et al.: Model-Driven Software Development, Technology, Engineering, Management. John Wiley & Sons, Ltd., Chichester (2006) 978-0470025703
12. Model Driven Architecture (MDA) v. 1.0.1, OMG (2003), <http://www.omg.org/mda>
13. Object Management Group (OGM), <http://www.omg.org>
14. Delgado, A., Ruiz, F.: Towards a Service-Oriented and Model-Driven framework with business processes as first-class citizens. In: 2nd Int. Conf. on Business Process and Services Computing (BPSC 2009), Leipzig (2009)
15. Business Process Maturity Model (BPMM), OMG, <http://www.omg.org/spec/BPMM>
16. Sánchez, L., Delgado, A., Ruiz, F., García, F., Piattini, M.: Measurement and Maturity of Business Processes. In: Cardoso, J., van der Aalst, W. (eds.) Handbook of Research on Business Process Modeling, pp. 532–556. Information Science Ref., IGI Global (2009)
17. Business Process Modeling Notation (BPMN), OMG, <http://www.omg.org/spec/BPMN/>
18. Soa Modeling Language (SoaML), OMG, <http://www.omg.org/spec/SoaML/>
19. Query/Views/Transformations(QVT)v.1.0,OMG (2008), <http://www.omg.org/spec/QVT/1.0>
20. Web Services Business Process Execution Language (WS-BPEL), OASIS, <http://docs.oasis-open.org/wsbpel/2.0/>
21. XML Process Definition Language (XPDL), v. 2.1, WfMC, <http://www.wfmc.org/xpdl.html>

22. van der Aalst, W.M.P., Reijers, H.A., Medeiros, A.: Business Process Mining: an Industrial Application. *Information Systems* 32(5), 713–732 (2007)
23. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2) (1993)
24. Business Process Definition Metamodel (BPDM), OMG, <http://www.omg.org/spec/BPDM>
25. García, F., et al.: Towards a Consistent Terminology for Software Measurement. *Information and Software Technology* 48, 631–644 (2005)
26. Delgado, A., Ruiz, F., García - Rodríguez de Guzmán, I., Piattini, M.: Towards an ontology for service oriented modeling supporting business processes. In: 4th International Conference on Research Challenges in Information Science (RCIS 2010) (2010)
27. Soa Reference Model, Oasis, <http://www.oasis-open.org/committees/soa-rm/>
28. Soa Ontology, Open Group, <http://www.opengroup.org/projects/soa-ontology/>
29. COMPETISOFT - Process Improvement for Iberoamerican SME, CYTED, <http://alarcos.infcr.uclm.es/Competisoft>
30. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. *Distributed and Parallel Databases* 14, 5–51 (2003)
31. Software Process Engineering Metamodel (SPEM), OMG, <http://www.omg.org/spec/SPEM/>
32. Eclipse Process Framework Composer (EPF Composer), <http://www.eclipse.org/epf/>
33. BizAgi Process Modeler, <http://www.bizagi.com/>
34. Visual Paradigm, <http://www.visual-paradigm.com/>
35. XML Metadata Interchange (XMI), OMG, <http://www.omg.org/spec/XMI/>
36. Eclipse, The Eclipse Foundation, <http://www.eclipse.org/>
37. Eclipse SOA Tools Platform Project (STP), <http://www.eclipse.org/stp/>
38. Medini QVT, ikv++ technologies ag, <http://projects.ikv.de/qvt>
39. ProM, Process Mining Group, Eindhoven University of Technology, Eindhoven, The Netherlands, <http://prom.win.tue.nl/research/wiki>
40. de Castro, V., Marcos, E., López Sanz, M.: A model driven method for service composition modelling: a case study. *Int. J. Web Engineering and Technology* 2(4) (2006)
41. de Castro, V., Vara Mesa, J.M., Herrmann, E., Marcos, E.: A Model Driven Approach for the Alignment of Business and Information Systems Models. In: 9th Mexican International Conference on Computer Science (ENC 2008) (2008)
42. Orriens, B., Yang, J., Papazoglou, M.: A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration. In: *Int. Conf. on Services Computing (SCC 2006)* (2006)
43. Quartel, D., Dijkman, R., van Sinderen, M.: An approach to relate business and application services using ISDL. In: 9th Int. Enterprise Computing Conference, EDOC 2005 (2005)
44. Roser, S., Bauer, B., Muller, J.: Model- and Architecture-Driven Development in the Context of Cross-Enterprise Business Process Engineering. In: *Int. Conference on Services Computing (SCC 2006)* (2006)
45. Zdun, U., Hentrich, C., Dustdar, S.: Modeling Process-Driven and Service-Oriented Architectures Using Patterns and Pattern Primitives. *ACM Transactions on the Web* 1(3), Article 14 (2007)



46. Herold, S., Rausch, A., Bosl, A., Ebell, J., Linsmeier, C., Peters, D.: A Seamless Modeling Approach for Service-Oriented Information Systems. In: 5th International Conference on Information Technology: New Generations (ITNG 2008) (2008)
47. Touzi, J., Benaben, F., Pingaud, H., Lorré, J.P.: model-driven approach for collaborative service-oriented architecture design. *Int. Journal of Prod. Economics* 121(1) (2009)
48. Henkel, M., Zdravkovic, J.: Supporting Development and Evolution of Service-based Processes. In: International Conference on e-Business Engineering, ICEBE 2005 (2005)
49. Gacitua-Decar, V., Pahl, C.: Pattern-based business-driven analysis and design of service architectures. In: 3rd Int. Conf. on Software and Data Technologies, ICSoft 2008 (2008)
50. Baghdadi, Y.: ABBA: an architecture for deploying business-to-business electronic commerce applications. *Electronic Commerce Research and Applications* 3(2) (2004)
51. Hu, J., Grefen, P.: Conceptual framework and architecture for service mediating workflow management. *Information and Software Technology* 45(13) (2003)
52. Liew, P., Kontogiannis, K., Tong, T.: A Framework for Business Model Driven Development. In: 12th Int. Workshop on SW Tech. and Engineering Practice, STEP 2004 (2004)
53. Papazoglou, M., van den Heuvel, W.: Service-oriented design and development methodology. *Int. J. Web Engineering and Technology* 2(4), 412–462 (2006)
54. Rychly, M., Weiss, P.: Modeling of Service Oriented Architecture: from business process to service realization. In: 3rd Int. Conf. on Evaluation of Novel Approaches to Software Engineering, ENASE 2008 (2008)

# Design for Adaptation of Service-Based Applications: Main Issues and Requirements\*

Antonio Bucchiarone<sup>2</sup>, Cinzia Cappiello<sup>1</sup>, Elisabetta Di Nitto<sup>1</sup>, Raman Kazhamiakin<sup>2</sup>,  
Valentina Mazza<sup>1</sup>, and Marco Pistore<sup>2</sup>

<sup>1</sup> Politecnico di Milano

Piazza Leonardo Da Vinci 32 20133 Milano, Italy

<sup>2</sup> Fondazione Bruno Kessler

Via Santa Croce 77 38100 Trento, Italy

{bucchiarone, raman, pistore}@fbk.eu,  
{cappiell, dinitto, vmazza}@elet.polimi.it

**Abstract.** Service-based applications are considered a promising technology since they are able to offer complex and flexible functionalities in widely distributed environments by composing different types of services. These applications have to be adaptable to unforeseen changes in the functionality offered by component services and to their unavailability or decreasing performances. Furthermore, when applications are made available to a high number of potential users, they should also be able to dynamically adapt to the current context of use as well as to specific requirements and needs of the specific users. In order to address these issues, mechanisms that enable adaptation should be introduced in the life-cycle of applications, both in the design and in the runtime phases. In this paper we propose an extension of a basic iterative service-based applications life-cycle with elements able to deal with the adaptation-specific needs. We focus, in particular, on the design phase and suggest a number of design principles and guidelines that are suitable to enable adaptation. We discuss about the effectiveness of the proposed methodology by means of real-world scenarios over various types of service-based applications.

## 1 Introduction

In the era of the Internet of Services, service-based applications (SBAs) are considered the most promising technology since they are able to offer complex and flexible functionalities in widely distributed environments by composing different types of services. Such services are often not under the control of systems developers, but they are simply exploited to obtain a specific functionality.

While this, on the one side, enables separation of concerns and highly simplifies the design effort of those in charge of building SBAs, on the other side, it introduces critical dependencies between SBA themselves and the services they are exploiting. These last ones, in fact, could change without notice or be unavailable for unprecised time intervals. Therefore, SBAs have to be able to *adapt* to these unforeseen changes. Adaptation

---

\* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

can be accomplished through various strategies that we will discuss in this paper. While the literature presents a good number of approaches that deal with self-adaptation of SBAs, most of them address this issue by hard coding in the infrastructure supporting the execution of SBAs a limited number of adaptation strategies that are triggered only when some specific and known events happen. We argue that this approach does not necessarily cover all needs that may arise. In some cases these needs are unknown and cannot be foreseen once for all.

Current methodologies for service-oriented applications are based on the results carried out in the fields of classical software and system engineering. Moreover, while almost all of the proposed approaches for life-cycle, (noticeable are the proposals by SOUP (Service Oriented Unified Process) [9] or ASTRO [12] focusing on the possibility to monitor and intervene on SBAs in order to recovery from unwanted and unexpected behaviour), assume human interventions, Linner et al. [8] propose a life-cycle supporting self-adaptation of the service-based application even if they lack of a explicit guidelines for the design of adaptable service based applications. Various frameworks supporting adaptation have been defined in the literature, each of them addressing a specific issue. Some authors focus on triggering adaptation strategies as a consequence of a requirement violation [11], or for satisfying some application constraints [13]. Adaptation strategies could be specified by means of policies to manage the dynamism of the execution environment [4][3][2] or of the context of mobile service-based applications [10].

All aforementioned approaches show interesting features, but even those that enable the definition of various adaptation strategies lack a coherent design approach to support designers in this complex task. The methodology we propose in Section 3 can be considered as a first step in this direction. The approach that we advocate is based on the idea that adaptation strategies can be programmed at design/implementation time and be associated with triggering events whenever possible, either before the execution or during the execution itself. This approach is adopted in our earlier work [3] and in other works (e.g., [10]). However, even in these cases the emphasis is on the mechanisms offered to design strategies and to trigger them, more than on a holistic, coherent, and easy to apply *design for adaptation* approach that supports developers in the usage of the available mechanisms. The objective of this paper is to go in the direction of this design for adaptation approach. We define a life-cycle for SBAs where adaptation is a first class concern.

As we think that adaptation works properly only in the case the application is designed to be adaptable, we focus, in particular, on the identification of a number of design principles and guidelines that are suitable to enable adaptation. The effectiveness of such principles and guidelines is analyzed with reference to some real-world scenarios. The rest of the paper is structured as follows: Section 2 discusses about the various facets of adaptation and evolution we deal with. Section 2.1 presents our life-cycle and Section 3 defines the design for adaptation strategies, principles, and guidelines. Finally, Section 4 assesses them with respect to the case studies and Section 5 concludes the paper.

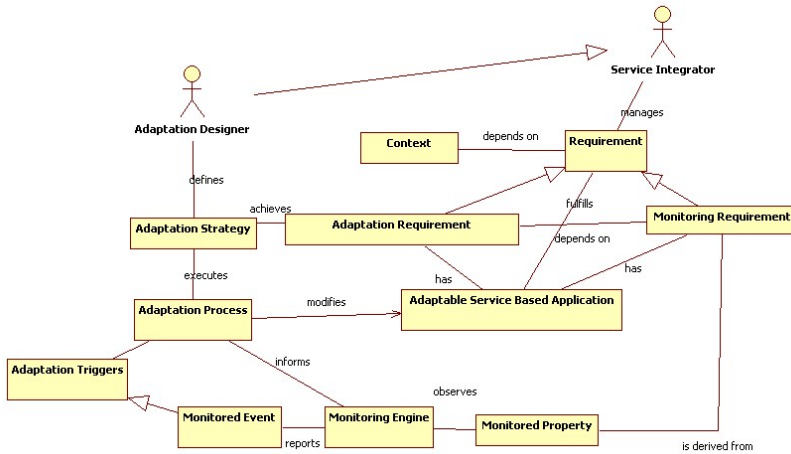


Fig. 1. Main concepts related to the definition and operation of Adaptable SBAs

## 2 Adaptation and Evolution in Service-Based Applications

Figure 1 shows the main ingredients that are needed for building and operating *Adaptable Service-Based Applications*. An adaptable SBA not only is usually able to satisfy some *Requirements*, but it also poses new requirements in terms of monitoring and adaptation aspects. *Monitoring Requirements* concern the need for detecting (part of) those situations that may trigger the need for adapting an SBA. From these requirements, designers should derive the properties to be monitored. These are then observed at runtime by a *Monitoring Engine* that, based on their values, is able to emit some *Monitored Events*. *Adaptation Requirements* are fulfilled by *Adaptation Strategies* that can be executed during the *Adaptation Process* that is triggered by *Monitored Events* or by any other external stimulus that can be acquired by the system and that leads to the modification of the Adaptable SBA. Note that analogously to the classification traditionally used to characterize software maintenance [1] we can identify similar types of adaptation: Perfective Adaptation, Corrective Adaptation, Adaptive Adaptation, Preventive Adaptation, and Extending Adaptation. An important role in our view is played by the *Context*. It includes users and execution properties. Users' characteristics and preferences can be obtained explicitly, for instance, by filling a user profile, or derived implicitly by profiling users at run-time. Other information such as the users' geographical position, the temporal details, and the actions that characterize the interaction of the users with the surrounding space can be obtained through monitoring. Execution properties are those that concern the conditions under which the SBA and its component services execute. Generally adaptation requires some temporary modification permitting to respond to changes in the requirements and/or in the application context or to faulty situations. An example of adaptation for a service composition could be the re-execution of a unavailable service or a substitution of a unsuitable service. Other situations could require the re-design and/or the re-engineering of the application modifying it permanently, in such case adaptation is called *evolution*. Moreover evolution could

be needed if a faulty situation requiring adaptation happens very often: in such case, a modification of the application logic would be preferred to the frequent enactment of the needed adaptation strategies.

## 2.1 Capturing Adaptation and Evolution Aspects in a Life-Cycle

As discussed in the previous sections, there is a need for introducing a life-cycle for SBAs that takes adaptation into explicit account. The life-cycle shown in Figure 2 highlights not only the typical design-time iteration cycle, but it also introduces a new iteration cycle at runtime that is undertaken in all the cases in which the adaptation needs are addressed on-the-fly. The two cycles coexist and support each other during the life-time of the application. In particular the design time activities allow for *evolution* of the application, that is, for the introduction of permanent and, usually, important changes, while the runtime activities allow for temporary *adaptation* of the application to the specific circumstances that are occurring at a certain time. Figure 2 also shows the various adaptation- and monitoring-specific actions (boxes) carried out throughout the life-cycle of the SBA, the main design artifacts that are exploited to perform adaptation (hexagons), and the phases where they are used (dotted lines). At the *requirements engineering and design* phase the adaptation and monitoring requirements are used to perform the design for adaptation and monitoring.

During *SBA construction*, together with the construction of the SBA, the corresponding monitors and the adaptation mechanisms are being realized. The *deployment* phase also involves the activities related to adaptation and monitoring: deployment of the adaptation and monitoring mechanisms and deployment time adaptation actions (e.g., binding). During the *operation and management* phase, the run-time monitoring is executed, using some designed properties, and help the SBA to detect relevant context and system changes. After this phase the left-side of the life-cycle is executed. Here, we can proceed in two different directions: executing evolution or adaptation of the SBA. In the first case we re-start the right-side of the cycle with the requirements engineering and design phase while in the second case we proceed identifying adaptation needs that can be triggered from monitored events, adaptation requirements or context conditions.

For each adaptation need it is possible to define a set of *suitable strategies*. Each adaptation strategy can be characterized by its complexity and its functional and non functional properties. The identification of the most suitable strategy is supported by a *reasoner* that also bases its decisions on multiple criteria extracted from the current situation and from the knowledge obtained from previous adaptations and executions. Details on these issues are discussed in Section 3. After this selection, the *enactment of the adaptation strategy* is performed.

## 3 Design for Adaptation: Main Ingredients

As discussed in the previous sections, in order to offer efficient and reliable applications, it is necessary to guarantee that the service components are always aligned with the changing world around them. At design time possible alternatives to support service adaptation should be identified. For the same SBA, several adaptation strategies can

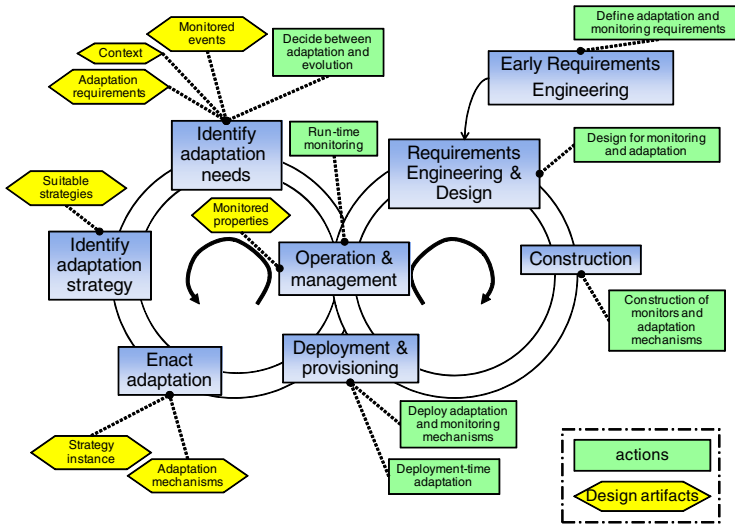


Fig. 2. The Life-Cycle of Adaptable SBAs

be adopted and the selection of the most suitable one to activate can be a complex issue multiple criteria have to be considered. In the following, guidelines to support this selection are provided.

*Adaptation Strategies:* While a SBA is executing, different changes might occur in the environment and cause inefficiencies. In order to avoid the application performance degradation, it is necessary to identify the most suitable adaptation strategy that is able to maintain aligned the application behaviour with the context and system requirements. Among the adaptation strategies, it is possible to distinguish domain-independent or domain-dependent strategies. The former are applicable in almost every application context while the adoption of the latter is limited to specific execution environments. Table 1 defines the most common domain-independent adaptation strategies.

*Identification of Adaptation Triggers:* The adaptation in SBA may be motivated by variety of factors, or *triggers*. Such triggers may concern the *component services* or the

Table 1. Description of the most common domain-dependent adaptation strategies

Adaptation Strategy	Description
<b>Service substitution</b>	Reconfiguration of the SBA with a dynamic substitution of the a service with another one
<b>Re-execution</b>	The possibility of going back in the process to a point defined as safe for redoing the same set of tasks or for performing an alternative path
<b>(Re-)negotiation</b>	Simple termination of the service used on the requester side and re-negotiation of the SLA properties to complex management on reconfiguration activities on the provider side
<b>(Re-)composition</b>	Reorganization and rearrangement of the control flow that links the different service components in the business application
<b>Compensation</b>	Definition of ad-hoc activities that can undo the effects of a process that fails to complete
<b>Trigger evolution</b>	Insertion of workflow exception able to activate the application evolution
<b>Log/update adaptation information</b>	Storage of all the information about the adaptation activities for different goals (e.g., service reputation, QoS analysis, outcome of adaptation)
<b>Fail</b>	The system reacts to the changes by storing the system status and causing the failure of the service and re-executing it

context of SBAs. As for the former we can identify changes in the *service functionality*: variation of the service interface (e.g., signatures, data types, semantics), variation of service interaction protocol (e.g., ordering of messages), and failures; and changes in the *service quality*: service availability, degrade of QoS parameters, violation of SLA, decrease of service reputation (e.g., black lists), etc. As for the contextual triggers, one can distinguish: changes in the *business context*, such as changes in agile service networks, new business regulations and policies; changes in the *computational context*, such as different devices, protocols, networks; and changes in the *user context*, such as different user groups and profiles, social environment or physical settings (e.g., location/time), different user activities. Some of these aspects may be interleaved. For example, if a user moves to a new location (i.e., change in the user context), new set of services may be available (i.e., change in the business context) with different bandwidth (i.e., change in the computational context). As represented in Table 2 each trigger can be associated with a set of adaptation strategies that are suitable to re-align the application within the system and/or context requirements. In order to select the adaptation strategy to apply, it is necessary to consider that adaptation triggers may be associated with other requirements that are important for designing and performing adaptation, in particular: the *Scope* of the change, i.e., whether the change affects only a single running instance of the SBA or influences the whole model and, the *Impact* of the change, i.e., the possibility of the application to accomplish its current task. Depending on these parameters different strategies may apply. For example, when the scope of the change concerns the whole application model “trigger evolution” strategy applies. As for the impact, such strategies as “re-execution” or “substitution” may apply when the SBA state did not change and the task still can be accomplished. On the other hand, “compensation”, “fail”, or “trigger evolution” apply when there is no way to complete the current task.

### 3.1 Design Guidelines

In order to design adaptable SBAs, it is necessary to come up with the principles and guidelines for:

- *Modeling adaptation triggers*, i.e., both the situation when the adaptation is needed (monitored property) and the specific adaptation need.
- *Realizing adaptation strategies*. This includes modeling strategies, their properties, and their aggregation, and relating them to the underlying mechanisms and run-time infrastructure.

**Table 2.** Relationships between Adaptation Triggers and Adaptation Strategies

Adaptation Trigger	Adaptation Strategy
Changes in the service functionality	Service Substitution, Re-execution, Re-negotiation, Re-composition, Compensation, Fail
Changes in the service quality	Service Substitution, Re-Negotiation
Changes in the business context	Service Substitution, Re-Negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information
Changes in the computational context	Service Substitution, Re-negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information
Changes in the user context	Service Substitution, Re-negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information

- *Associating adaptation strategies to triggers.* We have already demonstrated how the scope and impact of change influence this relation. Other factors may include autonomy (i.e., if the adaptation should be done without human involvement) or performance (e.g., how fast an adaptation strategy is).

One of the key aspects cross-cutting to these design tasks is the dynamicity of the environment with respect to the adaptation problem. This refers to the *diversity* of specific adaptation needs and of factors the adaptation strategies depend on. According to this distinction, the following design approaches may be defined:

- *Built-in adaptation.* If possible adaptation needs and possible adaptation configurations are fixed and known a priori, it is possible to completely specify them at design time. The specification may be performed by extending the standard SBA notations (e.g., BPEL) with the adaptation-specific tools [5] using ECA-like (event-condition-action) rules [3], or aspect-oriented approaches [7]. Typical strategies suitable for such adaptations are: service substitution (by using a predefined list of alternative service), re-execution, compensation, re-composition (by using predefined variants), fail.
- *Abstraction-based adaptation.* When the adaptation needs are fixed, but the possible configurations in which adaptation is triggered, are not known a priori, the concrete adaptation actions cannot be completely defined at design time. In such a case, a typical pattern is to define an abstract model of an SBA and a generic adaptation strategy, which are then made concrete at deployment/run-time. For example, it is possible to use the abstract composition model in which concrete services are discovered and bound at run-time based on the context [14]. Otherwise, it is also possible to define at design time only the final goal or utility function and then it is achieved or optimized by dynamic service re-composition at run-time on the basis of based on the specific environment and available services [15]. Strategies that may be used for such adaptation are service concretization, service substitution (by dynamic discovery), re-composition (based on predefined goal/utility function), re-negotiation.
- *Dynamic adaptation.* It is possible that adaptation needs that may occur at run-time are not known or cannot be enumerated at design time. In such a case, it is necessary to provide specific mechanisms that select and instantiate adaptation strategies depending on a specific trigger and situation. The scenarios in which such adaptation is needed may include modifications or corrections of business process instances via ad-hoc actions and changes performed by business analyst, changes in the user activities that entail modification of current composition and creation of new ones. At run-time, these mechanisms are exploited to (i) identify one or more suitable adaptation strategies depending on a concrete situation, (ii) define concrete actions and parameters of those strategies, and (iii) execute them using the appropriate mechanisms. This type of adaptation may be built on top of the others to realize specific adaptation needs; the focus, however, is on the mechanisms for extracting specific adaptation strategies and actions at run-time. Accordingly, different strategies may apply here: re-composition, service substitution, and compensation, re-execution, evolution, fail. The realization mechanisms, however, are



different; they may require active user involvement (e.g., for making decisions, for performing ad-hoc changes, etc.).

## 4 Discussion

In this section we illustrate how the design for adaptation activities may be performed in different scenarios that target different domains and focus on different adaptation aspects. In particular, given the specific characteristics of the scenario, we show the factors triggering adaptation, the types of adaptation realization suitable for the scenarios, and the appropriate adaptation strategies. Table 3 summarizes all these aspects with reference to the considered case studies.

*Automotive scenario.* Let us consider complex supply-chain business processes in the automobile production domain. The activities of the processes include ordering and importing automobile body parts from suppliers, manufacturing activities, customization of the specific products according to the needs of the customers, etc. The processes are usually long-running and involve a wide range of enterprise services provided by organizations such as various suppliers, logistics providers, warehouses, and regional representatives. All these participate in an Agile Service Network (ASN) where they rely one on each other services in a dynamic way. The critical changes that require adaptation in this scenario range from instance-specific problems (e.g., failures and SLA violations, specific customers) to the changes that affect the whole SBA (e.g., changes in business context). In the former case, it is possible to apply built-in adaptation and define the reactions at design-time by completely describing the corresponding strategy (compensation activities, process variants for different customers) or its parameters (for SLA re-negotiation, for service substitution). In the latter case, the specific adaptation strategy is chosen at run-time as the effect of changes on the system is not known. In the business settings, such a choice can hardly be automated; the business requirements and decisions require human involvement. In particular, business analysts make decisions on triggering evolution and/or on how the running process instances should be changed (i.e., ad-hoc process modifications).

*Wine production scenario.* In the wine production application domain, the activities of vineyard cultivation handling, the control of grapes maturation, their harvesting and

**Table 3.** Adaptation characteristics of the scenarios in Section 4

Case Study	Properties	Adaptation Trigger	Design Approach	Adaptation Strategy
Automotive	Stable context and potential partners, long-running SBAs, diversity of adaptation needs, decisions require human involvement	functional changes, failures, SLA violations, changes in business context	Dynamic adaptation (human-driven); built-in adaptation (for compensation or process customization)	Service substitution (selecting from ASN partners); SLA re-negotiation; re-composition by ad-hoc changes of process control/data; re-composition by selecting predefined process variants; compensation; trigger evolution
Wine	Fully dynamic and unreliable services, fully autonomous SBA	degrade of service (sensor) QoS	Abstraction-based adaptation	Re-composition of services (to optimize resource utility function), domain-specific actions (e.g., data transfer frequency changes)
Mobile user	Strong dependency from context and goals of users	context changes, changes of user activities	Abstraction-based (for context changes), dynamic	service substitution (by dynamic discovery); re-composition.

fermentation rely on extensive use of a SBA realized on top of a Wireless Sensor and Actuator Network (WSAN). In this context sensors and actuators are seen as service providers able, respectively, to report information regarding the state of the vineyard and to execute some specific actions. These devices are not fully reliable. They may crash, run out of battery, or provide incorrect information. This may happen due to changes in physical context (e.g., humidity) or to the activation of new measurement activities (e.g., depending on the season). In this scenario the dynamically changing state of the WSAN network requires continuous monitoring and optimization of the resource usage. For this purpose, the adaptation should *(i)* re-arrange the sensor network in order to minimize the sensor energy consumption, and *(ii)* optimize the modes, in which the sensors operate, e.g., by optimizing the data transfer frequency. While the latter solution requires domain-specific realization mechanisms, the former may be achieved by dynamic re-composition of services to minimize of the utility function corresponding to the energy consumption (see, e.g., [15]).

*Mobile user application scenario.* An increasing number of modern applications aims to give end users access to various services through mobile devices. Such services include route planning, transport ticket booking, services for accessing social networks, and a wide range of information services mashed up by those applications. In this scenario the SBA should adapt *(i)* to the changes in its context (e.g., changing location and time, different user settings), and *(ii)* to the changes in the user activities and plans. The former may be very dynamic: different services may apply for different locations or user settings. Abstraction-based adaptation is indeed required in this case: the abstract activities (e.g., buy a ticket for local transportation) are defined at design-time and made concrete at run-time using service concretization techniques (e.g., buy a ticket using online service of public transport company). To deal with the changes in user activities and plans, it is necessary to understand the impact of those changes on the current processes and state of the SBA (i.e., perform dynamic adaptation). Depending on the outcome, different adaptations may apply (e.g., compensate or re-compose some tasks, fail). Differently from automotive scenario where the business analysts are high-level domain experts, these decisions can not be delegated to the mobile user, as they may have no expertise on the low-level technical details of the SBA. Therefore, it is necessary to design such decision mechanisms that at run-time may reason on the specific situation in order to reveal an appropriate strategy and its parameters (e.g., to decide whether re-composition may be done, to derive concrete composition goal and the corresponding composition, etc.). In [6], in particular, such decision mechanism relies on the analysis of personal information of the user (e.g., context, agenda, tickets and reservations, etc.).

## 5 Conclusions

This paper proposes a design method for SBAs that targets the adaptation requirements of those applications and aims at overcoming the fragmentation in current approaches for SBA adaptation. The approach is based on a novel life-cycle that considers adaptation as a first class concern and that covers the different facets of adaptation, both during the design phase and at run-time. Admittedly, this paper is just a first step towards our ultimate goal of defining a holistic design method for adaptable SBAs. Still,

the effectiveness of such principles and guidelines is witnessed by their capability to capture the key aspects of adaptation in the different, heterogeneous real-world scenarios considered in this paper. Our future roadmap includes a refinement of guidelines and principles presented in this paper, their formalization into patterns, and the definition of more precise criteria to decide on the patterns that are most appropriate for a given adaptation need. We also intend to work on the development of mechanisms and tools supporting the methodology, building on top of the actions and artifacts identified in Figure 2. Finally, we intend to work on a stronger empirical evaluation of the proposed methodology, by applying it to the real-world scenarios we already exploited in this paper.

## References

1. International Standard - ISO/IEC 14764 IEEE Std 14764-2006. pp. 1–46 (2006)
2. Baresi, L., Guinea, S., Pasquale, L.: Self-healing BPEL processes with Dynamo and the JBoss rule engine. In: ESSPE 2007, pp. 11–20. ACM, New York (2007)
3. Colombo, M., Nitto, E.D., Mauri, M.: Scene: A service composition execution environment supporting dynamic changes disciplined through rules. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 191–202. Springer, Heidelberg (2006)
4. Erradi, A., Maheshwari, P., Tasic, V.: Policy-driven middleware for self-adaptation of web services compositions. In: van Steen, M., Henning, M. (eds.) Middleware 2006. LNCS, vol. 4290, pp. 62–80. Springer, Heidelberg (2006)
5. Karastoyanova, D., Houspanossian, A., Cilia, M., Leymann, F., Buchmann, A.P.: Extending BPEL for Run Time Adaptability. In: EDOC, pp. 15–26 (2005)
6. Kazhamiak, R., Bertoli, P., Paolucci, M., Pistore, M., Wagner, M.: Having Services “Your-Way!”: Towards User-Centric Composition of Mobile Services. In: FIS (2008)
7. Kongdenfha, W., Saint-Paul, R., Benatallah, B., Casati, F.: An Aspect-Oriented Framework for Service Adaptation. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 15–26. Springer, Heidelberg (2006)
8. Linner, D., Pfeffer, H., Radosch, I., Steglich, S.: Biology as Inspiration Towards a Novel Service Life-Cycle. In: Xiao, B., Yang, L.T., Ma, J., Muller-Schloer, C., Hua, Y. (eds.) ATC 2007. LNCS, vol. 4610, pp. 94–102. Springer, Heidelberg (2007)
9. Mittal, K.: Service oriented unified process, <http://www.kunalmittal.com/html/soup.html>
10. Rukzio, E., Siorpaes, S., Falke, O., Hussmann, H.: Policy based adaptive services for mobile commerce (2005)
11. Spanoudakis, G., Zisman, A., Kozlenkov, A.: A Service Discovery Framework for Service Centric Systems. In: IEEE International Conference on Services Computing, vol. 1, pp. 251–259 (2005)
12. Trainotti, M., Pistore, M., Calabrese, G., Zacco, G., Lucchese, G., Barbon, F., Bertoli, P., Traverso, P.: ASTRO: Supporting Composition and Execution of Web Services. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 495–501. Springer, Heidelberg (2005)
13. Verma, K., Gomadam, K., Sheth, A.P., Miller, J.A., Wu, Z.: The METEOR-S Approach for Configuring and Executing Dynamic Web Processes. Technical report, University of Georgia, Athens (June 2005)
14. Verma, K., Gomadam, K., Sheth, A.P., Miller, J.A., Wu, Z.: The METEOR-S Approach for Configuring and Executing Dynamic Web Processes. In: Technical report (2005)
15. Zeng, L., Benatallah, B., Dumas, M., Kalaganam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW, pp. 411–421. ACM, New York (2003)

# Towards Runtime Migration of WS-BPEL Processes<sup>\*</sup>

Sonja Zaplata, Kristian Kottke, Matthias Meiners, and Winfried Lamersdorf

Distributed Systems and Information Systems  
Computer Science Department, University of Hamburg  
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany  
{zaplata,3kottke,4meiners,lamersd}@informatik.uni-hamburg.de

**Abstract.** The decentralized execution of business process instances is a promising approach for enabling flexible reactions to contextual changes at runtime. Most current approaches address such process distribution by *physical fragmentation* of processes and by dynamic assignment of resulting static process parts to different business partners.

In order to enable a more dynamic segmentation of such responsibilities at runtime, this paper proposes to use *process runtime migration* as a means of *logical process fragmentation*. Accordingly, the paper presents a general migration metadata model and a corresponding basic privacy and security mechanism for enhancing existing process models with the ability for runtime migration while respecting the intentions and privacy requirements of both process modelers and initiators. The approach is conceptually evaluated by applying it to WS-BPEL processes and comparing the results to the general concept of process fragmentation.

## 1 Motivation

In today's networked business environments, cross-organizational collaborations composing complementary services and thus realizing new, value-added products gain increasing importance. As a technical representation of such business processes, executable workflows allow for flexible, dynamic and loosely-coupled collaboration among several business partners. The *Business Process Execution Language for Web Services (WS-BPEL)* [1] is currently one of the most relevant practical approaches. It allows for distributing resources such as employees, machines and services, whereas process control flow logic is typically executed by one single component at one single site [2].

However, due to the autonomy of participants, a single centralized process management system to control the execution of cross-organizational processes is often neither technically nor organizationally desired. As an example, required services and resources often cannot be accessed by a centralized process engine

---

<sup>\*</sup> The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

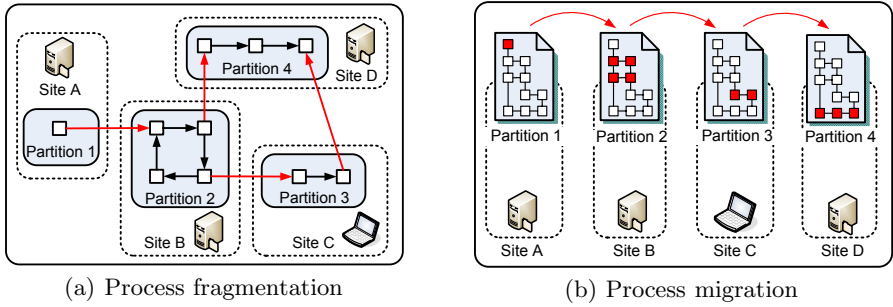


Fig. 1. Process decentralization variants

because of technological differences or due to security policies [3]. Furthermore, in some cases the location where a process fragment is executed is relevant to perform the required functionality or is necessary for judicial reasons, e.g. in the context of eGovernment. Related to this, other non-functional aspects such as execution time, performance, navigation cost and capacity utilization can be optimized by load balancing and thus improve flexibility and scalability of participating systems [4].

Most current research is approaching decentralization of control flow navigation by a *physical fragmentation* of processes – splitting the overall executable process into several subparts which are then distributed to a number of available process engines (cp. figure 1(a)). In contrast to that, this paper proposes *process instance migration* as a means of *logical fragmentation*, fragmenting only the responsibilities for the execution of the process into a set of sub-responsibilities while preserving the original structure of the process description for all of the participating systems (cp. figure 1(b)). Such migration is the most “natural” way of executing a distributed process – as inherited by traditional human-based workflow management: A process is described in subsequent steps which are passed from one workplace to another. Logical fragmentation by process instance migration has several advantages over physical process fragmentation:

- Process migration allows for fragmenting the responsibility to execute a process at runtime – depending on the availability of business partners or other contextual incidences. Furthermore, the granularity of fragmentation and the range of distribution can be selected on the fly by each executing participant.
- Coordination and merging of multiple process fragments are not necessary in the case of sequential execution. Global variables, scopes, errors and transactions are easier to handle, because all these aspects of the process (i.e. data and control flow) are available to all executing parties. Thus, there is less communication and coordination overhead.
- Process instance migration is applicable to modern distributed systems including mobile devices because it does not depend on a single centralized system and allows for dynamic sharing of restricted resources [3,5].

However, process migration has also some drawbacks and still includes some interesting challenges which this paper proposes to address. First, the process description needs to implement a formal or technical model to communicate the current state of the migrated process instance. To preserve interoperability, this model should not require modifying the original business process [2]. Second, an important motivation for physical process fragmentation is given by the resulting separation of process fragments. If the process is to be fragmented for privacy reasons, process migration lacks proper security mechanisms in order to protect private information carried within the process.

This paper presents an approach to enhance existing processes with non-intrusive migration metadata and an overall system architecture to support runtime process migration among cooperating process execution systems. Therefore, we analyze which information has to be attached to the process at design time in order to execute the (logical) process fragments as it was originally intended by the designer or the initiator of the process in whole. Furthermore, we present an initial privacy mechanism to protect the migrating process instance against unwanted changes and unauthorized access. Finally, the approach presented here is applied to WS-BPEL and a respective prototype implementation, before the paper concludes with a short summary.

## 2 Related Work

Distributed and decentralized process execution becomes increasingly important and, consequently, many such approaches demonstrate the relevance of this research (cp. [6] for a brief overview). A first possible solution for distributing the control flow of a process is to change the service granularity. The activities which should be outsourced are wrapped, encapsulated behind a new service interface and the remaining process model is changed accordingly (e.g. [7,8]). However, such process fragmentation is mostly carried out at design time and is realized by weaving additional activities into the resulting fragments in order to realize a standard-compliant communication between them at runtime. Another general approach is to split the original process, deploy the resulting parts at the desired system and induce *choreography* between the separated processes (e.g. *CiAN* [9]). However, choreography and process fragmentation need a joint preparation phase for the physical distribution of each (sub-)process where all participating parties have to be available. Therefore, this approach is more advantageous in case of a similar recurrent execution of the same process than for spontaneous reactions to (unfrequent) ad-hoc changes.

In order to avoid the introduction and maintenance of new services, Martin et al. [2] propose a non-intrusive approach for process fragmentation and decentralized execution. Here, fragmentation is achieved by transforming the orchestration logic represented in WS-BPEL into a set of individual activities which coordinate themselves by passing tokens over shared distributed tuple-spaces. Decentralized process execution has also been considered in Mentor [10] by partitioning a process based on activity and state charts. Addressing more

dynamic environments, the approach of *MobiWork* [11] realizes mobile workflows for ad-hoc networks and is focused on the allocation of tasks to mobile participants also using process fragmentation to generate “sub-plans”.

However, all presented approaches support at most dynamic allocation and assignment on the basis of a *static* fragmentation. All fragments and responsible parties are determined either at design time or once after invocation but mostly before executing the first activity of the process instance. Considering long-running processes, this flexibility may not be enough in order to also allow reactions to spontaneous contextual changes. In contrast, a *dynamically continuable* runtime segmentation implies that fragments and responsible parties are determined dynamically according to the current context and with respect to previous results and requirements of upcoming activities during the actual execution of the process instance.

A way to address such dynamic behavior is the runtime migration of entire process descriptions as introduced by Cichocki and Rusinkiewicz [12] in 1997. More recently, the framework *OSIRIS* [13] relies on passing control flow between distributed workflow engines in order to execute service compositions. Process data is kept in a distributed peer-to-peer-database system which can be accessed from each node participating in the process execution. In *Adept Distribution* [14] a similar approach supports dynamic assignment of process parts to so-called *execution servers*. A process instance can be migrated from one execution server to another and the next participant is dependent on previous activities which are able to change the participant to execute the next partition. Related to this, Atluri et al. [4] present a process partitioning algorithm which creates self-describing subprocesses allowing dynamic routing and assignment.

Process migration has also particularly been applied to the area of mobile process execution, e.g. by Montagut and Molva [5]. Their approach relies on passing control flow between distributed WS-BPEL engines and addresses security on an application level by integrating a public/private process model in order to access applications internal to mobile devices. However, such a solution represents a choreography-like approach which only uses process migration in order to hand-over control flow – and thus also has some of the aforementioned disadvantages. The last example is the *DEMAC* middleware [3] which is able to delegate process execution (in whole or in part) to other stationary or mobile process engines. Its restriction to a proprietary process description language is, however, an obstacle to migrate existing business processes and to integrate standard process engines of external parties.

Complementing existing migration approaches, we therefore introduce a more technology-independent migration model which can also be applied to existing processes. Furthermore, most migration approaches give scope for rather too much flexibility, i.e. the process instance is migrated without control or the decision about the next participant is determined by one of the execution systems - but often cannot be influenced by the process modeler or initiator. The following section presents a respective concept while considering the above-mentioned user-defined requirements for logical process fragmentation.

### 3 Overview of Process Instance Migration

The proposed methodology of process instance migration is depicted in figure 2. The development starts with the original modeling of the underlying business process which produces a process model, specified in an executable process description language such as WS-BPEL (step 1). Optionally in step 2, this process model can now be enhanced by a *migration metadata model* which holds all information required for migration and all user-defined requirements. In the following, process model and migration model are deployed (step 3) and can be instantiated by an application or a user (step 4). If required, parameters are passed to customize the process (i.e. normal invocation parameters) or the migration model. The latter is advantageous if the initiator should be allowed to influence non-functional aspects about the way a process is executed (e.g. if the user pays for a higher service quality, the selection of migration partners is influenced accordingly). After that, the resulting process instance is executed following the guidelines of the associated migration metadata. However, if migration metadata is omitted or migration is not supported, the unaffected process can still be deployed and executed the usual (centralized) way.

The proposed migration model and its relationship to general process elements are depicted in Figure 3. As a starting point, we assume a common minimal process model consisting of a finite number of *activities* representing the tasks to be fulfilled during process execution, and a finite number of *variables* holding the data which is used by these activities. Activities can represent a specific task (*atomic activities*) or a control flow structure as a container for other activities (*structured activities*). Furthermore, variables can be specified on process level (*global variables*) or at activity level (*local variables*). Optionally, variables can contain *initial values* which are assigned at design time.

A process description complying to these properties (e.g. XPDL or WS-BPEL) can be enhanced by migration metadata documenting the execution state of the process (*process state*) and of each activity (*activity state*), such that the progress in processing the activities is well-defined and visible for every participating

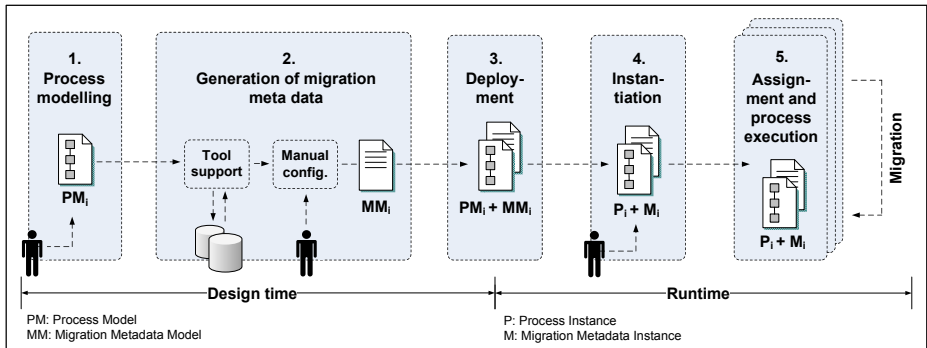


Fig. 2. Process migration: methodology



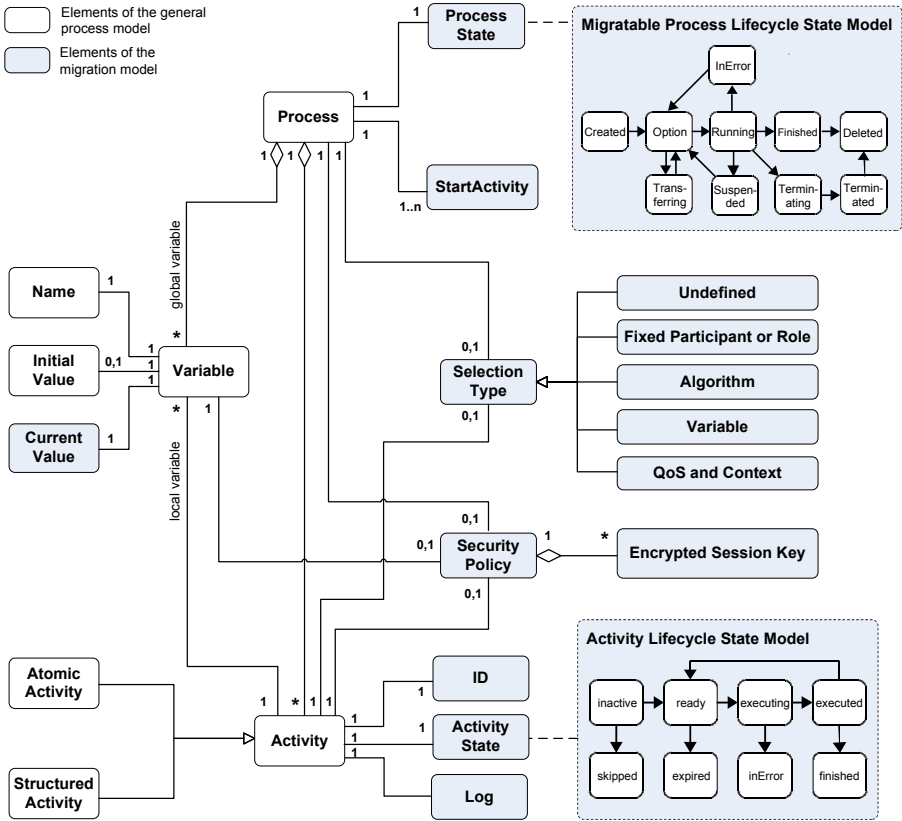


Fig. 3. Overview of the migration model

device at any time during execution. The process state can take a value from the *migratable process lifecycle model* (cp. [3]) as depicted in the upper right corner of Figure 3. In addition to that, a set of activities can be referenced as *startactivities* to mark the first activity to be executed after process migration. The model allows for multiple startactivities in case the order in which the activities have to be executed is irrelevant or the activities should be processed in parallel. Furthermore, the states of the variables have to be documented, i.e. their *current values* have to be attached to the migration data.

Up to this point, basic migration metadata can be generated automatically, i.e. by setting the process to the state *created* and all activities to *inactive* (cp. first part of step 2 in figure 2). If variables have been specified with an initial value, the given value is set as the current value of the variable. However, the process modeler or the actual initiator often wants to customize the way the distributed process is executed. The *selection type* determines which strategy is used to assign an activity to a specific process engine. If the selection type is *undefined* (default) the process engine which is currently working on the process instance decides about further migrations, e.g. to shift processes to other engines

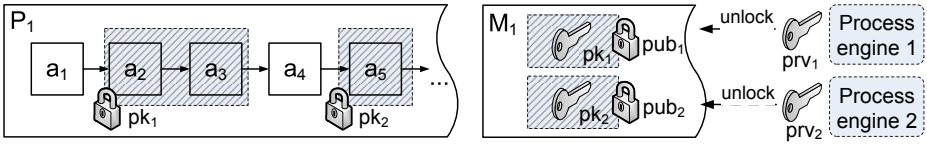


Fig. 4. Process encryption and key distribution

which operate at a better performance. In contrast, the type *fixed participant or role* determines that a specific executing entity (e.g. a human or a concrete process engine) or a subject of a defined group of such entities has to execute the process or a specified set of activities. More dynamically, as proposed by [14], the next participant can also be picked from a *variable* within the process description itself. If no such entities should be specified, but the participant should be selected as a result of a computation (e.g. picking the process engine which can execute as much of the process as possible), the respective *algorithm* is referenced. Finally, the selection can be based on specific *quality of service or context* requirements such as current workload or geographical location.

To prevent privacy and security threats, the access to process data can be restricted to specified subjects or roles, as e.g. determined in the above mentioned selection types *fixed participant or role*. Figure 4 shows the general idea of “masking” critical parts of a process description in order to ensure that only dedicated participants can execute sensitive activities and access corresponding data. The approach assumes a basic cryptographic key infrastructure, such as PKI (Public Key Infrastructure) or subject-related shared keys. However, most process description languages (such as also WS-BPEL) allow for the definition of global variables which can be referenced in several activities – and thus might belong to more than one participant. In consequence, these parts cannot be directly encrypted with the personal key of the authorized subjects. Alternatively, the encryption of the different parts of the process (i.e. activities, variables or even the whole process) uses different *session keys* which are only used once. A corresponding *security policy* of such an element therefore contains a number of symmetric keys (e.g.  $pk_1$  and  $pk_2$  in Figure 4). The procedure of key distribution is based on a concept which is derived from broadcast encryption [15] where the same encrypted content is sent to all receiving parties without the need for two-way authentication or authorization. In the approach presented here, the keys necessary for decryption are sent together with the protected content. In case of an existing PKI the entries are encrypted with the public key  $pub_i$  of the appropriate subject (cp. Figure 4) and can be unlocked with the private key  $prv_i$ . Accordingly, an entry for each authorized subject is created and added to the migration meta data of the protected process element. As the result of this step, only the legitimate receiver is able to obtain the keys and decrypt the content and even encrypted global variables can be accessed by different authorized subjects using the same session key [16]. Using MACs (Message Authentication Codes) for each security-related process part, the process initiator is optionally able to also ensure the integrity of the process description. Neither

an additionally interaction between the process initiator and the subjects nor an authentication is needed. As a positive side-effect, the use of unique session keys also increases the resistance of the cryptographic approach to attacks.

## 4 Migration of WS-BPEL Process Instances

WS-BPEL is a block-structured XML-based process description language which allows composing web services. According to the WS-BPEL 2.0 specification by OASIS [1] it is essentially comprised of two kinds of activities: Basic activities for web service interaction (*invoke*, *receive*, *reply*), basic control flow activities (*empty*, *wait*, *exit*, *throw*, *rethrow*) and activities for data manipulation (*assign*). Structured activities are used to compose the basic activities and define control-flow dependencies between them (*sequence*, *if then else*, *pick*, *flow*, *while*, *repeat until*, *for each*). Based on this characterization, the activities have been assigned to the elements of the general model in figure 3 and migration possibilities have been analyzed using a prototypical implementation of a software component responsible for performing the migrations.

Considering atomic activities, it shows that WS-BPEL has a very interactive character which makes the distribution of the control flow logic (both for migration and for fragmentation) difficult. The *invoke* activity initiates the invocation of a web service which is specified within the process description (or references associated parts such as WSDL files) in either an abstract or a specific way. Thus, migration of a process containing an *invoke* activity is not only possible, but even advantageous if the required service is not reachable from the current system. In case of a synchronous service call (*request-response* pattern) the receipt of the response message is part of the atomic activity. In case of asynchronous messaging, sending an associated *reply* subsequent to a migration is also not critical as the required information about the receiver (e.g. its physical address) is logged. Nevertheless, receiving a reply (*receive*) requires the specification of a specific participant because the sender of the reply has to know where to send the message. In this case, the flexibility of distribution is limited.

The assignment of a variable (*assign*) is not a problem as the current value is stored within the migration metadata. The same is true for *wait*, *empty* and *exit* activities as these have a rather simple behavior. Notifications about faults are also uncritical as in case of process migration all the relevant information for fault handling (i.e. *scopes*, *fault handler*, *compensation handler*) are available to the executing party. If required, the occurrence of faults can also be documented in the log, e.g. if the control flow logic has to return to the failed activity when fault handling is finished. In contrast to process fragmentation, coordination is not required in order to notify other process fragments.

Structured activities such as *sequence* or *while* do not have to be finished in order to allow the migration of the process instance. This is another advantage over physical process fragmentation where e.g. loops often have to be distributed as a whole: By storing the current value relevant for the evaluation of the loop condition in the migration data, migration is even possible within iterations.

If the condition has to be evaluated only once (such as in the case of *if then else*) the selected branch is determined by the process's startactivity. In case of process fragmentation, the execution of resulting *dead path eliminations* requires further coordination if process fragments are distributed physically. In case of process migration, the dead path can be processed automatically by setting all upcoming activities (until the next join condition) to the *skipped* state. As this information is hold in the migration document, this does generally not involve communication with other participants.

The *pick* activity waits for the occurrence of an event from a set of events and then executes the activity associated with that event. If the process is fragmented physically, this is a problematic issue. Either all the necessary data has to be replicated (i.e. all event/reaction pairs) or the events have to be fragmented as well. If the reaction to an event affects other fragments, additional coordination is necessary. In case of process migration, this is not a problem as the whole spectrum of possible events and reactions is available to the responsible participant. If, furthermore, other activities are temporarily suspended because of the event, the activity states indicate where the execution must be continued. However, each process participant has to subscribe to each required event as long as it is responsible for the execution of the process instance. Thus, during migration time, there is a remaining risk that some events may be not be noticed.

The *flow* activity contains activities which should be processed in parallel. As long as the process is migrated to exactly one participant, migration within the execution of a flow is uncritical as the states of each included activity are well-defined. Nevertheless, the process cannot be transferred until all atomic activities have reached a stable state and thus may have to wait for long-running activities to be finished. Since the execution of parallel paths on a single machine cannot be considered as "real parallelism", a copy of the (entire) process can be distributed to different participants which are each responsible for the execution of one of the parallel paths. In order to synchronize parallel paths, there has to be a defined meeting point. In consequence, distributed parallel execution needs advanced coordination mechanisms for both migration and fragmentation. However, using replication instead of fragmentation allows for a local detection of shared variables and thus avoids unnecessary synchronizations.

As a drawback for process migration, privacy can only be realized by artificially masking private process parts, whereas physical fragmentation of the process makes such mechanisms unnecessary. In consequence, the effort for developing migratable processes containing private parts is a little higher – and process modelers must be aware of the fact that applying such policies reduces the number of potential migration partners and thus again may restrict flexibility. Nevertheless, even within user-defined limitations, process migration allows for more flexibility than physical process fragmentation – and especially long-running distributed process instances benefit from the possibility to adapt the remaining unrestricted execution of control flow to changing conditions.

## 5 Conclusion

This paper focuses on distributed process execution involving multiple engines in order to increase flexibility and to improve reactions to ad-hoc context changes. As an alternative to physical process fragmentation, a concept for realizing logical process fragmentation on the basis of process migration has been presented. A prototype system covering the proposed system architecture for XPDL and WS-BPEL processes shows basic applicability of the proposed concepts. Compared to physical fragmentation, process migration provides more flexibility by allowing to distribute running process instances at runtime while respecting the guidelines of the process modeler. On the other hand, privacy and security-related issues have to be considered explicitly as also addressed in this paper.

## References

1. OASIS: Web Services Business Process Execution Language Version 2.0. Technical report, OASIS (2007)
2. Martin, D., Wutke, D., Leymann, F.: A Novel Approach to Decentralized Workflow Enactment. In: *Enterprise Distributed Object Computing*, pp. 127–136. IEEE, Los Alamitos (2008)
3. Zaplata, S., Kunze, C.P., Lamersdorf, W.: Context-based Cooperation in Mobile Business Environments: Managing the Distributed Execution of Mobile Processes. In: *Business and Information Systems Engineering (BISE)*, vol. 2009(4) (October 2009)
4. Atluri, V., et al.: A Decentralized Execution Model for Inter-organizational Workflows. *Distrib. Parallel Databases* 22(1), 55–83 (2007)
5. Montagut, F., Molva, R.: Enabling Pervasive Execution of Workflows. In: *Collaborative Computing: Networking, Applications and Worksharing*. IEEE, Los Alamitos (2005)
6. Jablonski, S., et al.: A Comprehensive Investigation of Distribution in the Context of Workflow Management. In: *ICPADS 2001*, pp. 187–192 (2001)
7. Khalaf, R., Leymann, F.: A Role-based Decomposition of Business Processes using BPEL. In: *IEEE International Conference on Web Services*, pp. 770–780. IEEE, Los Alamitos (2006)
8. Baresi, L., Maurino, A., Modafferi, S.: Towards Distributed BPEL Orchestrations. *ECEASST 3* (2006)
9. Sen, R., Roman, G.C., Gill, C.D.: CiAN: A Workflow Engine for MANETs. In: Lea, D., Zavattaro, G. (eds.) *COORDINATION 2008*. LNCS, vol. 5052, pp. 280–295. Springer, Heidelberg (2008)
10. Muth, P., et al.: From centralized workflow specification to distributed workflow execution. *J. Intell. Inf. Syst.* 10(2), 159–184 (1998)
11. Hackmann, G., Sen, R., Haitjema, M., Roman, G.C., Gill, C.: *MobiWork: Mobile Workflow for MANETs*. Technical report, Washington University (2006)
12. Cichocki, A., Rusinkiewicz, M.: Migrating Workflows. In: *Advances in Workflow Management Systems and Interoperability*, NATO, pp. 311–326 (1997)

13. Schuler, C., Weber, R., Schuldt, H., Schek, H.J.: Scalable Peer-to-Peer Process Management - The OSIRIS Approach. In: ICWS, pp. 26–34 (2004)
14. Bauer, T., Dadam, P.: Efficient Distributed Workflow Management Based on Variable Server Assignments. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 94–109. Springer, Heidelberg (2000)
15. Lotspiech, J., Nusser, S., Pestoni, F.: Broadcast Encryption's Bright Future. *Computer* 35(8), 57–63 (2002)
16. Bertino, E., Castano, S., Ferrari, E.: Securing XML documents with Author-X. *IEEE Internet Computing* 5(3), 21–31 (2001)

# Encapsulating Multi-stepped Web Forms as Web Services

Tobias Vogel, Frank Kaufer, and Felix Naumann

Hasso Plattner Institute, University of Potsdam, Germany  
`firstname.lastname@hpi.uni-potsdam.de`

**Abstract.** HTML forms are the predominant interface between users and web applications. Many of these applications display a sequence of multiple forms on separate pages, for instance to book a flight or order a DVD. We introduce a method to wrap these multi-stepped forms and offer their individual functionality as a single consolidated Web Service. This Web Service in turn maps input data to the individual forms in the correct order. Such consolidation better enables operation of the forms by applications and provides a simpler interface for human users.

To this end we analyze the HTML code and sample user interaction of each page and infer the internal model of the application. A particular challenge is to map semantically same fields across multiple forms and choose meaningful labels for them. Web Service output is parsed from the resulting HTML page. Experiments on different multi-stepped web forms show the feasibility and usefulness of our approach.

**Keywords:** deep web, web services, html forms.

## 1 Multi-stepped Web Forms

The Web has changed several tasks that have been achieved by other means, earlier. Telephone numbers are found on **yellowpages.com**, books can be bought on **amazon.com**, etc. These pages have a common feature, which is the usage of HTML forms to collect user input (names, book titles). Hence, the content of the resulting pages, e.g., the availability of a book, is “hidden” behind these forms. This phenomenon is commonly coined the *Deep Web* or *Hidden Web* [2]. Even more common, results are not shown after filling a simple form, but after having stepped through several subsequent forms. Research on how to exploit the data within the Deep Web itself is already an ongoing research topic [7,10,12,13].

Figure 1 illustrates such a flow of sequential forms (a multi-stepped web form), the New York City public transport<sup>1</sup>. On the start page (1), details such as origin and destination of the trip, but also temporal and route preferences are requested. Depending on the inserted values, one or both of the intermediate pages (2a/2b) come up, asking for additional/corrected values, until finally the resulting page (3) is presented.

<sup>1</sup> <http://lirr42.mta.info/sfweb/faces/index.jspx>

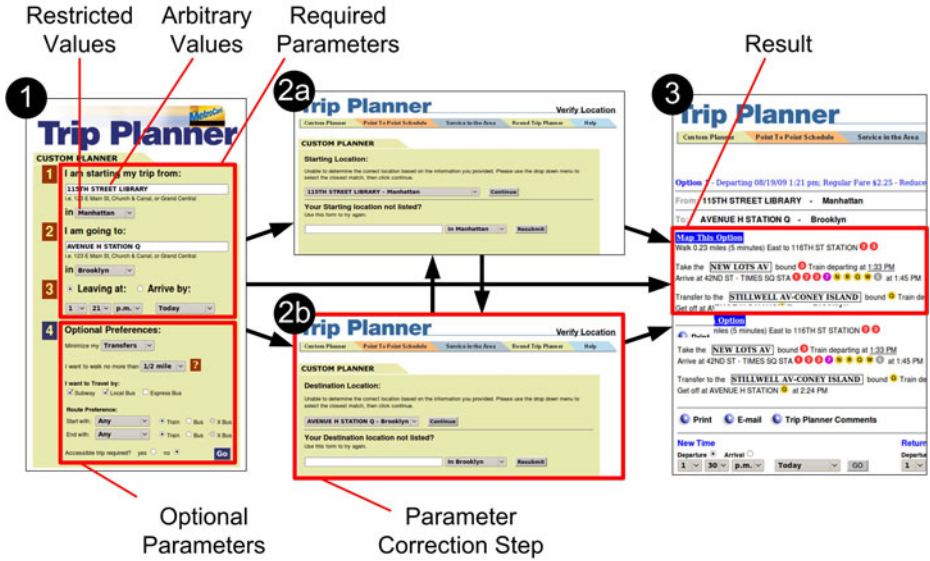


Fig. 1. Example for a common web application (public transportation) and possible user navigation paths through it

### 1.1 Challenges

Multi-stepped web forms are not easy to analyze, because correspondences between forms in different steps (c.f. Section 2) have to be captured, in contrast to single-stepped web forms. The thereby unified view on the forms is offered as a Web Service. Thus, the resulting Web Service interface has to be a subset of the elements occurring within the forms. This subset has to be defined and the corresponding parameters have to be annotated in a way that the Web Service can be used properly, afterwards. Furthermore, it is not sufficient to create the plain result pages, because these have to be handled again to return a Web Service compliant result.

We put forth two use cases, one for provisioning of human-oriented web applications to computer programs and one for leveraging the use of web applications for humans.

*First*, web applications wrapped as Web Services enable computer programs to access these applications originally intended and designed for human use, only.

*Second*, also humans benefit directly from these Deep Web queries by facilitating the user interface. Often, input data forms are spread out over several web pages to reduce the amount of input data per page for the user or to cope with dependencies. However, for re-occurring queries this means unnecessary overhead for the end user, because the user might never decide for different settings, so he does not need to be asked for it each time he uses the application.

When wrapping these applications as Web Services, they can be presented in a much more condensed form, all input fields and controls can be located in one place, if XForms are used to create user interfaces from WSDL descriptions [9].



## 1.2 Contributions

We make the following contributions:

- We present a technique to match schemas of different web forms to a joint schema, which is offered with a Web Service interface.
- We also show how to retrieve meaningful annotations for these schema elements, which are then used in the Web Service’s WSDL description.
- Further, we introduce a method for extracting the result of the aforementioned queries and transforming it into a valid Web Service response.
- The whole prototype system is provided as a web-based application, which proxies the user’s interaction with the original page. Since some web applications use session information and user login fields in their forms, these are also considered.

A web-based prototype of the system was implemented and integrated into the PoSR (Potsdam Service Registry) project [1]. PoSR is available at <http://posr.ws/><sup>2</sup>

This paper is structured as follows: Section 2 goes deeper in the common practice of web applications and explains what we call “background model”. Based on that, Section 3 explains the web application analysis process and the heuristics and algorithms used. The results are evaluated in Section 4. Section 5 gives an overview of related research in this field. The contribution is summarized in Section 6.

## 2 A Model for Multi-stepped Web Forms

### 2.1 Variables in Web Applications

Web applications make use of variables to save internal states and previous user input, much like other applications. However, since the user interaction takes place in a web browser, these variables are observable during the interaction, e.g., as name attributes of input elements or as parameters in HTTP requests.

Although the complete set of variables is used for the application to achieve the task, not all variables appear in every HTML form of the multi-stepped process. We call this complete set of application variables “background model”. It has to be reconstructed as far as possible from the different subsets of variables that occur in each step. This is possible due to the fact, that web applications usually are using a single application engine for the comprised pages and therefore, the variables can be matched.

Figure 2 illustrates such a schema of internal application variables while a formal description is given in Section 2.2. Each row stands for a distinct variable, each column represents a form, in which a variable might or might not appear. In case it appears, it might be renamed, initialized with a value, or given a

---

<sup>2</sup> The sources are available within the InstantSVC project under <http://instantsvc.sourceforge.net/>

type (corresponding to the type of the form element, e.g., text, hidden, select, check box...). The last column sums up all the variables and thus, comprises the reconstructed model, which is supposed to be the application’s background model.

When using the wrapped application via the generated Web Service, all of these variables/values have to be sent to the server, but only a subset of them is specified by the user. For example, variables that are always hidden (`<input type="hidden">`) or those, which remain static values (`<input type="disabled">`) are not contained in the Web Service parameter list, but are nevertheless sent by the Web Service.

Step $s_1$ : Form $f_2 = ("next.php", \mu_{POST})$	Step $s_2$ : Form $f_3 = ("next.php", \mu_{POST})$	Step $s_3$ : Form $f_4 = ("next.php", \mu_{POST})$	Model
("A", $\tau_{text}$ ), $\perp$ , "foo"		("A", $\tau_{text}$ ), "foo2", "foo2"	("A", $\alpha_b$ , {(label: "tag")})
("B", $\tau_{hidden}$ ), "bar", "bar"	("B", $\tau_{hidden}$ ), "bar2", "bar2"	("B", $\tau_{hidden}$ ), "bar3", "bar3"	("B", $\alpha_u$ , {(label: "status")})
("C", $\tau_{submit}$ ), "s", "s"			("C", $\alpha_u$ , {(label: "create")})
	("D", $\tau_{file}$ ), $\perp$ , "-----"		("D", $\alpha_b$ , {(label: "file")})
	("E", $\tau_{submit}$ ), "n", "n"	("E", $\tau_{submit}$ ), "n", "n"	("E", $\alpha_u$ , {(label: "submit")})
		("F", $\tau_{textarea}$ ), $\perp$ , "baz"	("F", $\alpha_b$ , {(label: "comment")})

**Legend**

new

re-occurring

**Form elements**  
(name, type)  
initial value,  
submitted value

**Model elements**  
(name, adornment,  
set of meta  
information)

**Adornments**  
 $\alpha_b$  bound (has to  
be specified)  
 $\alpha_u$  unspecified

Fig. 2. Illustration of the reconstructed background model of a web application

## 2.2 Terms and Concepts

Once the schema (Figure 2) is filled, all relevant data are collected. Therefore, the following information has to be captured.

**Steps/Pages.** A multi-stepped web application comprises several pages. These pages are stored to save the order of proceeding pages.

**Forms.** Forms contain the controls, the user employs to enter or manipulate data. In general, each page contains one or more forms from which one is implicitly selected by the user when he decides to enter data and to click the “submit” button.

**Form elements.** There is a large variety of different form elements<sup>3</sup>. Each element has a *name* and a *value*. Form elements can be initialized with values (which can be perceived as default values), but they are also submitted with a value. Both values may differ from each other.

Moreover, there might exist meta information for each form element that contains additional textual descriptions, e.g., `<label>` or `<class>` elements.

<sup>3</sup> Form controls: <http://www.w3.org/TR/html401/interact/forms.html#h-17.2>

**Model elements.** The application’s background model consists of model elements, which are represented as form elements in HTML forms. Thus, a set of one or more corresponding form elements constitutes a model element. Figure 2 shows model elements as rows in the most-right column. It makes sense for the user to assign a value to one model element when using the wrapped application, rather than to assign it to multiple form elements. Section 3.2 shows how the matching works.

### 3 Analysis of Web Forms

#### 3.1 Phase 1: Monitoring

This section explains how the analysis process is executed, where we follow a supervised approach. Each interaction between the user or—to be more specific—his browser and the web application is monitored. This is technically achieved by using a proxy through which all HTTP requests and responses are routed. After having loaded the web page, the user selects one of possibly several existing forms by filling in values and sends the form. The proxy catches the submission, inspects all form elements of the submitted form (name, description, type, etc.) and stores this information. Afterwards, the request is forwarded to the actual web server and the response is retrieved by the proxy, again. To help the analysis component identify the selected form some information are injected into the web page when retrieved from the web server. When the user submits the form, the proxy does not submit the additional, injected information. Instead it sends a corresponding request with the original form elements/key-value-pairs only. When the user has finished his interaction with the web application, he stops the monitoring process manually.

During the monitoring phase, among others, form elements are inspected. The subsequent matching phase makes use of a multitude of information, which are captured while the monitoring takes place. This information is illustrated in Listing 1.

```

42 <tr>
43   <td>
44     <label for="tel_field">Tel:</label>
45     <input type="text" name="telephone" value="012345..."
46       id="tel_field" class="address_field"
47       disabled="disabled">
48   </td>
49   <td>Insert your telephone number</td>
50 </tr>

```

**Listing 1.** Example snippet from an address entry form

For each form element, its type and its name (line 45) are saved. These pieces of information are the core information of any form element. There might be an initial value, which becomes the default value of the later model element if this form element is its first participant. Furthermore, other meta information might be available. In the example, an `id` and a `class` attribute are given (line 46). However, better descriptions can be found as visible rendered text as in lines 46 and 53. These descriptions can be offered as labels (line 44) or as text in near proximity (line 49) of the form element [10,12]. Additional meta-information such as in line 47 is also saved. This plays an important role when the WSDL description is created.

### 3.2 Phase 2: Element Matching

Model elements consist of one or more corresponding form elements. This correspondence relates to similarities in form elements' names, values, or meta information which permit to match those form elements. The resulting mapping contains the model elements that reflect the background model of the application, which are used in the Web Service interface.

The purpose of the matching process is to assign form elements to an existing model element or – if no appropriate model element can be found – to create a new model element for this form element, respectively. This assignment is based on the information gathered in the monitoring phase: form element names (“schema data”) and values (“instance data”).

For the matching in a given step, all new form elements have to be compared to all existing model elements. Actually, new form elements are compared to the most recently added form elements of the existing model elements. If no appropriate form/model element can be found, a new model element is created.

The focus of this work lies within the retrieval of form structures from web pages and therefore, only a simplistic matching approach is used. Further comparison criteria are possible, e.g., aggregation between different form elements or algebraic manipulations. See Kaljuvee et al. [6] for a comparison between different algorithms.

The comparison comprises two steps. First, a similarity matrix between all possible pairs of form elements and all model elements is created. Note that not model elements, but their last-added form elements are used.

Due to space limitations, we refer to Vogel [11] for details. We used a threshold of 0.6 and the following weights of 0.5/0.75/1.0 for matching values/meta data/-names of two different form elements. However, these weights are preliminary and will be refined in future work.

Given this similarity matrix, a mapping has to be generated in the second step. This is a consistent selection of pairs from this matrix and defines which form elements are assigned to existing model elements or which ones build up new model elements. We employ the *Royal Couples* [8] algorithm.

Once all model elements are updated or created, the Web Service interface definition has to be set up. Not all model elements become input parameters, i.e., are specifiable by the Web Service caller. The ability of model elements

whether and how to be specified is expressed via adornments [14]. Adornments are attributes that state that a variable is (a) mandatory or (b) optional and is arbitrary or fixed-valued (1-out-of-n). See Figure 1 for an intuition. Next to these combinations, a variable can also be unspecifiable, e.g., hidden form fields.

### 3.3 Phase 3: Result Definition

Web Services do not only need a well-defined input provided via the Web Service interface description, they also have to provide a specified format for their output. Web applications can have many different types of results. In most cases, the result is embedded in the HTML page. Depending on the web application, the result consists of a single or a larger, structured message. XPath provides a means to extract simple or well-structured pieces out of HTML pages.

At the end of the analysis phase, the user is confronted with the result page. He provides a name and a short description to outline the service's purpose. Furthermore, he specifies the XPath expression of the relevant part of the result page by clicking on it or explicitly entering the expression. He also decides whether this expression shall be treated as a string (simple), a list or a table (structured).

In case of a table, each row can be converted into a record of key-value pairs and the Web Service in the end returns an array of these records. All cells are cleaned from markup such as `<b>` or `<font>`, so that the desired content is contained in the records.

## 4 Evaluation

A preliminary prototype of the system has been implemented. The prototype has been tested on a selection of web applications from different domains. This section shows which types of applications have been monitored and to which extend the generated Web Services are able to wrap these applications.

### 4.1 Wrapped Web Applications

**Streamer Radiative Transfer System.** As mentioned in Section 1 this tool converts data between different formats, allowing for specification of a couple of conversation parameters.

**Public Transport Advisory.** Most public transportation companies provide a lookup service for connections and fares. On <http://ding.eu/>, the potential passenger can select origin and destination of the intended trip. In the next step, he can define the time of the trip and correct possibly miss-spelled input from the first form. Afterwards, the passenger gets a tabular list with matching trips.

**Online Shopping.** When buying goods online, the shopping cart metaphor comprises a sequence of steps for browsing, selecting, and paying articles, such as at <http://amazon.com/>

**URL Shortening.** Services to shorten URLs need the address that is to be reduced in size and often an optional short name. They are only single-stepped. A representative for these services is <http://tinyurl.com/>

All of the four scenarios have been put into the monitoring phase. They yielded different, but typical results. In the Streamer case, the interaction with the application is correctly reflected in the captured workflow. The form elements of the settings and the data page are aggregated to a distinct background model. The result in this case is of a simple type: it is the full body of the final page. However, the processing of the provided data takes some time for the web application. This is not reflected inside the Web Service. Thus, the result page is requested before the conversation has finished. To resolve this, the Web Service needs an additional statement to wait a specific amount of time before the last step is executed.

In the public transportation scenario, the input opportunities (origin and destination, timestamp) are successfully wrapped in the Web Service while at the same time respecting their adornments (optional and mandatory parameters). The structure of the tabular result is properly reflected in the Web Service's response. Many of the tested transportation applications have some unspecified hidden variables changing their values during the process depending on the provided input. In this case, the created Web Service did not work directly in the generated form, instead the created code had to be adjusted manually to reflect this change.

The Amazon web page actively tries to distinguish between human and machine users and by this, detects the proxy as robot and refuses to fill the shopping cart.

The TinyURL address shortener service can be wrapped without drawbacks. The two fields for the address and the optional alias name are properly recognized and annotated with adornment and description ("url" and "alias").

## 4.2 Results

The results of the wrapping of four scenario representatives are described in a more general point of view. Generally it is obvious that some practical reasons yet prevent the system from generating proper Web Services respectively prevent the Web Services from working as intended.

- The matching algorithm performs well on the provided information. The Web Service interface contains the expected parameters without duplications or missing parameters. They also obtain helpful descriptions which allow a user familiar with the wrapped application to decide on which values to put in.
- Single-stepped web applications impose a smaller number of barriers to the system. They cannot raise issues such as changing URIs or model elements because these are settled before the user enters any case-specific data. There are a couple of services which wrap these single-stepped applications<sup>4</sup> in RSS

---

<sup>4</sup> <http://www.dapper.net/>, <http://feed43.com/>, <http://www.feedyes.com/>,  
<http://ponyfish.com/>

format. With the approach presented here, a standardized interface definition format (WSDL) as well as a standardized messaging format (SOAP) can be used.

This loss of generality might render the approach as impractical for an overall use. However, the system can be taken as a framework to create Web Service stubs. If the monitored user has a developer background, he can employ the generated Web Service code as a foundation for manual adjustments to achieve the originally intended goal. This is necessary for peculiarities such as waiting time (Streamer example) or changes in schema data (Transportation example), described above.

## 5 Related Work

This work incorporates different research areas. The overall task is to provide means to extract data from the Hidden Web [2,10]. To reveal this data, forms have to be filled with meaningful data [7]. For the sake of scalability, this is usually done automatically. However, there are also supervised approaches [3], which use wrappers to assist web crawlers in their information retrieval process.

Moreover, schema extraction and label recognition is a crucial part of this work. Prior work [6,12] presents sets of methodologies to extract labels for form elements as well as descriptions for semi-structured (tabular) data as it appears on result pages, frequently. This is supported by methods to identify the relevant sections in these pages to make the extraction process more effective [13].

Dury et al. approach a quite similar challenge [5]. They monitor the HTTP traffic between a client and a server to infer the behavior of the web application represented as a finite automaton. The authors trace the application's behavior by causing many client server interactions and applying a rule inference engine to find out the conditions, branches, and guards between the automaton's states.

Kabisch et al. [4] exploit web databases by extracting query interfaces. They analyze web forms regarding their text and form field nodes and arrange them into tree structures. The benefit is that non-leaf nodes provide additional information about the meaning of input parameters.

## 6 Summary

In this paper, we presented a novel approach to generate wrappers for multi-stepped web applications. These wrappers are implemented as Web Services, which encapsulate the offered functionality. The HTML forms used as graphical user interface of the applications are analyzed and corresponding elements, contained in these forms, are matched. With this, it is possible to create distinct, comprehensively described parameters for the Web Service. Tabular application results are parsed into objects.

Experiments show that it is indeed feasible to encapsulate multi-stepped applications. The matching algorithm finds corresponding form elements in different

steps and reconstructs the application's background model. However, in practice, the resulting Web Services are not usable out-of-the-box in every case. E.g., delays, redirects, or dynamic `action` attributes have to be overcome manually. We plan to enhance our prototype to handle these cases, too.

We see more future work in supporting different work-flows within the application. Currently, only the monitored process is available. If there are deviations, for instance resulting from specific input data, this should be recognized and the Web Service call should react appropriately. Furthermore, the result specification can be facilitated by discovering the key parts of the result page [13] or using more sophisticated label assignment algorithms [4,12].

## References

1. AbuJarour, M., Craculeac, M., Menge, F., Vogel, T., Schwarz, J.-F.: PoSR: A comprehensive System for Aggregating and Using Web Services. In: International Conference on Web Services (2009)
2. Bergman, M.K.: The Deep Web: Surfacing Hidden Value. Journal of Electronic Publishing (2001)
3. Carme, J., Ceresna, M., Frölich, O., Gottlob, G., Hassan, T., Herzog, M., Holzinger, W., Krüpl, B.: The Lixto Project – Exploring New Frontiers of Web Data Extraction. In: British National Conference on Databases (2006)
4. Dragut, E.C., Kabisch, T., Yu, C., Leser, U.: A Hierarchical Approach to Model Web Query Interfaces for Web Source Integration. In: Very Large Databases
5. Hallal, H.H., Dury, A., Petrenko, A.: Inferring Behavioural Models from Traces of Business Applications. In: International Conference on Web Services (2009)
6. Kaljuvee, O., Buyukkokten, O., Garcia-Molina, H., Paepcke, A.: Efficient Web Form Entry on PDAs. In: World Wide Web Conference (2001)
7. Madhavan, J., Ko, D., Kot, L., Ganapathyand, V., Rasmussen, A., Halevy, A.: Google's Deep Web Crawl. In: Proc. VLDB Endowment (2008)
8. Marie, A., Gal, A.: On the Stable Marriage of Maximum Weight Royal Couples. In: Workshop on Information Integration on the Web (2007)
9. Menge, F.: Generation of User Interfaces for Service Compositions. Master's thesis, Hasso Plattner Institute at the University of Potsdam (2009)
10. Raghavan, S., Molina, H.G.: Crawling the Hidden Web. In: International Conference on Very Large Databases (2001)
11. Vogel, T.: Generierung von Web Services zur Kapselung mehrstufiger Webformulare. Master's thesis, Hasso Plattner Institute at the University of Potsdam (2009)
12. Wang, J., Lochovsky, F.H.: Data Extraction and Label Assignment for Web Databases. In: International Conference on World Wide Web (2003)
13. Wang, J., Lochovsky, F.H.: Data-rich Section Extraction from HTML pages. In: International Conference on Web Information Systems Engineering (2003)
14. Yerneni, R., Li, C., Garcia-Molina, H., Ullman, J.D.: Computing Capabilities of Mediators. In: International Conference on Management of Data (1999)



# Adapter Patterns for Resolving Mismatches in Service Discovery<sup>\*</sup>

Hyun Jung La and Soo Dong Kim

Department of Computer Science  
Soongsil University  
1-1 Sangdo-Dong, Dongjak-Ku, Seoul, Korea 156-743  
hjla@otlab.ssu.ac.kr, sdkim777@gmail.com

**Abstract.** The theme of service-oriented computing is largely centered on reusing already existing services. Service providers model common features among potential applications, realize them as reusable services, and publish in service registries. Service consumers discover appropriate services and subscribe them. In developing application with reusable services, there exists a key technical problem, called mismatch problem which is a gap between the required feature and the feature of a candidate service. The adaptability of available services is a key factor in determining the reusability of the published services by resolving mismatches. Hence, we claim that the design of adapters should be an essential activity for developing service-oriented applications. In this paper, we identify recurring mismatch types in discovering services. And, we present four adapter patterns handling the mismatch problems. By using the adapter patterns, service providers could develop highly reusable services, and service consumers will be able to reuse more services available.

**Keywords:** service mismatch problem, adapter pattern.

## 1 Introduction

As an effective reuse paradigm, the theme of service-oriented computing is largely centered on reusing already existing services. Service providers model common features among potential applications in a domain, realize them as reusable services, and publish in service registries. Service consumers discover appropriate services and subscribe them in their systems. Hence, being able to reuse published services is the most fundamental underlying notion of service-oriented computing.

However, there exists a key technical problem in reusing the services; *mismatch problem*. Let  $FEA_1, FEA_2 \dots, FEA_n$ , be the features needed in developing a target system. And, let  $SVC_1, SVC_2 \dots, SVC_m$ , be the services available in services registries. Service consumers would try to locate the right service  $SVC_j$  for a feature  $FEA_i$  for the target system, and then the discovery will result in one of the following categories;

---

<sup>\*</sup> This research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development.

- Category 1) Finding  $SVC_j$  which fully fulfills  $FEA_i$ . This is the happy case that the located service can be included and reused in the target system, yielding a high reusability.
- Category 2) Finding no  $SVC_j$  which fully fulfills  $FEA_i$ . Since there is no fully matching service, the required feature  $FEA_i$  should be newly implemented.
- Category 3) Finding  $SVC_j$  which partially fulfills  $FEA_i$ . Due to the partial matching, the located service  $SVC_j$  may or may not be reusable in the target system depending on its adaptability. This is the motivation for our research.

Hence, *mismatch* is defined as the gap between the required feature  $FEA_i$  and the feature of a candidate service  $SVC_j$ . *Adaptation* is an activity of adapting such service  $SVC_j$  to make it fulfill the required feature  $FEA_i$ . Hence, the adaptability of available services is a key factor in determining the applicability and reusability of the published services. Without adaptation capability, services of partial fulfillment (in the category 3) could not be utilized in developing target systems. Hence, we claim that the design of adapters should be an essential activity for developing service-oriented applications.

Like the motivation for object-oriented design patterns [1], we observe recurring problems of mismatch in discovering services, i.e. typical patterns of service mismatches. Accordingly, we believe that developing patterns for designing adapters which can effectively resolve the recurring mismatch problems would be feasible.

In this paper, we first identify recurring mismatch types in discovering services, and define the roles of service providers and consumers to support adaptability. Then, we specify each adapter pattern. For the paper organization, we give a survey of related works in section 2. And, we present taxonomy of service mismatches, and roles of service providers and consumers in section 3. Each pattern is specified in details in section 4, and assessment work is presented in section 5. By using the adapter patterns, service providers could develop highly reusable services, and service consumers will be able to reuse more services available.

## 2 Related Works

Kongdenfha and his propose the aspect-oriented approaches to adapt mismatches in *interfaces* and *protocols* [2]. First, they define interface and protocol mismatch patterns including adaptation logic that resolves the mismatch. Then, they present applying aspect-oriented approach to modify the services. At runtime, adaptation logic is generated and woven into the adapted service. To do this, they define three mismatch patterns including *Signature Mismatch Pattern*, *Missing Message Pattern*, and *One-to-Many Pattern*. Although their approaches are presented in much detailed level to show the applicability and practicability, this work only focuses on the interface and workflow mismatches.

Benetallah's work proposes methods for developing adapters which can resolve mismatches in service interfaces and business processes [3]. A mismatch pattern specifies situation, information for adapter instantiation, and pseudo-code. Each pattern is provided for resolving mismatches of service interface and business process

level. This work proposes the mismatches confined to service component and business process and the steps to resolve mismatches without giving details for developing the adapters.

Sam's work proposes methods for identifying service mismatches and dynamically adapting the services [4]. The method for identifying mismatches utilizes *service configuration* which is a specification of syntactic aspect, semantic aspect and constraints of service interfaces. The method for adapting services utilizes *context association rules* to transform interfaces of published services to the service interfaces expected by consumers. This work mainly deals with interface mismatches and interface transformation.

Erl presents various kinds of patterns for realizing service-oriented architecture [5]. Most of the patterns are related to designing services, but they also address how to resolve differences in data formats, data models, and communication protocols by using three patterns; *Data Format Transformation*, *Data Model Transformation*, and *Protocol Bridging*. The mismatches are only for interface-level information, and the proposed patterns need to be enhanced in a practical manner.

### 3 Fundamentals of Service Mismatches

#### 3.1 Types of Service Mismatches

Services are published in registries with their *Service Specification*, which becomes vital information for service discovery. Service consumers look up the specification and may find mismatches. Hence, we can derive typical mismatches from the key elements of the service specification. A service is commonly specified with its interface, functionality, and provided Quality of Service (QoS)[6].

*Service Interface* element of a service has *Provide Interface* and *Required Interface*. *Provide Interface* specifies the functionality delivered by the service, and *Required Interface* specifies external services or objects which should be plugged into the current service to make the service fully functional [7].

*Service Functionality* consists of *External Behavior* and *Internal Behavior* of operations in the service. The external behavior is typically specified with a functional overview, pre-condition, post-condition, invariant, and constraints. The internal behavior of a service operation specifies the logic, algorithm, rules, and any other internal details of the operation, and hence it is typically invisible to outside.

*QoS* consists of one or more quality attributes and their expected values at runtime, which is typically specified in *Service Level Agreement*. Hence, services are specified with certain levels of quality attributes.

Types of mismatches can be derived for each element in *Service Specification* by considering its plausible abnormality, i.e. *Interface Mismatch*, *Functionality Mismatch*, and *QoS Mismatch*. Furthermore, each mismatch type can be specified into more specialized mismatch types. For example, *Interface Mismatch* is specialized into *Signature Mismatch* and *Semantic Mismatch*. Fig. 1 represents all the mismatch types which are derived from the each element of the service specification.

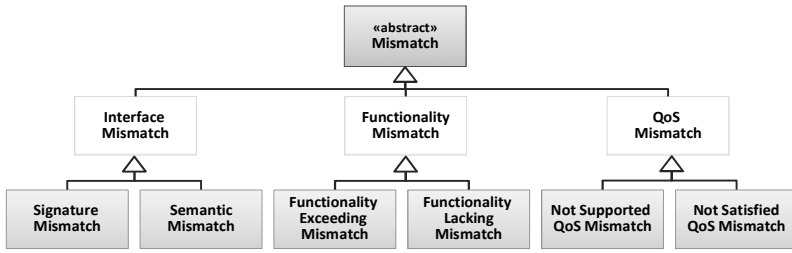


Fig. 1. Hierarchy of Partial Mismatches

The rest of the mismatch types are further specified within adaptation patterns in section 4.

### 3.2 Roles of Service Providers and Consumers for Adaptability

When developing applications with services, service consumers can face the mismatch problems so that they take responsibility for *adapting* the services. Therefore, service consumers develop static adapters by realizing the most appropriate *adapter pattern*. To do so, they identify the type of mismatches, design how to resolve the mismatches, and utilize the service after adapting it.

But, it is not always possible to resolve the mismatches since the service consumers should access enough information to develop the suitable adapters. That is, if service providers do not provide methods for static adaptation, service consumers have a limitation on adapting the services. Therefore, service providers *enable* the consumer to resolve the mismatches by providing sufficient service information. To enable the service adaptation, it is feasible for the service provider to define *required interface* which can assert consumer-specific variants [8].

## 4 Specification of Adapter Patterns

The mismatches identified in section 3.1 occur repeatedly when the services are invoked by various service consumers. To deal with these repeated mismatches, we propose several adapter patterns in terms of overview, applicable situation, structure, collaboration, consequence, and instruction.

Note that we do not cover the adapter patterns for resolving *Not Supported QoS Mismatch* and *Not Satisfied QoS Mismatch* since QoS cannot be managed at the structural and dynamic designs but the architecture design. Hence, we suggest utilizing architecture design approaches such as [9].

### 4.1 Signature Matching Pattern

**Overview:** This pattern is to adapt the interface of a candidate service so that the interface expected by a consumer and the interface of a provided service match.

**Applicable Situation:** When there is a candidate service but its interface do not fully match to the interface expected, this pattern can be used to resolve the mismatch.

In SOA, service interfaces are pre-defined by providers, and they cannot be re-defined by service consumers. There can be a situation where the interface of a candidate service does not fully match to the interface expected by a consumer application. We call this situation as *Signature Mismatch*.

**Structure:** The *Adapter* pattern such as [1] can be used. As shown in Fig. 2, *Caller* which is a class in an application defines a required interface, *op1()*. *Callee* provides a candidate service having an interface, *op2()*. Then, an *Adapter* (i.e. *adaptedClass*) resolves the mismatch by transforming signatures as shown in the note in the figure.

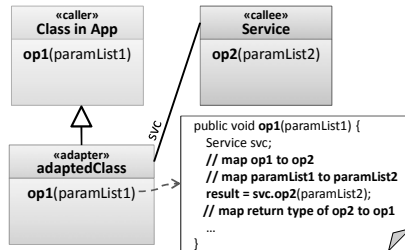


Fig. 2. Object Model for Signature Matching Pattern

**Collaboration:** At runtime, the participants in this pattern work together to resolve the signature mismatch as shown in Fig. 3. If there is more than one operation to transform the expected operation to provided one, there may be several self invocations in the instance of *adaptedClass*.

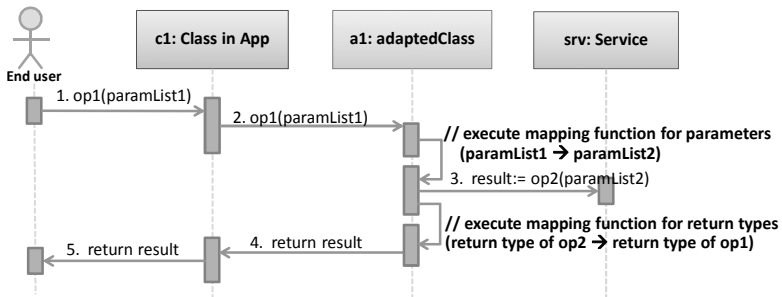


Fig. 3. Dynamic Model of Signature Matching Pattern

When the user invokes the method *op1(paramList1)* in the *Class in App*, its overridden method in *adaptedClass* is invoked by the principle of dynamic binding. As shown in the algorithm for the method in the figure, it adapts the interface mismatch by using mapping function for input parameter. Then, the method *op1(paramList1)* invokes the method in the *Service*, *op2(paramList2)*.

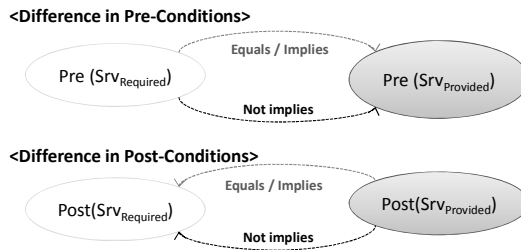
**Consequence:** Candidate services which have mismatches with services expected not becomes reusable, increasing the reusability of candidate services. There can be a minor performance penalty for having to invoke services through the *adapters*.

**Instruction:** To apply this pattern, first, we identify the mismatch between expected and provided interfaces which is *mismatch on the names of operations, mismatch on the datatypes of parameters and return value, mismatch on the orders of parameters, and mismatch on value ranges of parameters*. Second, we define a mapping rule for each mismatch. Mapping rules may vary from simple type casting to complex calculation. The next step is to add a code segment invoking the adapter in the application.

### 4.2 Semantic Matching Pattern

**Overview:** This pattern is to adapt a candidate service to make its semantic fit to the semantic of an expected service.

**Applicable Situation:** When developing the service-oriented applications, behavioral semantics of the service interface may not be satisfied with the consumer’s expectation although the signature is compatible. Being different in semantics implies that there are minor gaps in *pre/post conditions* and *invariant*.



**Fig. 4.** Detailed View of Semantic Mismatches

Fig. 4 explains the applicable situations of this pattern in detail. Let  $Pre(Srv_{Required})$  be the pre-condition of service  $Srv_{Required}$  and  $Post(Srv_{Required})$  be the post-condition of service  $Srv_{Required}$ .

When  $Pre(Srv_{Required})$  is equal to or implies  $Pre(Srv_{Provided})$ , the target service can be applied. Otherwise, the service is not usable. Consider an example where  $Pre(Srv_{Required})$  says  $A$  should be larger than 0, and  $Pre(Srv_{Provided})$  says  $A$  is equal to or larger than 0. A problematic case is when  $A$  is 0, which requires an adaption.

Differences in the post-condition are opposite to ones in the pre-condition. Consider an example where  $Post(Srv_{Required})$  says  $x$  should be equal to or larger than  $y$ , and  $Post(Srv_{Provided})$  says  $x$  is larger than  $y$ . In this case, a problem occurs when  $x$  is equal to  $y$ , which requires an adaption.

As the third element describing the semantic, an invariant is a condition which must always hold while a service is being executed. Since services are provided as a black box, it would generally be not feasible to adapt invariants. This situation is called *Semantic Mismatch* as depicted in Fig. 1.

**Structure:** Adapting semantics of blackbox form of services is not trivial due to the limited accessibility and visibility. To mitigate gaps in the pre-conditions, adaptation will apply to both consumer and provider sides. For consumer side, the underlying mechanism of *Signature Matching Pattern* can be well applied. For provider side, we apply a plug-in mechanism [8]. This method can only be applicable when *required interface* is pre-defined. Through the required interface, service consumer can plug an external object which carries the consumer-specific logic, i.e. *variant*.

Unlike the pre conditions, the differences in the post conditions are only resolved on the provider side by using plug-in mechanism.

Fig. 5 depicts how plug-in mechanism is applied to this pattern. Service has a static attribute, *vp*, of *PlugType* type which is set through *plugIn* operation Class in App sends a message with the its own plug-in object, *myObj*.

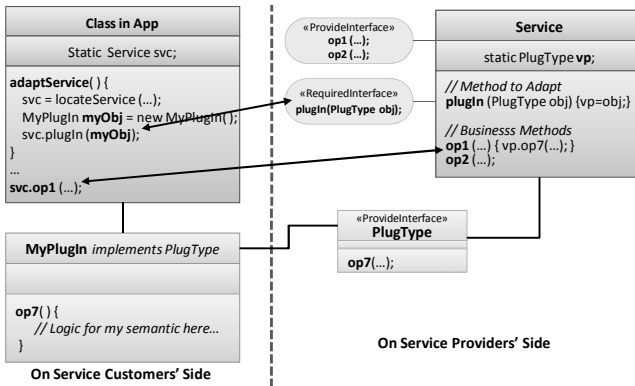


Fig. 5. Design of Semantic Matching Adapter with Plug-in Object

The Plug in object, *MyPlugIn*, embodies the adaptation functionality. To resolve this mismatch, the plug-in may perform additional functionality that reads the results provided by the service in reverse order. Note that in this plug-in mechanism, most of the functionality is fulfilled by the services provided and the small amount of the functionality is satisfied with the plug-in object.

**Collaboration:** When *Class in App* invokes the *Service*, the *Class in App* first creates its own plug-in object, an instance of *MyPlugIn* and then passes the object (i.e. *myObj*) as an argument by using the required interface, *plugIn(PlugType obj)*.

Then, on the service side, the argument *myObj* is copied to a static attribute, *vp*, for a variation point. Since then, whenever *op1()* gets invocation, the *Service* internally invokes *op7()* defined in *MyPlugIn* implementing consumer-specific functionality.

**Consequences:** Utilizing this pattern increases the reusability of candidate services, but there can be a minor performance penalty for invoking services after creating the plug-in objects. And, there may be some limitations to develop plug-in object since this pattern is only applicable when a service providers offers additional mechanism such as *Required Interface* which enables the object to be plugged in.

**Instruction:** To apply this pattern, we suggest the following process. First, we identify the mismatch the mismatch between expected and provided semantic. Second, we define a mapping relationship by applying semantic adaptation methods available such as [10] and [11]. When several semantic mismatches are adapted, some complications among the adapted items may occur.

The next step is to add a code segment invoking the adapter in the application. The place of the code segment for the adaption is different whether the semantic mismatches are about precondition or postcondition. In the former case, the code segments put before the actual service operation invocation. And in the latter case, the code is placed behind the service invocation.

### 4.3 Functionality Enhancing Pattern

**Overview:** This pattern is to supplement the functionality which is required by the service consumer but not offered as a service.

**Applicable Situation:** When service consumers look up the services based on their functional requirements, it is inevitable to find the service whose functionality does not fully match consumer’s expectation. In this case, most of the functionalities are provided by the services but a little amount of functionality is not provided. This situation is called *Functionality Lacking Mismatch* as shown in Fig. 1.

**Structure:** To resolve *Functionality Lacking Mismatch*, we suggest using *Decorator pattern*[1] to append the needed functionality as shown in Fig. 6.

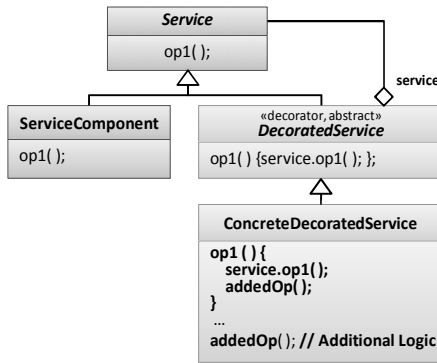


Fig. 6. Object Model of Functionality Enhancing Pattern

In the figure, the *Service* defines the interface for the *ServiceComponent* that provides its functionality but may have lack functionalities of the required ones. Hence, we define a service with an additional functionality, called *DecoratedService*. *DecoratedService* aggregates a reference to its superclass in recursive manner. At runtime, an instance of *ConcreteDecoratedService* substitutes its superclass object, and the *op1()* in the class embeds an additional functionality in the method *addedOp()*.



**Collaboration:** When the consumer invokes *op1()* of *ServiceComponent*, *DecoratedService* gets the invocation. In *op1()* defined in *DecoratedService*, it first invokes the *ServiceComponent* and then execute additional functionalities.

**Consequence:** The functionality-inappropriate service becomes useable so that the reusability of the service will be increased.

**Instructions:** To apply this pattern, we suggest the following process. First, we identify the lacking functionality by comparing consumer’s *functional requirements* with *service functionality*. Second, we design the functionality. The next step is to add a code segment invoking the adapter in the application. The code segments for the adaptation are put after the actual service operation invocation.

### 4.4 Functionality Disabling Pattern

**Overview:** This pattern is to disable the unnecessary functionality of the service so that the expected functionality and the one of a provided service match.

**Applicable Situation:** This pattern is applied when the functionality of the service provides additional one required by the service consumer. Only if the accidental invocations of functionality create potential problems such as integrity violations and undesirable side-effects, this is problematic. This situation is called *Functionality Exceeding Mismatch* as shown in Fig. 1.

**Structure:** The *Proxy* pattern such as [1] can be used. As shown in Fig. 7, there are two subclasses of *ServiceInterface* implementing *op2()*. Functionalities of *op2* are almost similar, but *ServiceComponent* provides additional functionality. Hence, *ProxiedComponent* redefines the *op2()*, which *op2()* of *ServiceComponent* is first invoked and disables additional functionality by throwing exceptions.

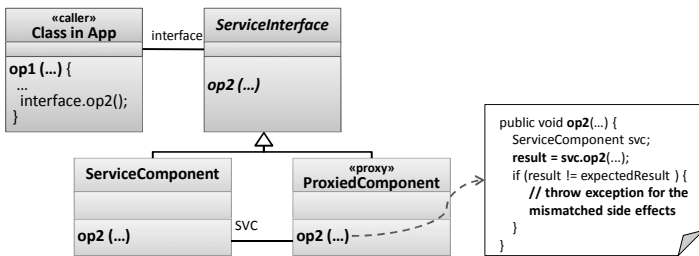


Fig. 7. Object Model of *Functionality Disabling Pattern*

**Collaboration:** When the user invokes the method *op2()* in the *op1()* of *Class in App*, its overridden method in *ProxiedComponent* is invoked. As shown in the algorithm in the figure, it internally invokes *op2()* in *ServiceComponent* and disables the additional functionality. Hence, additional functionality is no longer effective.

**Consequence:** The functionality-inappropriate service becomes useable so that the reusability of the service will be increased.

**Instructions:** To apply this pattern, we suggest the following process. First, we identify the exceeding functionality which is not required by the service consumer by comparing the functionality requirement and service functionality. Second, we check whether the exceeding functionality can be disabled or not. If not, such as side effects on the database, do not consider applying this pattern. Third, we design the way to disable the exceeding functionality such as treating as an exception. Then, we develop a proxy-like adapter which provides the same interface in the *ProxiedComponent*. The next step is to add a code segment invoking the adapter in the application.

## 5 Assessment and Conclusion

The adapter patterns proposed in this paper result in high reusability and profitability for the service providers and high ROI for the service consumers. We also should consider the performance penalty and additional efforts to develop adapters when adopting the adapter patterns. Hence, if a consumer's application imposes strict QoS requirements, it may be better not to use the service after adaptation. And, the consumers should analyze business cases by comparing the cost to develop application without services and with services and adapters.

In this paper, we identify six types of recurring mismatch in discovering services and define the roles of service providers and consumers to support adaptability. Each adapter pattern is specified in terms of its overview, applicable situation, structure, collaboration, consequence and instruction. By using the adapter patterns, service providers could develop highly reusable services, and service consumers will be able to reuse more services available.

## References

- [1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley Professional, Reading (1994)
- [2] Kongdenfha, W., Motahari-Nezhad, H.R., Benatallah, B., Casati, F., Saint-Paul, R.: Mismatch Patterns and Adaptation Aspects: A Foundation for Rapid Development of Web Service Adapters. *IEEE Transactions on Services Computing* 2(2), 94–107 (2009)
- [3] Benatallah, B., Casati, F., Grigori, D., Motahari-Nezhad, H.R., Toumani, F.: Developing Adapters for Web Services Integration. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 415–429. Springer, Heidelberg (2005)
- [4] Sam, Y., Boucelma, O., Hacid, M.: Web Services Customization: A Composition-based Approach. In: In Proceedings of IEEE International Conference on Web Engineering (ICWE 2006), pp. 25–31 (July 2006)
- [5] Erl, T.: SOA Design Patterns. Prentice Hall, Englewood Cliffs (June 2008)
- [6] MacKenzie, C., Laskey, K., McCabe, F., Brown, P., Metz, R. (eds.): Reference Model for Service Oriented Architecture 1.0, OASIS Standard, October 12 (2006), <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> (accessed September 21, 2009)
- [7] Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual, 2nd edn. Addison-Wesley, Reading (2005)

- [8] Kim, S.D., Min, H.G., Rhew, S.Y.: Variability Design and Customization Mechanisms for COTS Components. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 57–66. Springer, Heidelberg (2005)
- [9] Rozanski, N., Woods, E.: Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. Addison-Wesley, Reading (2005)
- [10] Giunchiglia, F., Shvaiko, P.: Semantic Matching. *The Knowledge Review* 18(3), 265–280 (2003)
- [11] Yeh, P.Z., Porter, B., Barker, K.: Using Transformations to Improve Semantic Matching. In: Proc. 2nd Int’l Conf. Knowledge Capture (K-CAP 2003), pp. 180–189 (2003)

# Lightweight Composition of Ad-Hoc Enterprise-Class Applications with Context-Aware Enterprise Mashups

Florian Gilles<sup>1</sup>, Volker Hoyer<sup>1,2</sup>, Till Janner<sup>1,2</sup>, and Katarina Stanoevska-Slabeva<sup>2</sup>

<sup>1</sup> SAP Research St. Gallen, Blumenbergplatz 9, 9000 St. Gallen, Switzerland

<sup>2</sup> University of St. Gallen, `=mcminstitute`, Blumenbergplatz 9, 9000 St. Gallen, Switzerland  
{florian.gilles, volker.hoyer, till.janner}@sap.com,  
katarina.stanoevska@unisg.ch

**Abstract.** The huge demand for ad-hoc and situational enterprise-class applications led to a new kind of Web-based applications, known as enterprise mashups. End users from the business units with no programming skills are empowered to combine and reuse existing company internal and external resources within minutes to new value added applications. In order to handle the growing number of mashable components, we propose a context-aware concept for enterprise mashups that supports users to find relevant components according to their current situation and to compose them automatically. The designed context model which is structured in the three domains agent, computing and environment is implemented in the SAP Research RoofTop Marketplace prototype to demonstrate its applicability and business benefits.

**Keywords:** Enterprise Mashups, Context-Awareness, End-User Development.

## 1 Introduction

A new trend for software development paradigm, known as enterprise mashups, has gained momentum in the recent years. They have the potential to bridge the gap between the automation transaction and peer production world. At the core of the mashup paradigm are two aspects: First, the empowerment of the end user to cover ad-hoc and long tail needs by reuse and combination of existing software artefacts; and second, broad involvement of users based on the peer production concept. According to Yorchai Benker, who coined the term peer production, “*it refers to production systems that depend on individual action that is self-selected and decentralized rather hierarchically assigned*” [1]. Thereby, the creative energy of a large number of people is used to react flexibly to continuous and dynamic changes of the business environment. Instead of long-winded software development processes, existing and new applications are enhanced with interfaces that are provided as user friendly building blocks. The explosive growth of these mashable components from company internal as well as external resources requires an efficient and agile organization [2]. Existing research efforts focus mostly on the technical composition of these components. However, end users from the business units with no or limited programming skills should be assisted by mashup systems in a passive manner to retrieve relevant components and to compose them by connecting their input and output ports.

*By means of a business scenario, we motivate the practical challenge: A sales manager (Max Meier) is usually responsible for maintaining knowledge of his company's products and liaising with customers. Although he has not much technical knowledge and because asking for support from the IT department would cost too much time, he decides to build his own enterprise mashup. Quite fast, he gets confident with the new platform and has success in building his first mashup which combines Customer Data with current Selling Data. Now, he wants to extend his first mashup, but he has no idea what other services would make sense in his context. One option would be to browse in the existing catalogue. But it's hard to find components which best meet his current business needs.*

*Components which are used by his colleagues from the same department (e.g., Peter Mustermann) or industry would be also relevant for our sales manager Max. Although adding new mashable components is quite easy, connecting components is the next problem occurring. Often the components have a lot of ports and it's difficult to decide how they can be connected with the correct inport and outport. From the viewpoint of Max, the platform should support him in providing relevant mashable components according to his current context. In addition, after selecting a component, the platform should automatically connect the components.*

The goal of this research paper is to fill this gap by designing a concept of how end users can be supported to navigate through the growing enterprise mashup ecosystem in order to adapt their individual working environment according to their context. The general research questions guiding this study are how to model the context space for enterprise mashups as well as how the context information can be organized.

The remainder of this study is organized as follows: After discussing related work in the area of enterprise mashups and context, section three presents the designed context approach. Section four demonstrates the implementation by means of the SAP Research RoofTop Marketplace prototype. A brief evaluation and a summary close this present paper.

## **2 Background and Related Work**

### **2.1 Enterprise Mashups – Definition and Characteristics**

*“An enterprise mashup is a Web-based resource that combines existing resources, be it content, data or application functionality, from more than one resource by empowering the actual end users to create individual information centric and situational applications” [4].* By simplifying concepts of Service-Oriented Architecture (SOA) and by enhancing them with the peer production philosophy, enterprise mashups focus generally on software integration on the user interface level instead of traditional application or data integration approaches. With the assistance of a layer concept, the relevant terms can be structured in an Enterprise Mashup Stack which consists of the elements resource, widgets, and mashups.

**Resources** represent actual contents, data or application functionality. They are encapsulated via well-defined public interfaces (Application Programming Interfaces; i.e., WSDL, RSS, Atom, etc.) allowing the loosely coupling of existing Web-based resources – a major quality of SOA. These resources are created by traditional developers who are familiar with technical development concepts. The layer above contains **widgets** which provide graphical and simple user interaction mechanism abstracting from the underlying technical resources. The creation of the widgets is done by consultants or key users from the business units who understand the business requirements and know basic development concepts. Finally, the end users from the business units are empowered to combine and configure such visual widgets according to their individual needs, which results in a **mashup**. According to the motivated scenario in the first section, we focus on the mashup layer in this paper.

## 2.2 Context and Context Awareness

Context is an important utilized source of information in interactive computing. In order to use context efficiently and to build context-aware applications, the term context needs to be defined. The common aspect of existing definitions [5, 6] is that they all try to define the term by simply giving examples for context. Often it is associated with location, but [7] pointed out that context is more than location. Besides conditions and infrastructure, *"location is only one aspect of the physical environment"* [7]. However, the above mentioned two types of definitions are too specific and show a strong focus on location which makes them difficult to apply. In order to describe situation, state, surroundings, task, etc., a more general definition of context will be needed. A definition which fulfills all these aspects, is the definition given by [8]: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. This definition points out the important aspect, that *"context is any information that can be used to characterize the situation of an entity"*. Systems which make use of context in order to provide context-related information to the user are called context-aware [8]. This definition provides an easy and simple way to conclude whether an application is context-aware or not.

## 2.3 Enterprise Mashups and Context

In order to interpret context and visualize the definition of context, a flexible and extensible context model is required. As described in the definition of context, an entity is a person, place, or object. [3] stated out, that these context entities can be structured along the following three domains: *Agent Domain* (Who are you?), *Computing Domain* (What resources?), and *Environment Domain* (Where are you?).

This kind of structure includes interrelation between entities that are within one domain and also interaction between the domains. Another important aspect is that context information is dynamic by nature. [5] claim that the important aspects of context are where you are, who you are, and what resources are nearby. However, this description only includes location and identifies information. The aspect of time is

completely missing. As a consequence of this, time needs to be included in our model, to characterize the situation in the past and future.

Concrete context entities inside those domains can now be expressed as 4-tuples [4]. A tuple is a sequence of specific values which are called the components of the tuple. The components of the 4-tuple result in "entity name", "feature", "value" and "time". Each context entity is identified by its unique name. For the different entities various features can be defined. The two feature categories are internal and external features. Internal features are characteristics inside the entity in its domain. External features describe the context information concerning the interaction of the entity with other entities, i.e. activities. The component *time* is mainly used to record history of context entities and to predict the future situation accordingly. The following list shows how this domain model is applied:

*(MaxMeier, isColleagueOf, PeterMustermann)*  
*(MaxMeier, isUsing, CustomerMashup, 2009/03/01)*  
*(MaxMeier, isUsing, GlobalStartPage, 2009/03/01)*

The three examples describe concrete context entities for the agent domain. The first example demonstrates an interaction of entities inside the domain. According to the definition of tuples for context representation, "MaxMeier" represents the entity name, "isColleagueOf" the feature, and "PeterMustermann" the value. Furthermore, this tuple has an internal feature, which means an interaction inside its domain. Because of that, the entity's value must be another entity, in this case "PeterMustermann". The item representing the time is missing in this tuple because it's not necessary to know the exact time of this activity. In contrast to the first example, the other two entities describe an interaction between domains. In the first case, it's the computing domain and in the second case it's the environment domain.

### 3 Context Aware Enterprise Mashups

#### 3.1 Context Model

After obtaining the potential context entities, these entities need to be categorized to the three domains agent, environment, and computing. In the terminology of enterprise mashups, the *agent domain* represents all registered human users, who are eligible to use the enterprise mashup platform. This domain also includes business information like department, industry, country, and position. Due to the fact, that users are using different development platforms for their enterprise mashup applications, the idea of the *environment domain* is to express where the enterprise mashup is built or executed. Besides execution inside a Customer Relationship Management (CRM) page, the prototype can also be executed inside a sales order environment. What actual widgets are used to build enterprise mashups with a special business purpose is shown by the *computing domain*. It mainly includes the widgets and the connection of the in- and output. The following table summarizes these findings and the categorization of these context entities into the three context domains.

**Table 1.** Context Entities of Enterprise Mashups***Agent Domain***

User	Person creating and executing enterprise mashups
Industry	Industry (i.e. Manufacturing)
Department	Specialized division of the large business organization where the user is working at (i.e. Sales Department)
Country	Country where the department is located at
Position	Status which the user can hold in his department (i.e. Sales Manager)

***Computing Domain***

Enterprise Mashup	Web-based application which consists of different connected Widgets
Widget	Visual representation of a resource
Port	Identifies the connection point between widgets
Business Purpose	Purpose that is desired of the enterprise mashup

***Environment Domain***

Environment	Environment where the enterprise mashup is created or executed
-------------	--

**3.2 Context Organization and Layers**

In contrast to most other typical and traditional applications, which process the input to an according output based on a simple function, context-aware systems need also include and interpret context data to provide relevant information to the user. The process of interpreting raw context data to relevant context information can be described with three context layers. The bottom layer, the *Source Data* layer is sourced with raw data, like sensed or profiled data. This layer considers every bit of information the system gets and strictly excludes interpretation of data which is done in higher layers. Sensed data is obtained with sensors which provide the physical environment in which the enterprise mashup is executed for example. Profiled data can be for example user's preferences about a special situation. With this source of information, profiled or sensed data, it's possible to generate and build *Context Facts* related to the situation of the entities. Context facts are defined as something that actually exist and can be verified as true. It interprets the raw information from the Source Data layer to the first meaningful context information the first time. Examples for context facts are "Max Meier is using the Customer Data Mapping mashup", "The Customer Data Mapping mashup is executed in the CRM Start Page", or "Customer Data Mapping mashup uses the Customer Data Widget".

By describing context at a higher level than facts, new *situations* can be interpreted. Through ordering and weighting with special scores for each context fact, confidence of situations can be achieved. Concerning to the context facts, an example for a situation would be: "Max Meier recommends the widget Google Maps as the best widget for mapping addresses on a map in the CRM Start Page". This example is based on interpretation of the context facts and can be used in order to express context



characterized situations and to extract relevant information, like recommended widgets for users building enterprise mashups in the same CRM Start Page.

### 3.3 Context Facts

Most context information is highly dynamic by nature and makes it hard to create and apply adequate models. In order to provide an adequate model for further practical exploration, the Context Modeling Language (CML) is applied [9]. This modeling approach reformulates modeling concepts as extensions to the Object Role Language and assist designers with the task of exploring and specifying the context requirements of context aware applications, by providing a graphical notation for describing types of information.

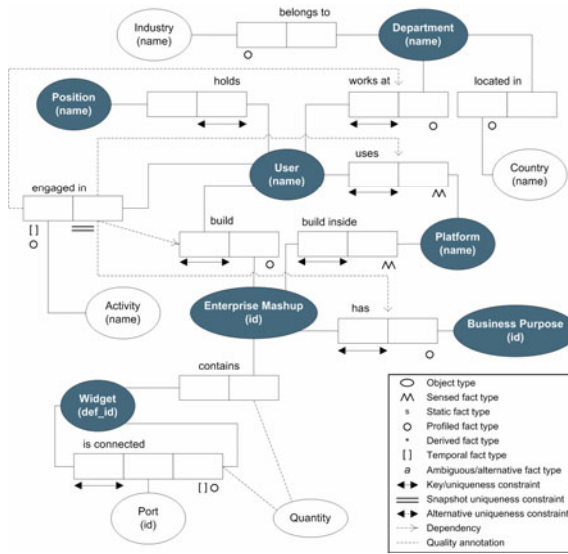


Fig. 1. Enterprise Mashup Context Facts

The figure above models context information and types. All necessary context entities are represented by a solid ellipse. The relationship between these entities is defined by roles and an association which is symbolized by connected boxes, one box for each role. Besides trying to visualize the interaction and relationship between the different context entities from table 1, it also includes capturing capabilities for user activities in the form of a fact type that covers historical activities. Besides the historical point of view, user activities also include the used environment, the user’s workplace and as well as the business purpose of the executed and built enterprise mashup. Due to the fact, that each role in the above figure also represents a context fact in form of a relational model, it’s possible to easily extract these context facts and use them for the base of the context-aware application.

### 3.4 Context Situations

Context situations are defined in terms of context facts and logical connectives (and, or, not, exists, forall). They can be unowned, owned by one entity, or owned by multiple entities. Context-aware systems in ubiquitous computing have the ability to determine and compute related context for situated behavior (also known as *Situated Computing*) [10]. According to the determined context facts in the previous section, a situation can be for example that the widget "Google Maps" is a recommended widget to display the customers address on a map. Without interpretation, the according context facts are only describing the quantity of how often the inport of the widget "Google Maps" is sourced with the address from a customer. But through interpretation, it's possible to say that if it's often used in this context, it could also be a recommended widget for other users building an enterprise mashup which includes customer data. The three context situations which are considered in this paper are:

1. **Recommendation of widgets by colleagues**
2. **Recommendation of widgets by environment**
3. **Recommendation of widgets by business purpose**

Each of the three recommendations is represented by a list and has a different focus on the used and processed entities. The first situation contains recommended widgets by colleagues and focuses on the agent domain. Besides the users social network, the users department, industry, country, and the position held by the users are mainly processed in order to generate the list. "Where" the enterprise mashup is built and executed is the main focus for the calculation on the second situation. It considers the different development platforms and mainly recommends different widgets for each environment. The last context situation focuses on the computing domain and considers all widgets which already have been used for the same business purpose.

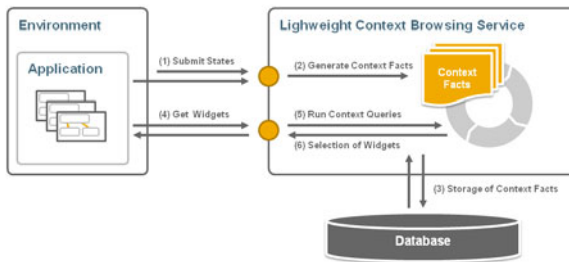
For each list of the context situation, the widgets need to have a special ordering which should depend on the importance of the widget, as well as how many times the widget has been recently used in enterprise mashups. The widgets quantifier and context related importance is achieved by calculating a scoring function. The scoring considers different parameters, like the users position in the department, the department itself, its industry, country, and as well as the business purpose and the development environment currently used. The following pseudocode tries to state out the main aspect of the used scoring algorithm:

```
For each connection in Context Facts where Widget is involved
    Calculate percentage of criteria multiplied with according
    weight
```

The algorithm tries to consider all connections where the queried widget is involved. For each of those connections, a score is calculated based on the percentage of occurrence in a special context. After that, the list is sorted by that score, in order to have more context-related widgets listed at the top and less context-related widgets at the bottom.

## 4 Demonstration: SAP Research RoofTop Marketplace

The SAP Research RoofTop Marketplace prototype is an intuitive Web-based application based on AJAX that enables end users to create ad-hoc enterprise-class applications. To demonstrate the applicability of the context model, we have implemented our concept in the SAP Research RoofTop Marketplace prototype. In order to make use of context related information, the RoofTop Marketplace client is linked to a service, called “Lightweight Context Browsing”. This service is responsible to gather and connect important context-related information from the repository which manages all widget or enterprise mashup information.



**Fig. 2.** Lightweight Context Browsing

The Lightweight Context Browsing service considers already stated out important issues in context organization and context interpretation from raw source data to context relevant information. The following six steps describe how this service interacts with the client and the prototype:

**Step 1: Submit States:** Both, the application and the environment are submitting their states to the Lightweight Context Browsing service whenever an enterprise mashup is saved by the user. These states include information about the connection of widgets as well as information about the environment, industry, country, position, and business purpose of the enterprise mashup.

**Step 2: Generate Context Facts:** The submitted context-related information about the widget connection is then used to generate a context fact object.

**Step 3: Storage of Context Facts:** For persistent storage and later retrieval, the generated context fact is saved in a database.

**Step 4: Get Widgets:** Whenever the application needs a list of follow up widgets, it can submit requests to the according Lightweight Context Browsing service. Due to the fact, that the recommendation of widgets is split in three categories, three requests are sent to the service, after the user added the widget to the design view of the enterprise mashup platform.

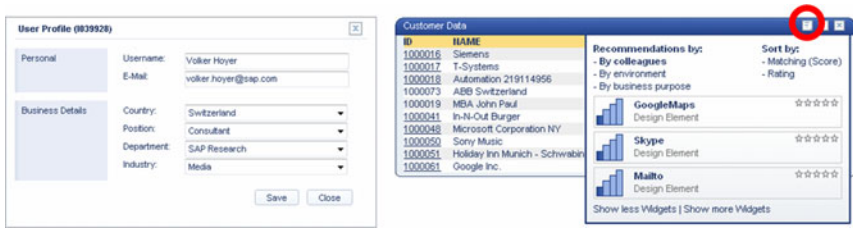
**Step 5: Run Context Queries:** After submitting these requests, the context queries are executed on the Context Facts in order to extract recommended and relevant

widgets. Extraction is based on relevance of the widget to the current context. To list most important widgets at the top of each list, the ordering of widgets is done by applying the previously described scoring algorithm.

**Step 6: Selection of Widgets:** At the end, the selection of recommended widgets is passed back to the application where they are presented to the user and from where the user can decide, what widget he wants to add for extending his enterprise mashup.

## 5 Evaluation and Discussion

As motivated in the introduction, the sales manager in our scenario will usually be responsible for maintaining knowledge of his company's products and liaising with customers. Before starting to build the first enterprise mashup, he has to maintain his user profile which includes his business details like the position, the industry, the country, or the department he is currently working at.



**Fig. 3.** User Context and Lightweight Context Browsing Menu

Quite fast, the sales manager added the Customer Data widget to the design view, to get a list of all customers of his company. However, this step was quite easy, but to extend his enterprise mashup he has not much and detailed knowledge about other widgets which would make sense for his context. Because of that, the sales manager clicks on a little button at the top of the Customer Data widget. This button is available at all widgets. The upcoming popup shows lists of recommended widgets from which the user can choose may it “recommendation by colleagues”, “recommendation by environment”, or “recommendation by business purpose”.

The sales manager is interested in widgets which have been often used by his colleagues in the same department (“recommendation by colleagues”). The presented list of recommended widgets includes “Google News”, “Flickr Photos”, and “Google Maps”. However, these widgets can also be found in the widget catalogue, where they are categorized in different categories. The great benefit of this small presented list is that if the user decides for one of the widgets, it is automatically added to the enterprise mashups. If the user had decided to add the widget from the widget catalogue, he then had the whole responsibility in connecting the widget ports which can also be a nightmare according to the number of the technically focused ports. Instead, if the user decides to add a widget from the lightweight context browsing service presented above, the widget is added to the editor and afterwards connected automatically. This

saves time dramatically and assists the end user in a passive manner without dealing of finding and connecting widgets.

The sales manager decided to add the “Google Maps” widget. With one click, the widget is automatically added. Also the output of the customer data widget is automatically connected to the inport of the “Google Maps” widget. In this case, the “Google Maps” widget is sourced with the address from a customer. As a result, the customers address is automatically displayed on “Google Maps”.

## 6 Conclusion and Outlook

The aim of this paper was the design of a model for context-aware enterprise mashups. After defining the main terms related to enterprise mashups and context, we presented a context domain model for enterprise mashups and a layer concept of how the context can be organized. An implementation of the concept in the SAP Research RoofTop Marketplace prototype demonstrated its applicability. By means of a typical business scenario, we evaluated and discussed the first benefits of the solution.

What is still missing is a broader application of the context model in various application domains. Further research will deal with a user evaluation in a field study to measure the benefits. In addition, there is no privacy enforcement in the current concept. The base of the described concept is the mass number of enterprise mashups which has been saved by the users and later processed. However, while saving enterprise mashups the end users agree that their sensitive data is stored and later be used to support others.

## References

- [1] Benkler, Y.: *The Wealth of Networks. How Social Production Transforms Markets and Freedom*. Yale University Press, New Haven and London (2006)
- [2] Yu, S.: *Innovation in the Programmable Web: Characterizing the Mashup Ecosystem*. In: *Proceedings of the 2nd International Workshop on Web APIs and Service Mashups*, Sydney, Australia (2008)
- [3] Sun, J.-Z., Sauvola, J.: *Towards a conceptual model for context-aware adaptive services*. In: *Proceedings of 4th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Chengdu, China, pp. 90–41 (2003)
- [4] Hoyer, V., Stanoesvka-Slabeva, K.: *Towards a Reference Model for grassroots Enterprise Mashup Environments*. In: *Proceedings of the 17th European Conference on Information Systems*, Verona, Italy (2009)
- [5] Schilit, B.N., Theimer, M.M.: *Disseminating Active Map Information to Mobile Hosts*. *IEEE Network* 8(5), 22–32 (1994)
- [6] Ryan, N., Pascoe, J., Morse, D.: *Enhanced Reality Fieldwork: the Context Aware Archaeological Assistant*. In: *Archaeology in the Age of the Internet: CAA 1997: Computer Applications and Quantitative Methods in Archaeology: Proceedings of the 25th Anniversary Conference*. University of Birmingham (April 1997); *British Archaeological Reports*.

- [7] Schmidt, A., Beigl, M., Gellersen, H.: There is more to Context than Location. *Computers & Graphics* 23(6), 893–901 (1999)
- [8] Dey, A., Abowd, G.: Towards a better understanding of context and context-awareness. Technical report, GIT-GVU-99 22 (1999)
- [9] Henriksen, K., Indulska, J.: A Software Engineering Framework for Context-Aware Pervasive Computing. In: *Proc. of the 2nd IEEE Conference on Pervasive Computing and Communications (Percom 2004)*, Orlando USA, pp. 67–77 (2004)
- [10] Hull, R., Neaves, P., Bedford-Roberts, J.: Towards situated computing. In: *First International Symposium on Wearable Computers 1997, Digest of Papers*, pp. 146–153 (1999)

# User-Centric Composition of Service Front-Ends at the Presentation Layer

Tobias Nestler, Lars Dannecker, and Andreas Pursche

SAP Research Center Dresden  
Chemnitz Str. 48, 01187 Dresden, Germany  
{tobias.nestler,lars.dannecker,andreas.pursche}@sap.com

**Abstract.** The emerge of web services in Service-Oriented Architectures (SOA) within companies or at the global internet offers new ways for the creation of web applications. Even though the composition of services via business processes are covered by existing tools and solutions, concepts for a lightweight service consumption are still in a preliminary phase. The complexity of state-of-the-art SOA technology prevents users with limited IT skills getting easy access to web services and their offered functionalities. This paper presents a user-centric design approach to model and create simple service-based applications in a graphical way without being necessary to write any code.

**Keywords:** Service Composition at the Presentation Layer, UI Integration, Service Front-ends.

## 1 Introduction

The introduction of Web 2.0 offers users the capability to take part in the development of the WWW. Non-technical users are able to create web pages in form of blogs or customize web pages such as iGoogle [1] to serve their daily needs. The next steps towards a user centric design of web applications are mashups that combine the philosophy of SOA and approaches of end user development [2]. This approach is dedicated to data aggregation and highly relies on computing knowledge and skills of the end user. The graphical composition style facilitates user empowerment by aggregating web feeds, web pages and web services from different sources using special builders. However, these existing approaches lack several concepts to support a real end user driven application development [3].

The concepts presented in this paper follow the approach of service composition at the presentation layer enhanced by user interface (UI) related service annotations [4]. We adopt the idea of integration at the presentation layer [5] to compose services by combining their presentation front-ends, rather than their application logic or data [6]. A design-time authoring tool, the ServFace Builder, utilizes the mentioned concepts and aims to empower even non-programmers to create their own service-based application. Following our preliminary investigations, this paper discusses the following contributions:

- We propose a mechanism to visualize the service front-ends already during the design time of the application development. An inference mechanism uses the information gained from the original service description and the attached annotations to create a UI for each service operation. In addition, also common usability recommendations were formalized and integrated to ensure the generation of usable UIs (see Sec. 3).
- We propose a graphical lightweight service composition approach in order to model data flows and control flows at the UI level. The modeling of data flows will be done by the graphical connection of single UI elements of the front-ends. In addition, the paper presents two ways of defining a control flow to create multi-page applications (see Sec. 4).
- We integrate the mentioned concepts into our authoring environment and evaluate their usability and acceptance in form of a user study (see Sec. 5).

## 2 Service Composition at the Presentation Layer

The general approach of service composition at the presentation layer aims to support non-programmers in the design and creation of simple service-based applications. The target user group refers to end users in general and to knowledge workers and skilled web users in particular. The composition process is fully integrated in a three step development methodology presented in 7.

The composition is based on annotated services that act as the foundation for the designed applications. The annotations are reusable information fragments attached to the service description (like WSDL or WADL), which are typically not available for an application developer or service composer. They are created by the service developer and stored in an annotation model based on a formally defined meta-model. Annotations provide extensive additional information covering the visual appearance of a service, the behavior of UI-elements and relations between services to further improve the visual appearance of resulting composite applications. The following examples explain the use of annotations:

- **Annotations defining visual appearance:**
  - **Visual Property:** This annotation adds information about the designated visual appearance of a service element; e.g. if a parameter is a string that represents a password, this parameter should be displayed obscured.
  - **Unit and Conversion:** This annotation assigns a unit to a specific service parameter. In addition, with a set of conversion rules it is possible to offer one value in different units and convert between them.
- **Annotations defining the behavior of UI-elements:**
  - **Suggestion:** This annotation assigns an external suggestion service to a specific service element to provide a list of suggested values while typing.
  - **Validation:** To proof the correctness of entered data, this annotation defines rules entered data is checked up on.
- **Annotations defining the relations between services**
  - **Bundle:** A set of disjunct services or service operations that beneficially work together, can be defined with this annotation.



In general, the 21 currently available annotations should thus be seen as a kind of knowledge transfer from the service developer to the service composer to facilitate the understanding and simplify the composition of web services.

During the design time the information gained from the service description and the annotations are used to create a service front-end, which is a UI for a specific service operation (detailed description in Sec. 3). The user, in his role as a service composer and application designer, interacts with these front-ends only to create the desired application in a kind of WYSIWYG (What you see is what you get) principle. No technical knowledge about service composition is required to build an application, so the user can model and layout his application in a graphical way without being necessary to write any code. An application consists of several pages that can be connected with each other to define a navigation flow through the final application. Each page acts as a container for the front-ends and represents a dialog visible on the screen. The complete design process including the integration and composition of the front-ends (detailed description in Sec. 4) is supported by a web-based authoring tool, the ServFace Builder. Figure 1 shows a screenshot of the current prototype. The ServFace Builder supports the user in the design of multi-page applications and the composition of the service front-ends. An internal object model represents the current modeling state and supports the generation of executable applications for different target platforms. This application model is constantly updated according to user changes and its serialization serves as input to the model-to-code generation process.

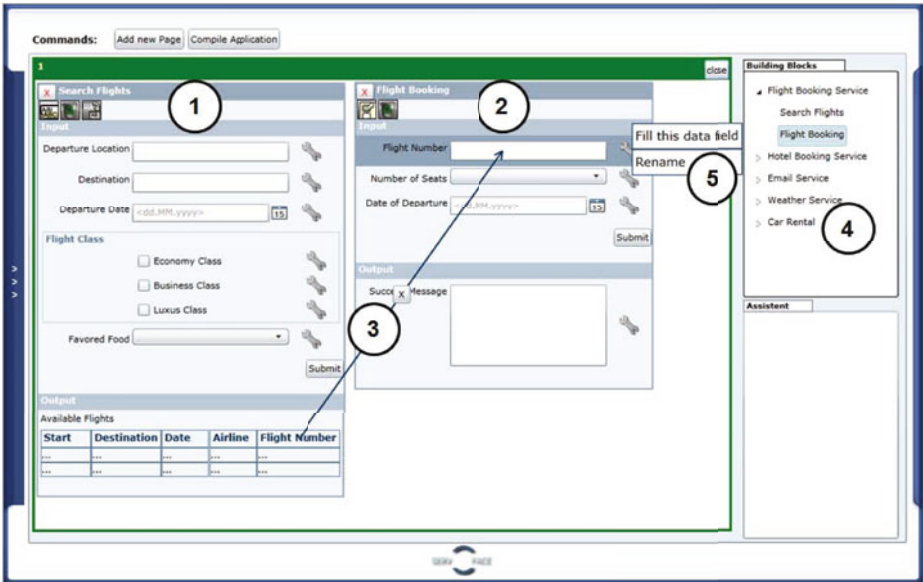


Fig. 1. Screenshot of the ServFace Builder

### 3 Generating Service-Front-Ends

As mentioned above, services and in particular the operations of those services are represented by generated service front-ends. Number 1 and 2 of figure 1 are examples of service front-ends for service operations from a service called "flight booking service". A service front-end representing a service operation comprises of a nested container structure including a root operation container and an interaction container for the input and output parameters of a web service. Those interaction containers comprehend the visualization of the operation parameters that are the key elements to later invoke the service when using the final composite application. Figure 2 illustrates the five step generation process:

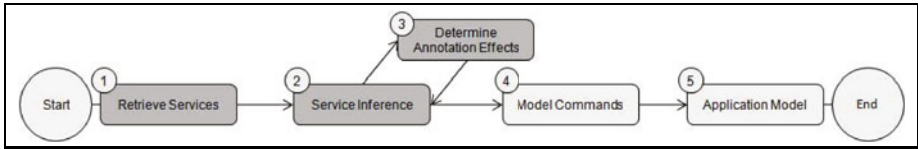


Fig. 2. Generation process for service front-ends

#### 1. Retrieve service elements and structure

Since a remote service repository manages the services, all available information about a service need to be retrieved from the repository at first. To avoid extensively communication affords the information about a service are stored in a local data model. This model represents all information necessary for the front-end generation process. All available services including their operations are presented to the user within the *Building Blocks* browser (Fig. 1 No. 4).

#### 2. Analysis of the services

To infer the structure and the elements of a service operation, an *Inference Engine* parses the model that represents the service and analyses the particular service operation. It generates UI elements for each service element that needs to be displayed within this service-front-end. To infer which UI element should be used for a specific service element the inference engine considers the following:

- parameter type (input or output)
- base data type
- data type enhancements and restrictions
- occurrence definitions

Complex data types need special consideration because they can be arbitrarily nested or can contain recursions. The inference engine does not follow recursions arbitrarily to avoid infinite loops, but rather aborts further inspection after a certain number of loops. For input parameters complex data types are represented by a nested structure, dividing the complex type into its child elements. For output parameters complex data types are presented by tables as standard

output format or lists. Depending on the depth of a nested complex type, the table could contain an expand mode to show complex child elements. Minimum and maximum occurrences are taken into consideration by redundantly displaying the data type element or allowing multi-selection in the range of the occurrence. As a result the inference engine creates model commands (4) that are executed by the API of the underlying application model. The execution affects the current state of the application model (5), e.g. by including new elements or changing existing ones. The following example explains this process:

A simple flight search operation needs two inputs to search available flights. A destination (data type: string) and a departure date (data type: date). The return value is a complex type containing the data type elements "FlightNo.", "Airline", "Price" that all are defined as string data types. Firstly, the inference engine creates the container structure including an operation root group, an input group and an output group. Secondly, both input parameters are inferred. Because the destination parameter is defined as string a textbox is chosen, whereas the departure date is a date type and therefore it is visualized by a calendar widget. Thirdly, for the output parameter as a complex type a structured output e.g. a table is chosen to display the results. Finally, an execution button is created to invoke the web service in the resulting composite application.

### 3. Fetching additional effects

The particularity of this generation process is the consideration of additional information provided by the ServFace service annotations. Furthermore, a consolidation of several HCI guidelines (e.g. provided by Apple or Microsoft) or perceptions of usability experts like [8]) result in a set of formalized UI design recommendations, which improve the service front-end visualization. Whenever the inference engine detects annotations attached upon a service element it requests a description for each annotation, which specifies the effects of this particular annotation. Possible effects include e.g. the creation of additional UI elements, the addition/change of UI element properties, the change of the appearance of UI elements, the restriction to a set of valid values, etc. The annotation effect determination component of the inference engine creates a description object for the annotation. The inference engine analyses the description and includes the effects of the annotation in its analysis process.

A continuation of the flight search example above explains this process:

The parameter of the destination input field is defined as string. Without the influence of annotations it is displayed as a standard textbox handling arbitrary phrases. The addition of an *Enumeration* annotation defines a collection of valid values e.g. a selection of all available destination airports. To avoid false inputs the inference engine would not choose a textbox as representing UI element, but rather a combobox to restrain the input possibilities. In addition a *Feedback* annotation is added to all parameters defining a name readable for humans. This causes the inference engine to not determine the parameter name from the service description that does not necessarily provide an understandable name for its elements, but to use the name provided by the *Feedback* annotation.

The service inference engine as well as the annotation effect determination take the formerly mentioned UI design recommendations into consideration. These recommendations influence the choice which UI element is used to represent a service parameter, or change the configuration of those UI elements. The following example lists the consolidated UI recommendations for the *Enumeration* annotation:

- Use a Radio-Button-Group whenever there is a choice between 2 or 3 values.
- Use a Drop-Down-List whenever there are more than three but under 20 valid values.
- Use a ComboBox (Drop-Down-List with search capability) whenever there are more than 20 but under 200 valid values.
- Use a Textbox when there are more than 200 valid values.
- Show at least 8 items at the same time before a scrollbar appears whenever a Drop-Down-List or ComboBox is used.
- Sort the items alphabetically whenever there are more than 12 items.

The generation of UIs for web services were also analyzed in former research projects like WSGUI [9] or Dynvoker [10]. Both approaches also use additional information to enhance the visualization result, in particular GUIDD [11]. Unlike these approaches the presented concept not only focuses on the generation of UIs to directly invoke services at run time, but to use those UIs already at design time to allow the composition of several services (description in Sec. 4). Furthermore, the used ServFace service annotations exceed the expressiveness of the GUIDD-annotations and a consolidation of recommendations from several end user design guidelines is used to further improve the ease of use of the generated front-ends.

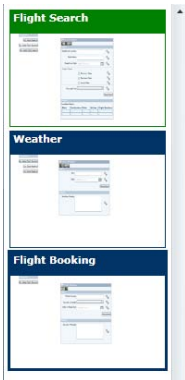
## 4 UI-Centric Composition via Service Front-Ends

Nowadays, there are already many mashup-editors available like Yahoo Pipes [12] or IBM Mashup Center [13] (detailed overview provided by [14]), where the majority focuses on the processing of data. A quite new area for mashups is the definition of simple (business) processes as discussed in [15]. This kind of mashup enables the user not only to aggregate data from different sources but also to combine the invocation of several service operations. Following the previously introduced example, the user is now not only able to search for flights but rather can create an application to book them. To compose such a service-based application, the user needs to define the order of execution of the used service operations and the data flow among the operations.

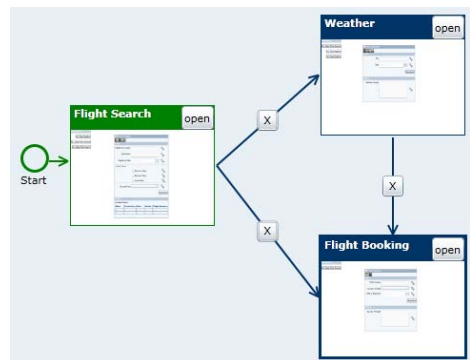
The common visualizations of data flows in mashups are ports (often displayed by little circles) for each variable and arrows from the source to the target port. The activity of connecting those ports is called "Wiring" [14]. In contrast to this approach, the main idea of our UI-centric composition is to directly use the input and output UI elements of the generated service front-ends. To model a data flow between two different front-ends, the user has to click on the target UI element that should be filled with the data (see context menu in Fig. 1 No.5).

To define the source element he directly clicks on the UI element that provides the designated data. Finally, the tool presents the connection in a visual way (see Fig. 1 No.3). The selection of the connection point via UI elements leads to a large target area for the mouse cursor and makes it easy to hit (according to Fitts' Law [16]). The direct use of UI elements for the composition benefits from the fact, that form-based applications are well-known and most users are familiar with them. Therefore, the barrier to use the editor is lower than in tools, which require additional technical concepts like ports. In addition, we assume that it is easier for an end user to handle the definition of data flows within the presentation layer.

For the definition of the order of execution we have two views offering different levels of expressiveness. Both use pages, which can contain front-ends for service operations. The transition from one page to another will cause the invocation of the corresponding web services. The first approach is intended for users who are new to the tool and want to create applications simply and quickly. Therefore, thumbnails of the pages are listed (Fig. 3 a) similar to the well-known slide listing in Microsoft Powerpoint. The order of execution is defined by the order of the pages in the list view. For example moving a page B before another page A, means that the service operations associated with page B are executed before the ones of page A. With this approach you can only define a sequential order without branches in the execution flow, but this is suitable for many simple processes and easy to use. The user has not to consider the details of the process flow. For an experienced user who wants to define more complex processes we provide a detailed flow view showing the transitions between the pages (Fig. 3 b). In this view he is able to create his own transitions and to define alternative flows for an application. We do not only use nodes in this view as it is known from process diagrams, but also the thumbnails of the pages. The user



(a) Sequential Page View



(b) Page Flow View

Fig. 3. Different Page Views

can click on a thumbnail and open it, to see the service front-ends contained by this page, integrate new ones or connect UI elements as described above.

## 5 Evaluation

To evaluate the presented approaches a user study was conducted. While for the first study [17] only some mockups and a UI prototype were available, a fully functional prototype could be used in course of this evaluation. The goals of the evaluation were to evaluate the acceptance of the concepts, the ease of use of the tool, the necessary period of vocational adjustment and other aspects regarding the usability of the ServFace Builder.

Since the aspired end user should not necessarily have advanced IT-experience, we ensured that the background knowledge of all participants was not related to computer science. The main scope of the evaluation was divided into two parts. The first part was an exploratory evaluation of the ServFace Builder with a duration of approx. 15 minutes. The participants had the chance to try out the tool without further tasks or guidance. They have not received extensive information beforehand either. However the observers stood by to assist or answer questions whenever necessary. As shown in figure 4 most users stated that the bench of the ServFace Builder is very clearly structured (92%) and the functionality is self-explanatory (67%). This means that the usage of the tool should be easily understandable even without a previous tutorial or extensive explanations. However many user (67%) had problems getting started right after they reached the first tool screen. This leaves room for further optimizations regarding the start-up process and the user guidance coordinating the first steps.

In the second part of the evaluation the users got the task to adopt the role of a company worker who tries to ease the business travel booking process. They had to create a composition that satisfies the requirements of the given scenario description. A selection of the results of the second part is presented in figure 5. As expected, the participants had less trouble to get started with the composition process now (92%). Both, the participants and the observers had the impression that in most cases the design process came more naturally to the participants. This impression was confirmed by the comments of the participants: 92% stated that the design process of a multi-page composite application is straightforward and all participants declared that the front-ends are self-explanatory (75%) or

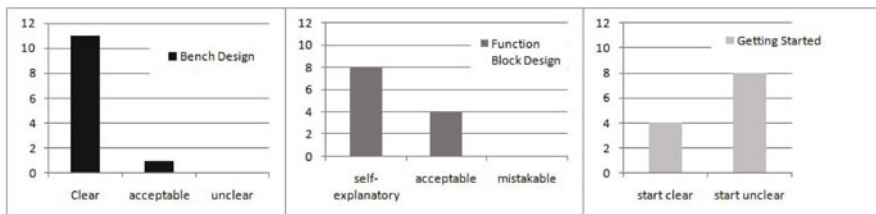


Fig. 4. Evaluation results part one

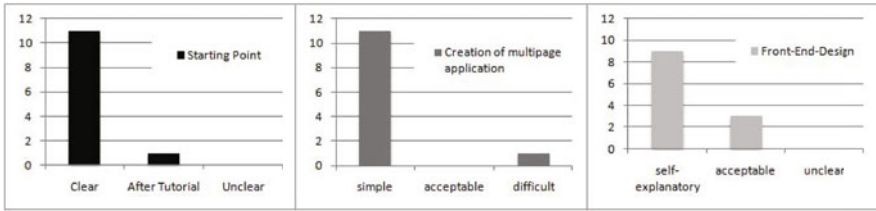


Fig. 5. Evaluation results part two

at least understandable (25%). This points out that the users of the tool are able to handle the composition process quite well with little familiarization. The 15 minutes exploratory try-out was sufficient to enable the participants to design the required application without further difficulty. Therefore the period of vocational adjustment is rather short even for users without IT-background. In contrast many users (67%) criticized the lack of guidance provided by the ServFace Builder. In addition, the Drag&Drop concept to add the service operations to the canvas was rated neither intuitive nor comfortable.

The flow view (Fig. 3 b) was used as the central view showing the pages during the user study. But only 25% of the participants asked for the possibility to influence the order or to define alternative flows. The observers noticed that most participants just added new pages and integrated service front-ends in the linear order they had created the pages before. This motivates the modification of the ServFace Builder, the simple sequential view should be the standard view on start-up and the flow view should be available on demand.

## 6 Conclusion and Future Work

The paper presented two concepts supporting the approach of service composition at the presentation layer. The generation and composition of service front-ends offers new ways in designing service-based applications for non-programmers. An evaluation with a working prototype showed that most users handled the tool without extensive difficulties after a short familiarization period. Most concepts pointed out to be understood and accepted by the users. However, there are further fields of improvement. An extended guidance and a better way to get the composition process started are two important examples. In addition, the visualization process of the service front-ends could be enhanced by an extended support for different platforms. An expert-based evaluation as well as a third iteration of the user study are planned to estimate the recommended changes gained from the presented evaluation.

## Acknowledgment

This work is supported by the EU Research Project (FP7) ServFace. In addition, we would like to thank the participants of the user study and the supervisors of our theses Marius Feldmann (L. Dannecker) and Gerald Hübsch (A. Pursche).

## References

1. iGoogle (2009), <http://www.google.de/ig?hl=de&source=iglk>
2. Hoyer, V., Stanoevska-Slabeva, K.: The Changing Role of IT Departments in Enterprise Mashup Environments. In: 2nd International Workshop on "Web APIs and Services Mashups" (2008)
3. Nestler, T.: Towards a Mashup-driven End-User Programming of SOA-based Applications. In: 10th International Conference on Information Integration and Web-based Applications & Services (2008)
4. Nestler, T., Feldmann, M., Preussner, A., Schill, A.: Service Composition at the Presentation Layer using Web Service Annotations. In: ComposableWeb 2009 Workshop at ICWE 2009 (2009)
5. Yu, J., Benatallah, B., Saint-Paul, R., Casati, F., Daniel, F., Matera, M.: A Framework for Rapid Integration of Presentation Components. In: WWW 2007, Banff, Canada (May 2007)
6. Daniel, F., Yu, J., Benatallah, B., Casati, F., Matera, M., Saint-Paul, R.: Understanding UI Integration: A survey of problems, technologies, and opportunities. In: IEEE Internet Computing (May/June 2007)
7. Feldmann, M., Janeiro, J., Nestler, T., Hübsch, G., Jugel, U., Preussner, A., Schill, A.: An Integrated Approach for Creating Service-Based Interactive Applications. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5727, pp. 896–899. Springer, Heidelberg (2009)
8. Krug, S.: Don't Make Me Think! A Common Sense Approach to Web Usability, vol. 2. New Riders, Indianapolis (2006)
9. Kassoff, M., Kato, D., Mohsin, W.: Creating GUIs for Web services. In: IEEE Internet Computing, CA, USA, vol. 7 (2003)
10. Spillner, J., Feldmann, M., Braun, I., Springer, T., Schill, A.: Ad-hoc Usage of Web Services with Dynvoker. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave 2008. LNCS, vol. 5377, pp. 208–219. Springer, Heidelberg (2008)
11. Kassoff, M., Spillner, J.: GUIDD: Standard and Specification (2006)
12. Yahoo! Pipes (2009), <http://pipes.yahoo.com>
13. IBM Mashup Center (2009), <http://ibm.com/software/info/mashup-center>
14. Hoyer, V., Fischer, M.: Market Overview of Enterprise Mashup Tools. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 708–721. Springer, Heidelberg (2008)
15. de Vrieze, P., Xu, L., Bouguettayay, A., Yangz, J., Chenx, J.: Process-oriented enterprise mashups. In: Grid and Pervasive Computing Conference (2009)
16. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* (1954)
17. Namoune, A., Nestler, T., Angeli, A.D.: End User Development of Service-based Applications. In: 2nd Workshop on HCI and Services at HCI 2009 (2009)



# On the Support of Dynamic Service Composition at Runtime

Eduardo Silva, Luís Ferreira Pires, and Marten van Sinderen

Centre for Telematics and Information Technology  
University of Twente, The Netherlands  
P.O. Box 217, 7500 AE Enschede

{e.m.g.silva,l.ferreirapires,m.j.vansinderen}@cs.utwente.nl

**Abstract.** Network-based software application services are receiving a lot of attention in recent years, as observed in developments as *Internet of Services*, *Software as a Service* and *Cloud Computing*. A service-oriented computing ecosystem is being created where the end-user is having an increasingly more active role in the service creation process. However, supporting end-users in the creation of a service, at runtime, is a difficult undertaking. Users are heterogeneous, have different requirements, preferences and knowledge. Furthermore, and since we cannot assume that all users are technical experts, we conclude that highly abstract mechanisms should be available to support the service creation process. To tackle these issues and provide end-users with personalised service delivery, we claim that runtime automated service composition mechanisms are required. In this paper we present the DynamiCoS framework, which aims at supporting the different phases required to provide users with automatic service discovery, selection and composition process. In this paper we also present the developed prototype and its evaluation.

## 1 Introduction

With the Internet becoming ubiquitous, the use of network-based application services is being increasingly adopted and it is expected to grow in the upcoming years [1]. This is being reflected in many technology developments and innovations, such as, for example, the *Internet of Services*, *Cloud Computing* and *Software as a Service* (SaaS). This is leading to the emergence of large sets of services in different domains. At the same time, the use of mobile devices with powerful communications capabilities is increasing quite rapidly. In [2] it is reported that by 2013 more than 38% of the European population will access the Internet on their mobile device, which is an increase of 300% compared to the current situation.

These developments are allowing and *pushing* new, more adaptive and personalised, application services where the end-users play an active role in the service creation process. Supporting end-users in this kind of process is a complex undertaking. Different users have different preferences and request services in different situations (contexts), which requires different actions to be taken. Furthermore, end-users expect a high-level of abstraction in the service creation process, since they are not properly trained to use complex (technical) tools. Given this, some degree of automation has to be provided to support the end-user in the service creation process. We claim that this can be

achieved by using semantic-based service composition approaches. Semantic information enables the use of computer agents, which can automatically reason on the services and user specified requirements. This alleviates the user from the burden of some of the details and the complexity of the service composition process. We denote the problem of automatic service composition based on user requirements as *dynamic service composition*. To cope with this problem, we propose a framework for dynamic service composition provisioning called *DynamiCoS*.

The aim of the DynamiCoS framework is to provide all the necessary support to users, namely to achieve automated runtime service composition. To achieve this automated support, DynamiCoS uses ontologies (domain conceptualisations). The framework allows different service developers to publish their semantically annotated services in the framework. These semantic descriptions have to refer to the framework's ontologies. Users may have different domain or technical knowledge, which implies that their service request interfaces have to be defined accordingly. DynamiCoS tackles this problem by prescribing a service request that supports different user interfaces. A service request consists of a specification of *goals* the user wants the service to achieve. A *goal* is likewise used to describe services, specifying the activities (or operations) the services can perform. Goals of users and services are specified according to the framework ontologies (representation of the domain of knowledge), which allows matching services to be found, whenever these services realise the user goals.

This paper is further organised as follows: Section 2 characterises different types of users based on their knowledge; Section 3 presents DynamiCoS, a framework to support users in the composition process; Section 4 provides an overview of related work; and Section 5 provides our conclusions and directions for future work.

## 2 Users' Knowledge

An user denotes a person that makes use of some functionality of a system. In the context of our work we consider that an user is a person with limited technical skills. Users may have different characteristics according to their knowledge of the composition process. In this work we show that not all users share the same characteristics. We present a classification of users according to their knowledge in the domain of the service being composed and their technical knowledge on the tooling supporting the service composition process. An user may play two roles in our context, namely to create or execute service compositions. A certain user may play both roles.

### 2.1 Domain Knowledge

In the process of service composition users need to have some knowledge or idea of the service they want, i.e., what the desired service does, who provides it, etc. We refer to this knowledge as domain knowledge. Domain knowledge is obtained by learning, advertisement, interaction with the service providers, etc. Users may have this type of knowledge when they want to use a service in a given application domain, but often they have limited knowledge and require some interaction with the elements of the application domain(s) to acquire the knowledge necessary to decide on the service to be requested.

In semantic services, the domain knowledge is explicitly represented in ontologies, which are domain conceptualisations produced by domain experts. This information can then be used to describe the services in the domain, find these services and reason on them.

For example, if a user wants to buy a phone online, he usually has an idea of what type of phone and the market, i.e., phone type/brands, price, stores that sell phones, etc. The application domain has the central concept of telephone store, to which resources (phones, phone stores, etc.) and actions the user can take (search phone, buy phone, etc) are stated. At the end, the user knowledge of the domain will allow him to decide what decision to take, and possibly to compose actions (or services) to realise the desired service.

## 2.2 Technical Knowledge

Service composition is being used in different domains and applications nowadays, mainly by professional users or developers, which can handle complex tooling and understand the composition process, and details associated with this process. For example, many companies nowadays use Web services technology, and apply WSDL (Web Service Description Language) to describe services in the company, and BPEL (Business Process Execution Language) to compose and coordinate services. However, end-users are not expected to know these technical details. A service composition environment for end-users has to provide a higher level of abstraction to their users, so that users without technical background can make use of this environment.

For example, if the end-user wants to find a phone and then buy it, this whole operation may consist of two services (find and buy services). The supporting tooling has to allow the end-user to find suitable services and then help him with the composition process, by automating this process or by suggesting possible next actions to the users. The supporting tooling may depend on the type of application domain. For example, if the application domain is e-health, where a caregiver decides the sequence of activities a given patient has to perform (e.g., measure blood pressure and if it is too high send a message to the patient's doctor), the caregiver has knowledge on the domain, i.e., he knows the services available in the domain. On the other hand, if an end-user wants to buy a phone, the user may not know the domain, i.e., the supporting environment has to deliver the necessary suggestions and guide the user towards the creation of the service (composition) he wants.

## 2.3 Types of User

Table 1 defines a possible classification of types of users, based on their domain and technical knowledge.

A *Layman* is a user who does not know the application domain in enough detail, neither has knowledge on the tooling supporting the composition process. A *Domain Expert* is a user who knows the application domain, but does not have knowledge on the technical tooling that supports the service composition process. A *Technical Expert* is a user who has knowledge on a service composition tool, but does not know the details of the application domain. An *Advanced* user is a user who has technical knowledge

**Table 1.** Types of Users

Type of User	Domain Knowledge	Technical Knowledge
<i>Layman</i>	No	No
<i>Domain Expert</i>	Yes	No
<i>Technical Expert</i>	No	Yes
<i>Advanced</i>	Yes	Yes

on the tooling supporting the composition process, and furthermore knows the application domain. Because this classification depends on the domain being considered, and sometimes the user uses services from several domains, a user can be, sometimes, of more than one of the types identified. Furthermore, the user may also not have a direct mapping to one of the identified user types, there may exist other types. Our intention is not to identify all the possible user types that may exist. We use the identified user types to motivate that there may exist users with different requirements and characteristics. Based on this we believe that supporting environments should be created according to the target population of users they have.

At the moment, DynamiCoS addresses mainly the users that have some knowledge about the application domain, i.e., *Domain Experts* and *Advanced* users. However, we are working in a methodology that will make DynamiCoS adaptable to different types of users, with different types of knowledge.

### 3 DynamiCoS Framework

*DynamiCoS*<sup>1</sup> (**D**ynamic **C**omposition of **S**ervices) supports the different phases and stakeholders of the dynamic service composition life-cycle, as discussed in our previous work [3]. Fig. 1 shows the DynamiCoS framework architecture. It also indicates (between parenthesis) the technologies used in the implementation of the different framework components in our prototype platform.

Given the complexity of the dynamic service composition life-cycle and its stakeholders' heterogeneity, we have made the core components of the framework technology-independent. In the framework, a service and a service composition are represented as a tuple and a graph, respectively. A service is represented as a seven-tuple  $s = \langle ID, I, O, P, E, G, NF \rangle$ , where  $ID$  is the service identifier,  $I$  is the set of service inputs,  $O$  is the set of service outputs,  $P$  is the set of service preconditions,  $E$  is the set of service effects,  $G$  is the set of goals the service realises,  $NF$  is the set of service non-functional properties and constraint values. We assume in this work that services are stateless *request-response*, i.e., they consist of a single activity or operation. A service composition is represented as a directed graph  $G = (N, E)$ . Graph nodes  $N$  represent services, i.e., each node  $n^i \in N$  represents a discovered service  $s^i$ . A node can have multiple ingoing and outgoing edges. Each graph edge  $E$  represents the coupling between a service output/effect (or a user input for the services) and a service input/precondition, i.e.,  $e_{i \rightarrow j} = n_{O/E}^i \rightarrow n_{I/P}^j$ , where  $i \neq j$  (a service cannot be coupled with itself).

<sup>1</sup> <http://dynamicos.sourceforge.net>

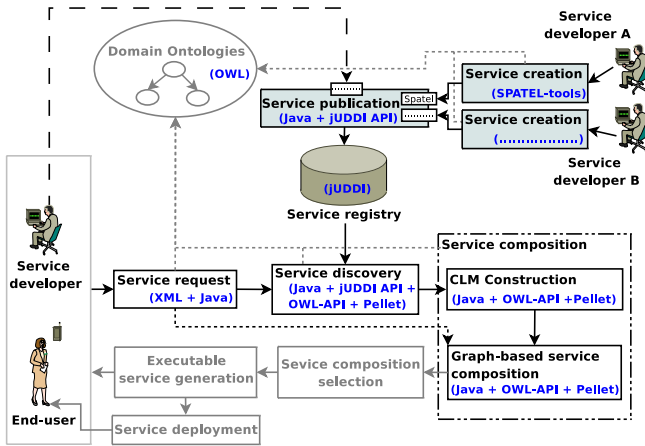


Fig. 1. DynamiCoS framework

### 3.1 DynamiCoS Modules

DynamiCoS consists of the following modules:

**Service Creation.** The creation of a new service “from scratch” by a service developer is performed outside the DynamiCoS framework. However, to comply with the capabilities of the DynamiCoS framework, the services have to be semantically described, in terms of inputs, outputs, preconditions, effects (*IOPE*), goals (*G*) and non-functional properties (*NF*), using the framework domain ontologies’ semantic concepts.

**Service Publication.** To support the publication of services described in different languages, the DynamiCoS framework has a two-step publication mechanism. First, there should be an interpreter for each supported service description language. The interpreter reads the service description document and extracts the necessary information for publication (*ID, I, O, P, E, G, NF*). This makes the service representation in the framework *language-independent*. Second, the extracted service information is published in the service registry using the DynamiCoS generic service publication mechanism. The service registry allows one to publish, store and discover semantic services.

**Service Request.** The user interface to define the service request may have different forms, as long as it gathers the user goals and the expected outputs/effects. Optionally it can also gather the inputs/preconditions and non-functional properties that the service should support. The number of parameters defined in a service request may depend on the type of end-user expertise. If the end-user has technical knowledge and domain knowledge, all the parameters (*G, IOPE, NF*) may be used. However, if the end-user has no technical knowledge, a simpler request may be created, for example, only based on the goals the user wants to achieve and the expected service outputs. The service request parameters are defined as references to semantic concepts available in the framework ontologies. Therefore, the service request consists of a set of semantic annotations (*I, O, P, E, G, NF*) that describe *declaratively* the desired service properties.

**Service Discovery.** The discovery of candidate component services is performed before the service composition phase. Service discovery is based on the service request parameters. The service discovery process consists of querying the service registry for all the services that *semantically* match the defined service request parameters. Since DynamiCoS uses a semantics-based service discovery and composition, it discovers not only exact matches with the service request  $G$  and  $IOPE$  semantic concepts, but also other services with partial semantic matches, e.g., services with parameters that are semantically subsumed by the service request parameter concepts, i.e.,  $RequestedConcept \sqsupseteq DiscoveredConcept$ .

**Service Composition.** To perform service composition, DynamiCoS first processes the set of discovered services and organises them in a so called *Causal Link Matrix (CLM)* [4]. The CLM stores all possible semantic connections, or *causal links*, between the discovered services input and output concepts. CLM rows (Equation 1) represent the discovered services input concepts ( $DiscServs_I$ ). CLM columns (Equation 2) represent service inputs concepts plus requested service outputs ( $ServReq_O$ ).

$$CLM_{rows} = DiscServs_I \quad (1)$$

$$CLM_{colu} = DiscServs_I + ServReq_O \setminus (DiscServs_I \cap ServReq_O) \quad (2)$$

We place a service  $s$  in the row  $i$  and column  $j$  position if the service has an input semantically related with the input on column  $i$  of the CLM and an output semantically related with the semantic concept on column  $j$ . Furthermore, we store in the matrix the *semantic similarity* of the service output  $i$  and the column semantic concept  $i$ , and the non-functional properties of the service.

Given a CLM, the composition algorithm has to find a composition of services that fulfils the service request. Our service composition algorithm is graph-based. Algorithm 1 presents the algorithm in pseudo code. The composition process starts by analysing the CLM matrix, checking if it contains the requested IOPE. After this, the CLM is inspected for services that provide the service request outputs. If there are services that provide the service request outputs, the algorithm starts by creating the initial matching nodes, otherwise it stops. If the service request outputs can be provided by the discovered services, the algorithm proceeds with a backwards composition strategy towards the requested service inputs. An *open* (or not yet composed) input of the graph is resolved at each algorithm iteration. The algorithm matches the *open* inputs of the services in the graph with the output concepts of services from the CLM matrix, or column concepts. If multiple services exist that match a given service input, a new composition graph is created, representing an alternative service composition behaviour. The algorithm finishes when all requested inputs, preconditions and goals from all the alternative service compositions are resolved.

### 3.2 Prototype

In our prototype we have used a language for semantic service description called Spatel [5]. Spatel has been developed in the European IST-SPICE project [6], where the DynamiCoS framework was partly developed. The ontologies used in the framework are described in OWL [7]. We used four ontologies: *Goals.owl*, which contains

**Algorithm 1.** Graph Composition Algorithm

---

```

Input:  $CLM, ServReq$ 
Result:  $ValidComps$ 

// Variables
1  $activeG$ ; // Graph that is active in the algorithm iteration
2  $activeN$ ; // Node that is active in the algorithm iteration
3  $openG$ ; // Set of open graphs
4  $validG$ ; // Set of completed and valid graphs
// Initialisation
5 if  $CLM_{rowsUcolu} \supseteq ServReq_{I,O}$  then
  // Create new graph instantiating the initial Node
  6  $activeG \leftarrow createNewGraph(ServReq)$ ;
  7  $createInitialNodes()$ ;
  8  $openG \leftarrow activeG$ ;
9 else
  // Discovered services cannot fulfil the service request
  10 Stop;

// Graph construction cycle
11 while  $|openG| > 0$  do
  // Close graph if it matches  $ServReq_{I,G}$ 
  12 if  $activeG_{I,G} \supseteq ServReq_{I,G}$  then
  13    $validG \leftarrow activeG$ ;
  14    $openG \leftarrow openG \setminus activeG$ ;
  15    $activeG \leftarrow openG^0$ ;
  16    $activeN \leftarrow activeG_{openN^0}$ ;
  17   break; // Goes to next  $openG$ 

  // Checks CLM for services that match open inputs
  18 foreach  $semCon \in activeN_I$  do
  19   if  $CLM_{colu} \supseteq semCon$  then
  20      $activeN \leftarrow CLM_{matchingNode}$ ;
  21   else
  22      $openG \leftarrow openG \setminus activeG$ ;
  23     break; // No possible composition, goes to next open graph

  // Check if graph NF props comply with requested NF props
  24 if  $activeG_{NF} \cap ServReq_{NF} = \emptyset$  then
  25    $openG \leftarrow openG \setminus activeG$ ;
  26   break; // If Not, composition is not possible

  // prepare next cycle
  27  $openN \leftarrow openN \setminus activeN$ ;

```

---

the services' supported goals and also goals that the user can specify in the service request; *NonFunctional.owl*, which defines non-functional properties that can be used to describe services; *Core.owl* and *IOTypes.owl*, which are used to describe services' *IOPE* parameters. However, the framework is general enough to support the use of other ontologies to define other application domains.

For service publication we have implemented a Spatel interpreter. The interpreter imports the Spatel service description by using a Java API generated from the Spatel Ecore model with the Eclipse Modelling Framework (EMF). The service is then published in a UDDI-based service registry that has been extended with semantic support. We use jUDDI [8] as service registry implementation, which is a Java implementation of the UDDI specification for Web services. jUDDI offers an API for publication and discovery of services. We have extended jUDDI with a set of UDDI models (*tModels*) to store the set of semantic annotations ( $I, O, P, E, G, NF$ ) that describe a service in our framework.

For testing we have created two interfaces to specify the service request: a simple Java-based graphical interface, and a web-based interface. Both allow the specification of the different parameters ( $IOPE$ ,  $G$ ,  $NF$ ) of the service request. The information introduced by the end-user is then transformed to an XML-based representation and sent to the composition framework.

Service discovery is performed based on the service request information. The  $IOPE$  and  $G$  annotations are extracted from the service request and the service registry is queried through the jUDDI API Inquiry function. To discover and reason on semantically related concepts/services we use the OWL-API [9] and Pellet [10].

The CLM matrix is constructed by using the OWL-API, which allows one to handle and perform semantic inference on OWL ontologies by using a semantic reasoner, in our case Pellet [10]. The service composition algorithm is implemented in Java.

In the project website (<http://dynamicsos.sourceforge.net>) we provide more information about the framework and prototype.

### 3.3 Evaluation and Validation

In a forthcoming publication we will present details on the evaluation and validation of DynamiCoS. The performed evaluation focuses mainly on performance and feasibility of the dynamic service composition process. We have concluded that the several phases of the service composition process can be automated, by using semantic descriptions. However, semantic reasoning is expensive in terms of processing time. Despite that, we consider that such expensive processing times are acceptable to support end-users in the creation of new service compositions on demand at runtime, since in other situations, mainly manual composition is used to tackle this problem, i.e., the user has to spend normally much more time performing a service composition. Furthermore, we expect manual composition, even with intuitive interfaces, to be too complex for most of the types of end-users.

## 4 Related Work

Dynamic service composition has received a lot of attention lately. We refer to [11] for an overview of some existing approaches. Most of these approaches focus on some phases of the dynamic service composition life-cycle, often the discovery and composition phases. However, some approaches cover most of the phases of the dynamic service composition life-cycle, as, e.g., METEOR-S [12]. METEOR-S provides mechanisms for semantic annotation of existing services, service discovery, and service composition. METEOR-S focuses mainly on design-time creation of service compositions, by developing *templates* that can have dynamic bindings at runtime. Our approach, as many other ones, has been inspired by METEOR-S, but we target an *on demand* runtime service composition creation, to support end-users at runtime. Kona et al. [13] propose an approach oriented to automatic composition of semantic web services. Similarly to DynamiCoS, they propose a graph-based service composition algorithm. The composition process is performed using a *multi-step narrowing algorithm*. The user specifies a service request, or a *query service*, specifying the  $IOPE$  for the desired service. The composition problem is then addressed as a discovery problem.



Recently new approaches have emerged to support specifically end-users in the service composition process. These approaches, for example [14] [15] [16], mainly focus on using techniques for mashup, with intuitive graphical representations that allow end-users to create their own services. We argue that these approaches are applicable if the user of the composition environment has some technical knowledge on the composition environment, has a clear idea of the service he wants and knows the application domain. However, if the end-user does not have a clear idea of the service he wants, but only some vague ideas about the goals that he wants to be fulfilled by the service, an approach similar to the one we are proposing may be more appropriate.

## 5 Final Remarks

In this paper we started by motivating that there are different types of users of service composition systems. Users may have different knowledge of the service composition application domains, and also of the technical tooling supporting the composition process. Based on this observation, we claim that supporting environments have to be created or adapted to match the user and his knowledge and expertise. To support the development of such supporting environments we propose DynamiCoS, which is a framework that supports more dynamic and automatic composition of services. To achieve automation in the composition process, DynamiCoS is based on semantic services. DynamiCoS is neutral with respect to the semantic service description languages used by the service developers. DynamiCoS supports service creation and publication by service developers at design-time, which make services available in the framework; and automatic service composition by end-users at runtime. We have experimented with DynamiCoS, showing that it is capable of providing *real-time* service delivery, and we observed that semantic reasoning is an expensive task in terms of processing time. However, these expenses may be worth paying for, since automated support of users in the creation of their own services can be enabled.

In the future we will investigate how the user properties can be used on the optimisation and personalisation on supporting him in the composition process. We are investigating mechanism to *guide* users through the process of specifying the service composition behaviour. These supporting mechanisms adapt according the user that is being supported. This should enable the support different types of users, namely the ones identified in Section 2.3, specially the ones without domain knowledge nor technical knowledge. This process will require several interactions between the platform and the user, and a dynamic negotiation to match the user interests with a composition created out of the available services. To validate this we will perform some empirical evaluations of the proposed mechanisms using users in some suitable application scenarios.

## References

1. Gartner: Gartner highlights key predictions for it organisations and users in 2008 and beyond (January 2008)
2. Forrester: European mobile forecast: 2008 to 2013 (March 2008)

3. Goncalves da Silva, E., Ferreira Pires, L., van Sinderen, M.J.: Defining and prototyping a life-cycle for dynamic service composition. In: International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing, Portugal, pp. 79–90 (July 2008)
4. Lécué, F., Léger, A.: A formal model for semantic web service composition. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 385–398. Springer, Heidelberg (2006)
5. Almeida, J.P., Baravaglio, A., Belaunde, M., Falcarin, P., Kovacs, E.: Service creation in the SPICE service platform. In: Wireless World Research Forum meeting on “Serving and Managing users in a heterogeneous environment” (November 2006)
6. Cordier, C., Carrez, F., van Kranenburg, H., Licciardi, C., van der Meer, J., Spedalieri, A., Rouzic, J.P.L.: Addressing the challenges of beyond 3G service delivery: the SPICE platform. In: International Workshop on Applications and Services in Wireless Networks (2006)
7. Smith, M.K., McGuinness, D., Volz, R., Welty, C.: Web Ontology Language (OWL) guide, version 1.0. W3C (2002)
8. Apache: Apache juddi, <http://ws.apache.org/juddi/>
9. Bechhofer, S., Volz, R., Lord, P.: Cooking the semantic web with the OWL-API. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 659–675. Springer, Heidelberg (2003)
10. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2), 51–53 (2007)
11. Rao, J., Su, X.: A survey of automated web service composition methods. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005)
12. Verma, K., Gomadam, K., Sheth, A.P., Miller, J.A., Wu, Z.: The meteor-s approach for configuring and executing dynamic web processes. Technical report, University of Georgia (June 2005)
13. Kona, S., Bansal, A., Gupta, G.: Automatic composition of semantic web services. In: International Conference on Web Services, pp. 150–158 (2007)
14. Liu, X., Huang, G., Mei, H.: A user-oriented approach to automated service composition. In: IEEE International Conference on Web Services, ICWS 2008, pp. 773–776 (September 2008)
15. Ro, A., Xia, L.S.Y., Paik, H.Y., Chon, C.H.: Bill organiser portal: A case study on end-user composition. In: Hartmann, S., Zhou, X., Kirchberg, M. (eds.) WISE 2008. LNCS, vol. 5176, pp. 152–161. Springer, Heidelberg (2008)
16. Nestler, T.: Towards a mashup-driven end-user programming of SOA-based applications. In: iiWAS 2008: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, pp. 551–554. ACM, New York (2008)

# Rethinking the Semantic Annotation of Services

Nikolaos Loutas<sup>1,2</sup>, Vassilios Peristeras<sup>1</sup>, and Konstantinos Tarabanis<sup>2</sup>

<sup>1</sup> National University of Ireland, Galway, Digital Enterprise Research Institute  
{firstname.lastname}@deri.org

<sup>2</sup> Information Systems Lab, University of Macedonia, Thessaloniki, Greece  
{nlout, kat}@uom.gr

**Abstract.** This work extends and enhances existing semantic service models by involving users and by including service metadata related to the user's view of the service and their behaviour. We borrow ideas and extend the models and practices for the annotation of Web content and information resources that has recently become popular in widely-used social platforms. Users are encouraged to describe in their own terms the services they use. Our approach strengthens user participation in the Web and more generally in the service industry by providing service metadata, which are later used as a form of lightweight user-side semantic annotation of services. This annotation is provided explicitly by the users and/or implicitly by identifying patterns in the users' behaviour. This type of service annotation acts supplementary to the service descriptions provided by the service providers and is linked to the actual use of the services. Finally, we harvest the collected metadata and use it for facilitating discovery and clustering of services, as well as to enable service recommendations and matchmaking with users' profiles.

**Keywords:** service, service-oriented, semantic, social contract, social metadata.

## 1 Introduction

Service-orientation is currently one of the most popular and most discussed computing paradigms. Service-oriented systems facilitate the (re)use and sharing services from different sources. A typical service-oriented model defines three main entities, i.e. the service provider, the service requestor and the service broker, and three fundamental operations, namely publishing, finding and binding [3].

Semantics have been applied in service-oriented systems as a means to enhance the three-partite service-oriented model. The idea is that semantically described services could enable and facilitate the dynamic discovery, invocation, execution, composition and monitoring of services at run-time [9]. This led to the definition of various service ontologies and Semantic Web Services (SWS) models, such as OWL-S [11], WSMO [17], WSDL-S [13] and followed by SAWSDL [5].

In this paper, we add an additional layer to existing semantic approaches. Apart from the formal service descriptions (semantic or not) that are made available by the service providers in service-oriented systems, we enhance the service descriptions by “capturing” and “attaching” information, which is related to the actual usage of the

services by users in the real world. Our intention is to add a social layer on top of service-oriented systems. We propose two main mechanisms for this:

- To allow users to annotate the services they use.
- To analyze the behavioural service usage patterns of the users', i.e. how the users consume the services, by monitoring their behaviour and their actions.

Afterwards, the information that we collect is used to enrich the existing service descriptions. We call this process *social annotation of services* and the metadata that emerge bottom-up from this process *social metadata*. The social annotation can be supplementary to the semantic service descriptions that are already provided by the service providers.

In order to accommodate the social aspects and characteristics of services, existing semantic service models need to be extended. In this paper, we discuss the required extensions; we introduce an approach for adding social annotations to services and demonstrate how to utilize the extra (social) semantics that emerge in order to facilitate service search, mashing and recommendation.

Before continuing, it is necessary to clarify that our approach focuses mainly on extrovert services which have a business value and are to be used by the end-users.

The remainder of this paper is organized as follows: section 2 presents our motivation. Section 3 discusses in detail the idea of social annotation of services and introduces the notion of the *social contract*. Section 4 shows how an existing semantic service models can be extended in order to include social metadata. Section 5 describes our prototype. Finally, section 6 concludes the paper and discusses our future research directions.

## 2 Motivation

All the semantic service frameworks proposed so far, e.g. WSMO, SAWSDL, OWL-S share a common principle: they assume that the (semantic) description of services comes solely from the service providers. This results into two serious limitations:

- The users are totally left outside of the service description process. These approaches do not take into account the way that the users of the service perceive it. For example, users cannot detail the reason they use a service, e.g. to book a flight, or the context in which the service is used, e.g. a service as part of a more general "travelling" context. Recently, both the research community and industry sensed this shortcoming. There are already attempts which focus on the user's perspective which was left completely out of picture in SOA e.g. [12] and the work conducted in the context of the COIP FP7 ICT IP project<sup>1</sup>.
- Semantic service efforts have still a low adoption rate. To come up with elaborated semantic service descriptions, the service providers have to be convinced about the additional value of these semantics in order to spend resources to annotate their services. However at the moment, it seems rather unlikely to convince service providers to use existing SWS frameworks, and as a result these efforts are not taking off [18, 23].

---

<sup>1</sup> <http://www.coin-ip.eu/>

So far, there has been no real large-scale application of SWS in industry. Among other problems in [24], the author claims that SWS have ill-defined semantics and that service ontologies usually describe the semantics of WSDL interfaces, which are different from the semantics of the WS. As such the existing SWS approaches have not managed to fully support automated discovery, matchmaking, composition, and execution. Generally, the high complexity of the SWS approaches discourages both technical and business people from adopting such solutions. These problems have created pressure to the SWS research community to come up with lightweight approaches, which may lack in expressivity but win in simplicity (e.g. SAWSDL, SA-REST [20], WSMO-Lite [22]). However, it is still early to evaluate the applicability and the adoption of such lightweight semantic service models.

Lately, Web 2.0 is emerging as a new computing paradigm. Web 2.0 preaches for active user participation in the Web through user-centric Web portals and applications [14]. In Web 2.0 there is no clear line of separation between service providers and users as the latter interact with the Web not just as information receivers but also as content providers [4]. In Web 2.0 platforms users add data and metadata: they add content, e.g. photos, multimedia and documents, and then use tags for attaching meaning to this content. Other users also add metadata (tags), which are then used to enable better search and discovery (e.g. [2, 9]), personalization of the user's experience etc.

Moreover, unlike SOA environments, in Web 2.0 semantics (metadata) come mainly from tags and folksonomies and as such emerge in a bottom-up fashion directly from the users (e.g. [7, 21]). We argue that as this user-defined metadata are used for creating richer descriptions for resources (e.g. photos, files etc.), they could be likewise applied to services for enhancing their descriptions. This metaphor is challenging and capitalizes on the view that Web 2.0 and service-orientation are two converging and complementary paradigms [19].

Within Web 2.0, new types of services appear which are created in a decentralized manner, e.g. RESTful services. These services are usually generated by users and not by service engineers. For example, mashups, are introduced as a new simple way of composing services and combining content from different sources. Thousands of mashups are available at the moment<sup>2</sup>, thus providing access to huge amount of distributed content and/or services. But services in Web 2.0 usually lack standardized descriptions from their providers, as there are neither standardized ways to describe services nor public repositories to store these descriptions.

In this work we will show how social metadata can be included in (and extend) semantic descriptions of services, either typical Web Services or RESTful services and mashups, in order to facilitate service search, mashing and recommendation.

### 3 Social Annotation of Services – The Social Contract

In [22] five complementary parts of a service's description, called service contracts, are defined which extend the typical definition of the service contract [1]. The five service contracts are:

---

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

- The *Information Model* which refers to the data model that is used to semantically describe the service inputs, outputs and fault messages.
- The *Functional Descriptions* which describe the service's functionalities.
- The *Non-Functional Descriptions* which define details related to the implementation or the running environment of the service, e.g. name, author, URL, version.
- The *Behavioral Descriptions* which define the service's choreography and internal workflow.
- The *Technical Descriptions* which define details regarding the format of the messages, the communication protocols and the access points of the service.

In our previous work, we reviewed several SWS efforts. We observed that they can be grouped into two distinct categories, namely typical *SWS frameworks* which refer to formal and complex efforts for semantically annotating services, and the emerging *lightweight semantic service frameworks*. In the first category we find OWL-S, SWSF, WSMO and WSMO-Lite, while WSDL-S and its successor SAWSDL, SA-REST and MicroWSMO are placed under the second category. Afterwards, we examined how the different SWS efforts address the five service contracts.

Summarizing our comparative analysis, we found out that SWS frameworks provide the language for encoding the ontologies that form their Information Model. On the contrary, semantic service models allow the use of any ontological language. Thus, their Information Model may be comprised by a set of ontologies encoded using different languages. Moreover, both the Functional and the Behavioural Descriptions can be expressed in detail using a SWS framework. Although the lightweight semantic service frameworks provide some means for specifying services' functionalities, these mechanisms lack in expressivity. Finally, regarding the Technical Descriptions both SWS frameworks and lightweight semantic service frameworks rely mainly on the WSDL specification, excluding the case of SA-REST and MicroWMSO that refer to RESTful services.

In all aforementioned approaches, the metadata included in all the service contracts come solely from the service providers. In this work we argue that it will be highly beneficial both for users and for service providers, if the semantic descriptions of the services were enriched with information that comes from the actual usage of a service in a bottom-up fashion. The semantic description in this case emerges from the usage of the service, thus giving a social aspect to service annotation.

In order to capture the social metadata and include them in the semantic description of the service, we introduce a new service contract, which we call *social contract*. The social contract expresses the way that the users' perceive the service, when and why they use the service etc. The social contract has until now been neglected when modelling and developing services. Fig. 1 summarizes our discussion so far and models it by means of a UML class diagram (the gray elements have been introduced by the authors).

The social metadata of services may derive explicitly or implicitly:

- *Explicitly*, where metadata is added by users who wish to describe the service in their own terms. Users can add metadata and annotate services, similarly to what they currently do for products, content and multimedia in platforms like Flickr or YouTube. Users would describe why they use a service, for what reason, on which occasion etc. They could also add annotations that are

related with attributes of the service like inputs or outputs. In other cases, users may express their satisfaction or dissatisfaction with regards to quality, usability, user-friendliness etc. However, some of the service attributes still remain to be described only by the service provider, e.g. the Technical Descriptions.

We propose the use of tags and tagging mechanisms as the means for enabling explicit service social annotation. Tagging is easy and straightforward and users are already familiar with it. Here lies a substantial difference between the annotations from the service providers and those that come from users. Service providers are more likely to use formal service models combined with structured vocabularies and ontologies. Users will rather use tags, either in the form of free text or coming from predefined vocabularies or folksonomies.

- Implicitly*, where information about the service can be inferred by monitoring the user’s behaviour while searching for or using a service and then enrich the respective service’s semantic description. For example, imagine that a statistically significant number of users search for services that provide benefits to families after having registered their marriage or users with similar profiles have an interest for some particular type of services, e.g. students search for services regarding housing/accommodation. In both cases, this information can provide us with interesting usage patterns that can be further exploited e.g. for implementing a service recommendation system or for identifying service chains or clusters of similar, complementary or mutually excluding services.

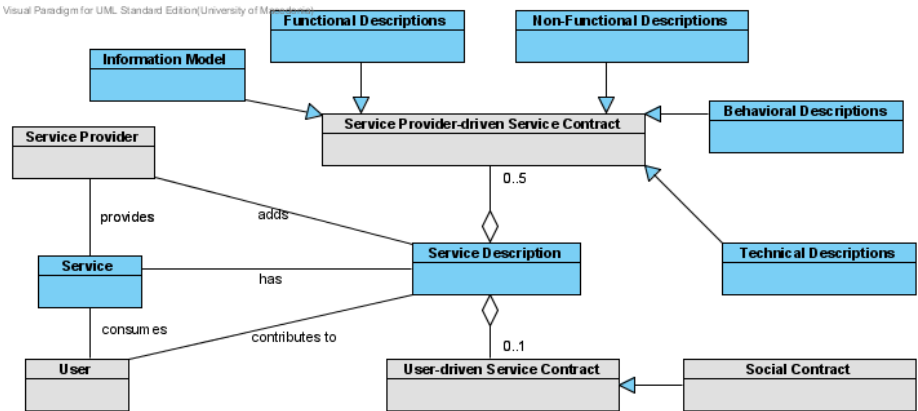


Fig. 1. The extended model of the semantic description of a service

#### 4 Including Social Metadata in Semantic Service Frameworks

Until now we have discussed at a conceptual level how social metadata can be included in existing semantic service models. In this section we will show how we applied this idea by extending an existing semantic service model, namely SA-REST.

SA-REST [20] introduces a lightweight approach for adding semantics to RESTful WS. It assumes that it is highly likely that when a WS is made available online, the provider will release an (X)HTML description of the service as well. Thus, SA-REST suggests to semantically annotate this (X)HTML service description using RDFa or GRDDL. SA-REST uses a set of predefined elements for annotating different attributes of a service, e.g. `input`, `output` and `sem-class`.

SA-REST is simple and easily extendable, mainly due to the fact that it is based on RDFa. RDFa allows the inclusion of bits of semantic information in XHTML files. RDFa became a W3C recommendation in 2008<sup>3</sup>. SA-REST became recently a W3C Member Submission<sup>4</sup>.

The lightweight semantic annotation of services, as suggested by SA-REST, fits very well with our line of work. Moreover, as the number of RESTful services is increasing, influenced by the growth of Web 2.0, we expect lightweight efforts to semantically annotate services using RDFa to gain popularity. Hence, we chose SA-REST in order to semantically annotate services and capture and include social metadata.

SA-REST like other similar approaches for semantically annotating services covers the five service contracts but does not support the social contract. SA-REST assumes that the user has a passive role and that the service descriptions come only from the service provider. In brief, the information model of a service is identified by its inputs and outputs. Both the behavioural and the functional Descriptions are weakly defined in the SA-REST model due to the fact that SA-REST aims at providing a lightweight approach for service descriptions sacrificing its expressivity. Finally, technical information can be represented by the method and protocol elements, while, non-functional descriptions can be derived from the `domain-rel`, `sem-class` and `sem-rel` elements.

In order to include social metadata in SA-REST, we first express social metadata by means of a simple folksonomy and then make use of the `sem-class` element to create a link between a service and this folksonomy. In order to indicate that this `sem-class` element refers to the social metadata of this service, we make use of the `typeof` element and make a reference to the social contact concept of our model.

It is worth mentioning that the social contract, as introduced in this work, can be combined with any other SWS framework. For example, WSMO non-functional properties Web Services or a SAWSDL model references enable the inclusion of social metadata in WSMO and SAWSDL services respectively.

## 5 Prototype

For the needs of our prototype, we have scoped our focus on the eGovernment domain. We have selected eGovernment as our application domain due to previous work and expertise in the domain, but also because eGovernment is a challenging test-bed with thousands of services provided worldwide by public agencies to billions of clients. The clientele of public administration is not restricted to a certain group of people with common needs or interests. In fact, public administration tries to cover the

<sup>3</sup> <http://www.w3.org/TR/rdfa-syntax/>

<sup>4</sup> <http://www.w3.org/Submission/2010/02/>



needs of practically each and every citizen. To achieve this, public administration tries to group the diverse needs of its clientele and translate them into services.

Social annotation of services can provide valuable input for public administration throughout the public service lifecycle, i.e. design, development, deployment, that could lead to higher quality services which would fit better to the clients' needs. Furthermore, public administration could predict future needs of its clientele, based on trends expressed through the social annotation of services. This will allow public administration to be more agile and proactive.

Our governmental portal plays the role of a national entry point to the services provided by public administration. In fact a prototype of the portal, which is available at <http://195.251.218.39/cyprus>, is currently used in a pilot study in Cyprus. Citizens use this portal to get information about public services. The descriptions of the services that are made available via the portal are semantically annotated using the extended semantic service model that is presented in section 4. Apart from the inclusion of social metadata, we have also included eGovernment domain specific semantics in the semantic descriptions of our services. Towards this direction, the GEA Public Service model [15, 16] was employed, which introduces a conceptual representation of a public service. As such, it introduces core concepts of a public service, such as service input and output, service provider, service preconditions, service domain etc. We used RDFa in order to include in our semantic eGovernment service descriptions eGovernment domain specific semantics (Table 1). The semantic descriptions are initially automatically created on the service provider's side, but once they are released and used by the users, they are enriched and become more expressive as social metadata are added.

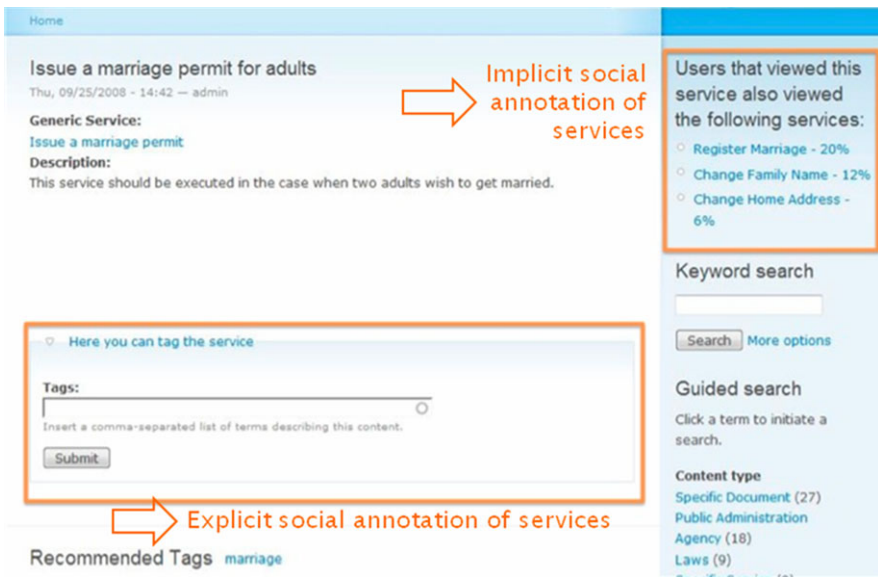


Fig. 2. Implicit and Explicit social annotation of services in the prototype

In our social eGovernment portal users can tag the services provided or use the tags already given by other users. Moreover, the users' behaviour while using the portal, e.g. when navigating from one service to another, is anonymously monitored. This allows us to extract social metadata from their behavioural patterns without violating their privacy. Fig. 2 illustrates our discussion so far.

In both cases mentioned above, social metadata is extracted, which is then used for enhancing the user's experience in the portal. Users can browse the underlying eGovernment service repository using different mechanisms, e.g. tags/tagclouds, keyword search. Moreover, they can get recommendations about services that are popular or beneficial or fit their profiles or are related with services that they have already used. A detailed description of the portal can be found in [8].

**Table 1.** Example of service description annotated with the extended SA-REST model

```
<div xmlns:SA-REST="http://knoesis.wright.edu/srl/SA-REST"
  xmlns:islab="http://islab.uom.gr">
  <div about="http://islab.uom.gr/DrivingLicense">
    The Driving License Issuance public service is provided by
    the <div rel="islab:serviceprovider"
      resource="http://islab.uom.gr/gea.owl#Prefecture"/> Regional
    Authorities and belongs to the <div rel="SA-REST:domain-rel"
      resource="
      http://islab.uom.gr/gea.owl#SubDomainCommunityAndSocialServices
      "/> Community and Social Services domain
    <div typeof="islab:social" rel="SA-REST:sem-class"
      resource=http://islab.uom.gr/DL_social.rdf/>
  </div> </div>
```

## 6 Conclusions and Future Work

In this paper we discussed the need for extending the existing SOA-based service descriptions by including socially derived metadata. We add a "social" layer on top of existing semantic or non-semantic service-oriented systems. This layer enriches the service description and facilitates the discovery and automatic clustering of services.

Two mechanisms for capturing this metadata have been proposed, receiving the annotations explicitly and directly by the users through tagging and implicitly through analysis of users' behaviour and usage patterns. In order to validate our arguments an extension of SA-REST which accommodates social metadata was proposed. In addition to that a social eGovernment portal was developed.

Harvesting the knowledge that can be extracted from social the social metadata of services is expected to benefit both users and service providers. From the user's perspective, it allows them to:

- Express their view of the services they use by annotating these services, e.g. using tags.
- Personalize their service searches, thus improving the quality and the coherence of the result set and bridging the service discovery gap [6]. Social metadata, e.g. tags, can be added as criteria to search queries, thus narrowing down the result sets.

- Get recommendations about related services. Service platforms may suggest to their users services that share common tags.
- Form communities of interest and/or practice. As described earlier, social semantics can also be extracted by monitoring the users' behaviour. In this case, users that tend to use similar services can be grouped in communities of interest. For example, a community of users who use online collaboration services.

Service providers can exploit the social annotation of services to:

- Improve the classification and clustering of services according to their functionality/output/behaviour etc. Since the semantics that emerge from the social annotation of services stem directly from the users, they can create a bottom-up classification of services.
- Get feedback on their services and improve their quality or design new services in order to cover emerging customers' needs.

The benefits that social metadata are anticipated to have both to uses and service providers, set also the pillars of our future research plan. Hence, we will try draft and develop possible applications and prototypes and enhance existing systems by including and harvesting social metadata.

**Acknowledgments.** This work is supported in part by Science Foundation Ireland under grant SFI/08/CE/I1380 (Líon 2) and in part by the Rural Inclusion project ([www.rural-inclusion.eu](http://www.rural-inclusion.eu)) under the EC Competitiveness & Innovation Framework Programme.

## References

1. Papazoglou, M.: The Challenges of Service Evolution. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 1–15. Springer, Heidelberg (2008)
2. Bao, S., Xue, G., Wu, X., Fei, B., Su, Z.: Optimizing web search using social annotations. In: 16th World Wide Web Conference, pp. 501–510 (2007)
3. OASIS SOA Reference Model TC: Reference Model for Service Oriented Architecture (2009), <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf>
4. Domingue, J., Fensel, D., González-Cabero, R.: SOA4All, Enabling the SOA Revolution on a World Wide Scale. In: 2nd IEEE International Conference on Semantic Computing (2008)
5. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema, W3C Recommendation (2007), <http://www.w3.org/TR/sawSDL/>
6. Fernandez, A., Hayes, C., Loutas, N., Peristeras, V., Polleres, A., Tarabanis, K.: Closing the Service Discovery Gap by Collaborative Tagging and Clustering Techniques. In: International Semantic Web Conference, Workshop on Service Discovery and Resource Retrieval in the Semantic Web (2008)
7. Gruber, T.: Ontology of Folksonomy: A Mash-Up of Apples and Oranges. International Journal of Semantic Web and Information Systems 3, 1–11 (2007)

8. Kopouki Papathanasiou, M., Loutas, N., Peristeras, V., Tarabanis, K.: Combining service models, semantic and Web 2.0 technologies to create a rich citizen experience. In: Lytras, M.D., Damiani, E., Carroll, J.M., Tennyson, R.D., Avison, D., Naeve, A., Dale, A., Lefrere, P., Tan, F., Sipior, J., Vossen, G. (eds.) WSKS 2009. LNCS, vol. 5736, pp. 296–305. Springer, Heidelberg (2009)
9. You, G., Hwang, S.: Search structures and algorithms for personalized ranking. *Journal of Information Sciences* 178(20), 3925–3942 (2007)
10. Martin, D., Domingue, J.: Semantic Web Services, Part 2. *IEEE Intelligent Systems* 22(6), 8–15 (2007)
11. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services. W3C Member Submission (2004), <http://www.w3.org/Submission/OWL-S/>
12. Meyer, H., Weske, M.: Light-Weight Semantic Service Annotations through Tagging. In: Dan, A., Lamersdorf, W. (eds.) ICSSOC 2006. LNCS, vol. 4294, pp. 465–470. Springer, Heidelberg (2006)
13. Miller, J., Verma, K., Rajasekaran, P., Sheth, A., Aggarwal, R., Sivashanmugam, K.: WSDL-S: Adding Semantics to WSDL - White Paper. Large Scale Distributed Information Systems (2004), <http://lsdis.cs.uga.edu/library/download/wSDL-s.pdf>
14. Murugesan, S.: Understanding Web 2.0. *IT Professional* 9(4), 34–41 (2007)
15. Peristeras, V., Tarabanis, K.: The Governance Architecture Framework and Models. In: Saha, P. (ed.) *Advances in Government Enterprise Architecture*. IGI (2008)
16. Peristeras, K., Tarabanis, K.: Towards an Enterprise Architecture for Public Administration: A Top Down Approach. *European Journal of Information Systems* 9, 252–260 (2002)
17. Roman, D., Keller, U., Lausen, H., Bruijn, J.d., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web Service Modeling Ontology. *Applied Ontology* 1(1), 77–106 (2005)
18. McCool, R.: Rethinking the Semantic Web, Part 1. *IEEE Internet Computing* 9(6), 88, 86 - 87 (2005)
19. Schroth, C., Janner, T.: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IEEE IT Professional* 9(3), 36–41 (2007)
20. Sheth, A., Gomadam, K., Lathem, J.: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. *IEEE Internet Computing* 11(6), 91–94 (2007)
21. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Franconi, E., Kifer, M., May, W. (eds.) *ESWC 2007*. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)
22. Vitvar, T., Kopecky, J., Fensel, D.: WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web. CMS WG Working Draft (2008)
23. Alani, H., Hall, W., O'Hara, K., Shadbolt, N., Szomsoz, M., Handler, P.: Building a Pragmatic Semantic Web. *IEEE Intelligent Systems* 23(3), 61–68 (2008)
24. Xuan, S.: Semantic Web Services: An Unfulfilled Promise. *IEEE IT Professional* 9(4), 42–45 (2007)

# Service Composition for Everyone: A Study of Risks and Benefits

Abdallah Namoun, Usman Wajid, and Nikolay Mehandjiev

Manchester Business School, Booth Street East,  
Manchester, M13 9SS, United Kingdom

{abdallah.namoun,usman.wajid,nikolay.mehandjiev}@mbs.ac.uk

**Abstract.** In this paper, we investigate web users' mental models of services, the underlying risks and benefits of service composition, and the problems anticipated while combining web services into complete interactive applications. The study comprised three focus groups integrating group discussions and questionnaires, with a total of 35 participants, the majority without specialist programming skills. The results of the focus groups revealed a high degree of optimism towards service composition and consumption. However, several concerns, primarily related to personal privacy, trust, and technical difficulty, were highlighted during the focus groups. This paper discusses these concerns and proposes some ideas about how to address them.

**Keywords:** Web services, service-based applications, service composition, end user development.

## 1 Introduction

Service Oriented Architecture (SOA) technologies are becoming very popular on the Internet, especially in the form of independent services [1]. Their key benefit is re-useability, enabling the loose coupling of existing services to produce new augmented web services through the process of "service composition". Whilst only a small proportion of users, often with considerable computing expertise, can construct rich service-based applications, the majority of online users are unable to exploit the advantages offered by SOA technologies and develop service-oriented applications tailored to their needs. This difficulty can be linked to the complexity of the composition process and to the limited technical knowledge of ordinary users. In this respect, the research challenge lays in simplifying the application composition process and abstracting this process from technical difficulties. Such research promises to promote the consumption and reuse of web services, especially among web users.

When creating such user-friendly service composition interfaces, we also need to consider user expectations regarding the trade-off between the costs of learning new tools and the benefits they expect to get from using them. For example, the spreadsheet interface hides aspects such as order of calculations and propagating updates, and minimises learning costs by using familiar metaphor of calculation tables and accounting books. The balance between costs and benefits is likely to differ for different groups of users and different target domains (e.g. [7, 8]), yet we believe that

identifying user attitudes and expectations towards service composition is a key to predicting successful uptake [7, 8, 14]. Therefore it is the focus of the study reported in this paper.

Currently, end users can add web services as widgets/gadgets to their personal pages in a lightweight manner; such as: Facebook<sup>1</sup>, iGoogle<sup>2</sup> and myYahoo<sup>3</sup>. Users of these websites can select from a list of services and position them on their personal pages. The services are visually represented as independent, interactive, and customisable windows. This widget-based model is simple and allows hosting various services together, but it does not support service composition. Indeed, the web services, represented as widgets, are autonomous and do not interact with each other, thus restricting their usefulness for creating more complex assemblies. For instance, given a flight, car, hotel, card payment, and insurance service, users should be able integrate them to form a mini-holiday organizer application. Service composition not only fulfils users' needs but also allows easy extension and customization of applications; thus, saving considerable time and resources.

Another advanced and rich approach to end user development of web applications follows the mash-up model. In this case, end users combine existing services and web feeds from multiple sources into a single web-based application using specialized mash-up editors, such as: Open Mashups Studio [12] and Yahoo!Pipes [15]. Unfortunately this approach relies on the modelling skills needed to understand the data flow between services, whilst placing strong emphasis on data aggregation and giving less importance to functionality aggregation. Whilst the widget-based model does not support any interaction between heterogeneous services, the mash-up based model is complex and lacks flexibility. This motivates the pressing need for more effective approaches to compose low-level services into interactive service-oriented applications by non-programmers. Easy to use and flexible service composition authoring tools that simplify the composition process should be offered. This is the main objective of the EC funded project, SOA4All<sup>4</sup>.

Here we report on a study which aims to identify the balance between user expectations about costs and benefits of the SOA4All vision, and to report users' concerns and background as relevant to this vision. This paper focuses on service composition and consumption by human actors and not by software agents. Focus groups were used as a self-contained method to conduct this study since no suitable prototype was available to evaluate at that stage and it is a useful technique for providing detailed insights into opinions and experiences of participants [9].

## 2 Service Composition by End Users

Service Composition is broadly supported by two main approaches: workflow-based scripting of service components, and AI-based automatic composition of service components, reasoning with pre- and post-conditions. Further details are available elsewhere [3, 13].

---

<sup>1</sup> <http://www.facebook.com>

<sup>2</sup> <http://www.google.com/ig>

<sup>3</sup> <http://my.yahoo.com/>

<sup>4</sup> <http://www.soa4all.eu/>

A large number of visual representations for service composition and interaction have been proposed with the purported aim to make the composition more user-friendly (e.g. Zenflow [5]). However, most of them are *ad hoc*, i.e. they use technology-led representations and metaphors, which are not derived from user studies. Only a few of them have been evaluated in terms of usability and cognitive effectiveness. For example, Lets Dance [16] has been evaluated using the framework of Cognitive Dimensions [2], but iterative testing and enhancement have not been documented in the related references. Vitabal WS [4] is a version of an earlier visual language tuned to the needs of web service composition. It has been evaluated using the cognitive dimensions framework, yet it targets experienced web service developers and hence would have different characteristics from the service composition representations to be developed by SOA4All.

We believe that technology-led *ad hoc* visualizations will not work. Indeed opening up service use and development to people who are not professional programmers (we call them end users) requires the delivery of user interfaces that are task-oriented rather than technology-oriented, that is they should be tuned to the expected skills and foreseen tasks of our target users. Activities such as service construction and composition will involve non-trivial problem-solving in a context called End User Development (EUD). EUD research results provide an insight into the type of software interfaces and motivational factors likely to support end user activities.

Sutcliffe *et al.* [14] see the trade-off between expected benefits and learning costs as a main determinant of uptake of an End User Development tool by its users. This has been extended to organizational context by Mehandjiev *et al.* [8], who identify a number of risks and benefits for end users being involved with the development of software, including the construction of software services. These factors have then been used to underpin a number of quantitative studies in concrete domains, aiming to elicit the likelihood of uptake for end user development ideas in the specific context of that domain (e.g. [7]). The workshops reported here are an example of one such application of this approach to the target domains of SOA4All.

Several research studies have attempted to explore end user perception of software development, for example: McGill and Klisc [6] argue that end user developers of web development are aware of the associated risks and benefits and it is crucial to involve them in the development of approaches to minimise risks. Due to the difficulty of learning traditional programming languages, Myers *et al.* [10] reported a number of studies aiming to elicit understanding of how people think about a particular task and design natural programming languages and environments that support the way end user developers are thinking. More recently, Namoune *et al.* [11] presented a user study in which potential problems of service composition were extracted using a low-fidelity visual composition prototype, showing that end users have difficulty connecting services together and understanding specialised service-related terms such as: operations, parameters, data types. Overall, review of available literature demonstrates that research in end user development of service-based applications is quite weak and most studies are in their infancy.

### 3 Methodology

Three separate focus groups, involving 35 participants without programming skills (25 students and 10 academic and research staff) (range 19 to 40 years with a mean of 26

years) were undertaken within the Centre for Service Research at the Manchester Business School to acquire a better understanding of end users' perception about web services, and the likelihood of uptake of user development. Each focus group lasted for approximately one hour; participant responses were recorded using audio recorders and questionnaires. The overall strategy was to first introduce participants to the topic of "web services composition by end users" through a presentation, secondly capture their subjective judgment about the topic using a questionnaire, and finally discuss several issues in small groups. All participants were invited to perform these tasks:

- 1- Provide a definition of "services"
- 2- Listen to a 20 minute presentation in which they were familiarized with web services and the concept of service composition; this was facilitated by examples
- 3- Fill in a service composition questionnaire
- 4- Discuss the potential risks and benefits of service composition and anticipate the composition-related problems; this was carried out in small discussion groups containing 5 participants each
- 5- Propose solutions to resolve the highlighted problems

### **3.1 Service Composition Questionnaire**

The service composition questionnaire used in our study focused on three main parts, user background and experience, rating of usability aspects of service composition, and ways of encouraging service composition by users (as illustrated in table 1). Although the questionnaire contains some questions which are difficult to assess at this stage, for example, it is practically difficult to assess whether "composition is easy to achieve" without actually trying it, the principal aim was to drive first impressions about service composition and most importantly to check users' acceptability of this innovative idea. Moreover, these results will provide a reference point to upcoming empirical evaluation studies when end users perform composition activities using our composition authoring tool (i.e. SOA4All Studio).

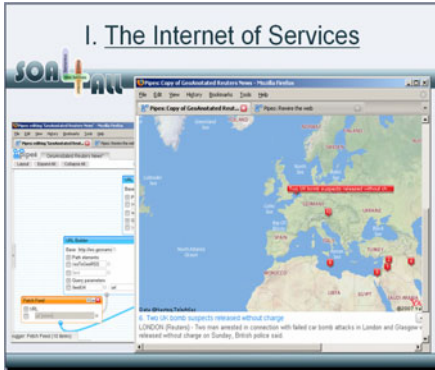
### **3.2 Introductory Presentation "The Internet of Services"**

The introductory presentation, presented by one of the authors, aimed to introduce the concept of service and provide examples of service composition. It explained the difference between conventional services, software services and hybrid services, where human-performed services are enabled through software interfaces and services, such as buying a book through Amazon.com. The influence of current Web2.0 technologies was argued to enable end users to take part in the development of the web, and the idea is to move this influence to the internet of services. Following this, Yahoo! Pipes was used as a motivating example (Figure 1). Figures about the number of web services found were also reported (27,684 services and 7284 providers during the last 2 years), as suggested by the SEEKDA<sup>5</sup> service crawler. Next, the motivation behind SOA4All was introduced to the attendees, with the project aiming to transform the current web of information into a web of services through which users of services could also become producers of applications (or what we call "Prosumers").

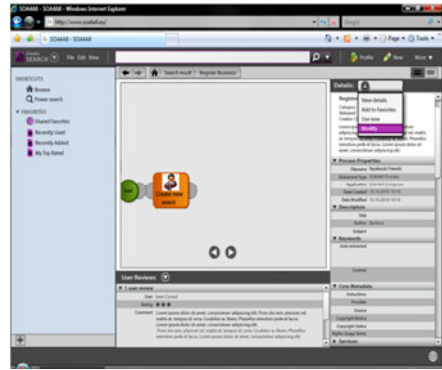
---

<sup>5</sup> <http://webservices.seekda.com/>





**Fig. 1.** Yahoo! Pipes as a stimulating example (Left)



**Fig. 2.** A mockup of the SOA4All Studio – a user-friendly composition tool under development in SOA4All (Right)

Then the scenario driving further discussions was introduced, the creation of a *Meet Friends* composite service. This hypothetical composite service allows a particular user to organise a meeting with friends at short notice. The Meet Friends composite service contains four services; service one fetches the address of friends from social networking sites (e.g. Facebook), service two finds out which friends are in the vicinity of the target venue, service three finds out weather and travel information for proposed meeting venue from a 3<sup>rd</sup> party, and service four sends out invites and directions using an SMS service. Finally, the presenter showed some mockups of a future authoring service composition tool (Figure 2). Participants were invited to ask questions related to aspects of the presentation before starting the focus groups.

## 4 Results

The results of the three focus groups undertaken are grouped into three main topics: perspectives of end users on services and service composition, risks and benefits of service composition, and issues of service composition:

### 4.1 Perspectives of End Users on Software Services and Service Composition

The pre-test questionnaires revealed that more than 85% of the participants considered themselves as “not experts” in software and service development. 60% of the users specified that they have “never or less often” composed services or built service based applications. The qualitative analysis of the responses gathered in the focus groups showed that 25 user comments relate to service understanding. The results demonstrated diverse user understanding/definitions of services; these definitions varied between: features assisting users, solutions to issues, components of business process, offerings to customers, information provision, and execution of transactions. In general, users’ definitions concentrated on two main aspects, (1) describing attributes/features of services

such as: services are intangible and they have a back end, (2) describing specific interactions with users in the form of service consumption, such as: providing users with information, helping users, and delivering expertise.

When asked whether service composition is interesting, 80% of users showed a high level of interest (mean = 4.20 /5, questions were rated on a five-point Likert scale where 1 corresponds to disagree and 5 corresponds to agree). Users also rated the usefulness of service composition high (mean = 4.44 /5), as well as the efficiency of service composition in promoting the accomplishment of online activities (mean = 4.12 /5). However, service composition by end users was regarded nor easy neither difficult (mean = 3.32 /5). In terms of error-proneness, fears were evident about the possibility of creating errors by ordinary web users (mean = 2.54 /5). Users concerns that relate to disruptive use of service composition (i.e. service composition can be used to break organizational rules and policies) were rated high (mean = 3.5 /5). Finally, 77% of the users disagreed or remained natural in regards to the question: “service composition by users is unfeasible” (mean = 2.26 /5).

In regard to user support, users agreed that successful examples (mean = 4.69) and training courses (mean = 4.38) could encourage people to be actively involved in the composition of services and development of service based applications. In summary, end users demonstrated a high level of interest and strongly agreed that service composition is useful and possible, but expressed uncertainty about the difficulty and potential misuse of service composition by the general public (Table 1).

**Table 1.** Service composition questions, rated between (1= disagree and 5= agree)

<b>Service composition by users</b>	<b>Average rating</b>	<b>SD</b>
I find web service composition interesting	4.20	0.76
... is useful	4.44	0.82
... brings about a more efficient way of conducting on-line activities	4.12	0.96
...is easy to achieve	3.32	1.19
... is unfeasible	2.26	1.18
... is error-prone	2.54	0.87
... can be used to break organisational rules and policies	3.50	1.08
<b>Ways of encouraging and supporting Service composition by users</b>		
Examples of successful SCU can stimulate one to try it	4.69	0.52
Recognising and rewarding SCU effort will make people more willing to try it	4.15	0.90
Attending a training course could help people to start SCU	4.38	0.77
SCU quality standards and testing will decrease risks	4.32	0.76

## 4.2 Risks and Benefits of Service Composition by Users

The discussion about the balance between risks and benefits is based on work [7, 8, 14] explaining the uptake of software development by end users (known as End User Development) as a rational economic decision based on the balance of perceived costs and perceived benefits of each user. The ongoing program of research in this area aims to analyse the factors which impact this perceived balance, and to discover organizational and technical strategies which aim to tip the balance in favour of the benefits, thus supporting the uptake of such technologies.

In terms of benefits, discussions in the focus groups mainly focused on the usefulness of reusing composition knowledge (40% out of all benefit responses), and the time users can save as a result of this (30% out of all benefit responses). Giving ordinary users control over service composition would empower them to produce various service oriented applications that can be tailored to their needs (15% out of all benefit responses), such as meta-search engines, thus saving them time and enabling them to obtain rich results.

In terms of risks, the biggest fear was about losing control over personal information (8% out of all risk responses), especially when the effect is mediated through the effect of social interactions (e.g. your friends exposing information about you), or through the service provider (information aggregator), which may pass your personal information (e.g. phone number) to other sub-contracting services, which may or may not be bound to the data protection principles. Technical difficulty imposed by service compose was also amongst the biggest fears of end users (8% out of all risk responses). Errors in putting information together were also possible, especially when the composition is performed by inexperienced users and un-trusted third parties.

Moreover, users felt that services may no longer be there when they need them, and that any recommendation support for services may be biased to a set of services. The participants also discussed what could be the social and organisational support for user-based service development. The following ideas emerged:

- “Go with the flow” – once everybody is doing it, people will join, mirroring success in other technologies;
- Non-trivial examples of successful use will also help (to sell benefits), this was felt quite strongly;
- Community-level control mechanisms such as feedback, etc. would ensure validation of services and, together with a validating body may help to ensure the trust, which is considered vital for uptake of user-driven service composition.

## 4.3 Service Composition Problems

Although users favoured the idea of assembling services to formulate interactive applications that fulfils their daily needs, several service composition-related issues were raised, in particular:

- Services complexity and terminology: services are usually represented using their functional elements (operations and parameters) which are often not understood by ordinary web users.

- Services compatibility: users expressed frustration in regards to aggregating heterogeneous services from different service providers. How do they ensure the business services they are trying to combine together are technically compatible with each other?
- Composition steps: users agreed that it might be problematic to define the single steps required to combine services together and the order in which these services should be executed due to their lack of technical knowledge and skills. This issue becomes more complicated in the case of many services (for example: 100 atomic services).
- Other less aggravated user interface-related concerns evolved around the use of the service composition editor, for example: direct manipulation of web services (i.e. selection, deletion, etc) within the design space could be the main source of frustration.

In terms of technical support which can be provided by the composition editor, the following themes emerged:

- The difference between naïve and professional users was felt to lie partially in the awareness about the consequences of one's actions; this awareness should be supported;
- Full automation such as Google search results will frustrate owing to lack of control by the end users, a balance should be maintained;
- Tools should offer clarity of process in respect to building and using;
  - Context and personalization;
  - Reuse of designs.

## 5 Discussion

End users with no or little computing knowledge showed either no or basic knowledge of the technical aspects of services, i.e. they could not provide a technical definition of services. This result is expected as our target group has no specialist technical skills. Essentially, they perceived services as elements which deliver services (be it information, help, solutions ... etc) to accomplish specified users goals. This view emphasises that services need to be abstracted from their technical complexity and presented in a way that efficiently describes their purpose/functionality, especially for ordinary web users.

Users showed a high likeability towards the idea of composing services into personalised interactive applications. This agrees with the current trends that end users are becoming proactive about developing the web. Users argued that service composition will save them time and enable them to develop applications on the fly and without the need to acquire considerable technical knowledge. Hence, it is important that end users are able to develop service-based applications without the need to learn programming languages and modelling notations.

To overcome the aforementioned problems, various tentative remedies that will form the functional requirements of a future visual service composition authoring tool -currently under development- are proposed as follows:

1. *Promote service composition awareness*: even though web users have experience adding autonomous services to their networking or personalised sites, the composition of services imposes a totally new and different challenge. Therefore, the composition editor should clearly communicate “the composition aspect” of services. Users’ awareness of the possibility to develop service-based applications should be elevated via the right amount of publicity to familiarize ordinary people with SOA technologies..
2. *Simple service composition*: this research aims to increase service reuse by ordinary users, it is therefore crucial to simplify service composition by hiding the technical aspects of services from users. Composition should be as easy as dragging and dropping a service into a design space (i.e. visual manipulation of services), followed by creating connections between the selected services. No programming knowledge or expensive training should be required.
3. *Guided service composition*: users should be supplied with wizards, tutorials, and help messages to guide them through the composition process within an easy to use composition tool. This is particularly important to overcome the services compatibility and composition steps definition problems.

## 6 Conclusion

This paper reports on the results of three focus groups aiming to gauge end users’ perception of web services and their acceptability of service composition. Generally, users showed a high willingness to develop interactive service-oriented applications, but expressed fears that relate to the complexity underlying the composition process and to the knowledge required to build software applications. In future research, various composition design approaches of different complexity levels will be offered to accommodate end users with various skills and backgrounds within an easy to use online authoring tool, formally known as SOA4All studio.

**Acknowledgments.** This research work is sponsored by the EC-funded project SOA4All.

## References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: *Web Services: Concepts, Architectures, and Applications*. Springer, Heidelberg (2004)
2. Green, T.R.G.: Cognitive dimensions of notations. In: Sutcliffe, A., Macaulay, L. (eds.) *People and Computers V*, pp. 443–460. Cambridge University Press, Cambridge (1989)
3. Rao, J., Su, X.: A Survey of Automated Web Service Composition Methods. In: Cardoso, J., Sheth, A.P. (eds.) *SWSWPC 2004*. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005)
4. Li, K.N.-L.: *Visual Languages for Event Integration Specification*. PhD Thesis, University of Auckland, Department of Computer Science (2008)
5. Martinez, A., Patino-Martinez, M., Jimenez-Peris, R., Perez-Sorrosal, F.: ZenFlow: A Visual Web Service Composition Tool for BPEL4WS. In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*. VLHCC, pp. 181–188. IEEE Computer Society, Washington (2005)

6. McGill, T., Klisc, C.: End User Perceptions of the Benefits and Risks of End User Web Development. *Journal of Organizational and End User Computing* 18(4), 22–42 (2006)
7. Mehandjiev, N., Stoitsev, T., Grebner, O., Scheidl, S., Riss, U.: End User Development for Task Management: Survey of Attitudes and Practices. In: *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing*, Herrsching am Ammersee, Germany. IEEE Press, Los Alamitos (2008) ISBN : 978-1-4244-2528-0
8. Mehandjiev, N., Sutcliffe, A., Lee, D.: Organisational View of End-User Development. In: Lieberman, H., Paterno, F., Wulf, V. (eds.) *End User Development. Human-Computer Interaction Series*, vol. 9(XVI), 492 p. (2006) (Hand cover) ISBN: 1-4020-4220-5
9. Morgan, D.L.: *Focus Groups as Qualitative Research*. Sage Publications, California (1997)
10. Myers, B., Pane, J.F., Ko, A.: Natural Programming Languages and Environments. *Communications of the ACM (Special issue on End-User Development)* 47(9), 47–52 (2004)
11. Namoune, A., Nestler, T., Angeli, A.D.: End User Development of Service-based Applications. In: *2nd Workshop on HCI and Services at HCI 2009*, Cambridge (2009)
12. Orange Labs, Open Mashups Studio, <http://www.open-mashups.org/> (last accessed on October 30, 2009)
13. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. *Computer* 40(11), 38–45 (2007), <http://dx.doi.org/10.1109/MC.2007.400>
14. Sutcliffe, A., Lee, D., Mehandjiev, N.: Contributions, Costs and Prospects for End-User Development. In: *Proceedings of HCI International*. Lawrence Erlbaum Associates, Inc., New Jersey (2003)
15. Yahoo! Pipes, <http://pipes.yahoo.com/pipes/> (last accessed on October 30, 2009)
16. Zaha, J.M., Barros, A.P., Dumas, M., ter Hofstede, A.H.M.: Let's Dance: A Language for Service Behavior Modeling. In: Meersman, R., Tari, Z. (eds.) *OTM 2006, Part I. LNCS*, vol. 4275, pp. 145–162. Springer, Heidelberg (2006)

# Using Personal Information Management Infrastructures to Facilitate User-Generated Services for Personal Use

Olaf Grebner

SAP Research, Vincenz-Priessnitz-Str. 1, 76131 Karlsruhe, Germany  
Olaf.Grebner@sap.com

**Abstract.** Ad-hoc and situational applications for personal use will gain more and more traction in the work support for knowledge workers (KWer). Personal information is a key element in these applications. Composition environments for situational applications like, e.g., Yahoo Pipes, enable end-users to compose services into an application targeting their individual problems. However, we analyze that these composition environments lack access for a KWer's personal information and require redundant development of services for common KWer activities. Addressing these issues, we present<sup>1</sup> an infrastructure that manages the KWer's personal information consistently and thus provides services that serve as basis for enabling end-user driven service composition for application for personal use. The infrastructure consists of two key components, a basic personal information management system to maintain a KWer's personal information cloud in a unified and integrated form and domain-specific services that offer business logic for frequently occurring activities in applications for a KWer's personal use.

**Keywords:** Service design, User-centric software development, Personal information management.

## 1 Introduction

*Ad-hoc and situational applications for personal use* will gain more and more traction in the work support for knowledge workers (KWer). The work conducted by *knowledge workers* is characterized by variety rather than routine and there are few activities that can be automated. Each KWer possesses an individual working style and skill level making it hard to design applications that fit to a broad range of KWers. *Situational applications* enable a KWer to adapt services to create functionality and solve problems in their work. This allows the KWer to leverage an individually created application tailored to the needs of that individual KWer for the particular use case. The *advantages of situational applications* supporting a KWer are manifold. On the one hand, the application fits to the KWer's individual skill level, targeting both novices and experts appropriately. On the other hand, KWers being experts in their respective domain can come up with design ideas for solutions target their individual problems of which they have a deep understanding.

---

<sup>1</sup> This work was has been partially funded by the European Commission as part of the NE-POMUK IP (Grant FP6-027705) and of the MATURE IP (Grant FP7-216346).

*Personal information* is a key element in applications for personal use. A KWer deals with personal information in applications for personal use. A KWer's *personal information* refers to the information that is owned by an individual KWer. This includes for example email messages in an email account or files on the computer's hard drive. A *personal information cloud* (PIC) is "the 'working set' of information that is relevant to the individual and his work" [Moran&Zhai, 2007, p. 338], i.e., the information an individual KWer manages for herself and deals with in the activities she executes. The KWer regards the personal information as "*one single body of information*" [Ravasio&Tschertter, 2007, p. 275], complementary to the personal information cloud. "Ravasio et al. (2004) indicate that users explicitly desire linking: 'most interviewees expressed the need to have their information linked together, e.g., article author and respective address book entry, or citation and cited article, etc.'" (p. 169)" [Jones&Teevan, 2007, p. 143] [Ravasio et al., 2004].

Composition environments for situational applications *lack access for a KWer's personal information* and require redundant development of services for common KWer activities. Research fields such as end-user-development acknowledge the importance of situational applications where individuals drive the creation of an application to support their individual problems. On the web, mash-up services enable an individual KWer to perform a specific form of end-user development already today. Mash-up services like, e.g., Yahoo Pipes [Yahoo! Inc., 2009] allow end-users to create custom situational applications for a particular use case by visually composing web-based services and designing corresponding user-interfaces. However, these mentioned mash-up services lack two core elements when targeting applications for personal use:

- Consistent access to a KWer's personal information is not possible, as the KWer's personal information is scattered across multiple applications. This is a key requirement as the KWer needs to have her personal information available during work with these applications.
- High effort for developing applications in the personal domain due to a lack of services that offer business logic for common, recurring activities in the KWer's personal domain, especially when dealing with personal information.

Personal information management research started to integrate the KWer's personal information cloud into such composition environments, for example Konduit [Dragan et al., 2009]. However, Konduit focuses solely on integrating desktop data into the built applications, not tackling the problem of high development effort for domain-specific components dealing with a KWer's activities.

The next section discusses in detail the respectively underlying problems of these issues. These issues so far prevented the large-scale adoption of mash-up services for personal use. To address these issues, we present an infrastructure that manages the KWer's personal information consistently and provides services that serve as basis for enabling end-user driven service composition for application for personal use. The infrastructure consists of two key components:

- We leverage a *basic personal information management system* to maintain a KWer's personal information cloud in a unified and integrated form by using a semantic



desktop. Services can access all of the KWer's personal information by querying the personal information cloud's consistent and integrated set of information.

- We provide *domain-specific services* for composition that cover common business logic which frequently occurs in applications for a KWer's personal use. These domain-specific services cover for example activities like task management or meeting management. They enable the end-user to access and act on the personal information and represent a basic infrastructure for the KWer's personal domain in end-user driven composition environments like the mentioned mash-up services.

The here presented infrastructure offers service composition environments the ability to access a KWer's personal information in a consistent manner and thereby offers reusable services that match a KWer's activities.

The remaining paper is structured as follows: We first discuss the core problems underlying for the identified issues of situational applications for personal use. This includes on the one hand the personal information fragmentation in workspaces which prevents efficient use of personal information as well as on the other hand that KWers conduct a common set of supporting activities in their work which leads currently to redundant implementations of supporting business logic. Then we propose a personal information handling infrastructure that facilitates creating user-generated services for personal use. It consists of the three elements of functionally-oriented applications, domain-specific services and a basic personal information management (PIM) system. We then discuss the benefits of this infrastructure and give a summary and an outlook.

## 2 Core Problems of Situational Applications for Personal Use

### 2.1 Personal Information Fragmentation in Workspaces Prevents Efficient Use of Personal Information

The KWer's *personal information is fragmented* across applications in digital workspaces, i.e., the *KWer's personal information is scattered across the desktop and its applications* [Jones, 2008]. Current desktop systems and applications don't represent the personal information cloud in a unified and integrated form as the KWer expects it, i.e., the personal information is fragmented across applications in the workspace. E.g., an email client folder containing emails has for the KWer no visible connection to a file system folder containing documents despite dealing with the same topic.

Workspace applications *managing only a particular type of personal information and locking it in* a proprietary storage causes one major type of personal information fragmentation. Karger mentions that these applications "often store their data in their own particular locations and representations, inaccessible to other applications" [Jones&Teevan, 2007, p. 127]. Karger outlines this type of information fragmentation as paradox as this "information is fragmented by the very tools that have been designed to help us manage it" [Jones&Teevan, 2007, p. 127]. Jones as well sees here "many examples of seemingly avoidable information fragmentation" [Jones, 2008, p. 392].

Another related type of *information fragmentation is produced by the KWer herself*. Due to application information silos, the KWer is forced to fragment the own personal information when keeping it. This is due as each application only manages particular own information types, i.e., information silos, regardless of the KWer's

intent for keeping the information. Bergmann reports this type of personal information fragmentation problem in the context of organizing project-related information, called project fragmentation problem [Bergman et al., 2006]. Figure 1 shows an example of "information related to a chemistry course" which is "fragmented into separate collections" across three applications. The involved applications' ability of managing each only a particular information element type forces the KWer to organize the information in a fragmented way.

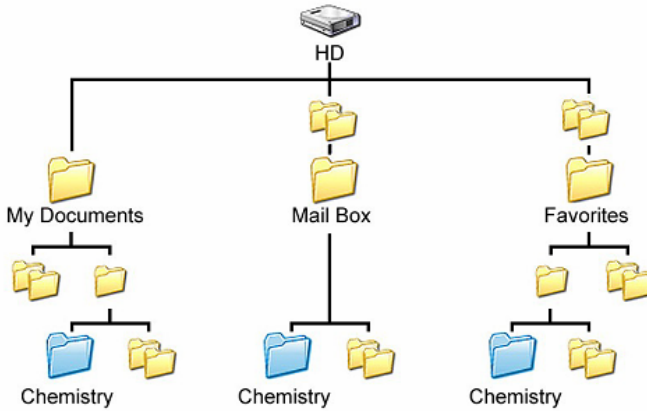


Fig. 1. Personal information fragmentation example – project-related [Bergman et al., 2006]

## 2.2 KWer Conduct a Common Set of Supporting Activities in Their Work

The KWer performs activities in a personal value chain in analogy to an enterprise's value chain [Porter, 1998], see Figure 2. The KWer's activities can be categorized into the two types of personal primary activities and personal support activities.

Personal primary activities are the working activities where the KWer actually works towards achieving the individual goals. They depend largely on the KWer's job role and the KWer's individual skills and capabilities. By executing personal primary activities, the KWer produces the output that she gets rewarded for.

Personal supporting activities enable the KWer to manage herself and things that support her. They are independent of the KWer's job role as they concern each individual KWer. These supporting activities are cross cutting primary in the execution through their support for primary activities. They as well vary in their concrete instantiation from KWer to KWer depending on, e.g., the KWer's job role, gathered experience, felt importance and skill level. For example, KWers with a busy schedule and workload feel a more urgent need to perform efficient task and time management than KWers whose workday leaves sufficiently enough time to cope with the workload. The following list covers the major supporting activities without claiming to be exhaustive.

- Task Management (Task): Plan, structure and prioritize a set of tasks.
- Time Management (Time): Plan and control available time.

- Personal Social Network Management (People): Build personal social network, maintain the network and activate nodes within the network as needed [Fisher&Nardi, 2007, p. 171].
- Meeting Management (Meeting): Prepare, conduct and post-process meetings.
- Information Management (Information): Collect, organize, browse and retrieve information.
- Collaboration Management (People, Information): Communicate and interact with people.

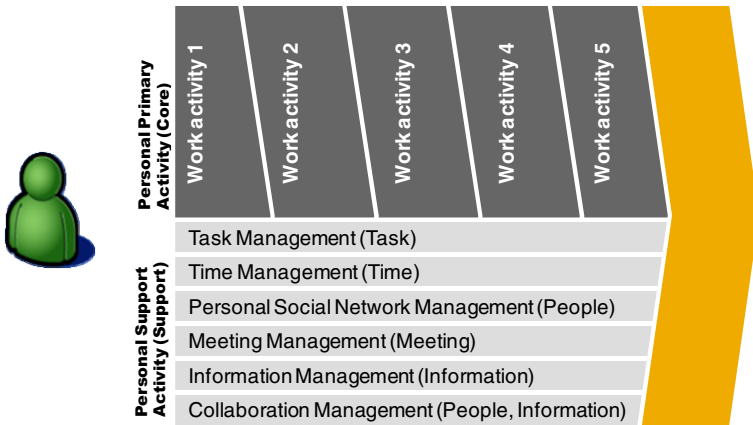


Fig. 2. Personal primary and personal supporting activities conducted by a KWer

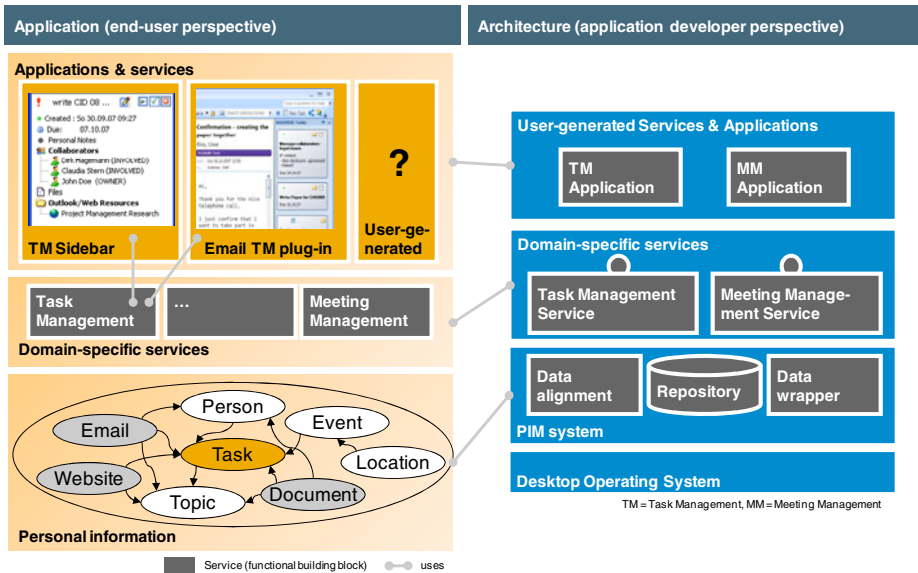
The KWer’s personal activities always include information processing activities. Information processing activities include the KWer’s (re-)acquiring, processing, organizing, and (re-)distributing of information [Jones, 2008, p. 60]. The information processing activities represent an orthogonal, information-centered perspective on the activities that a KWer conducts, compared to this activity-centered perspective.

### 3 Personal Information Handling Infrastructure Facilitates Creating User-Generated Services for Personal Use

We present an *infrastructure* that tackles the identified issues when developing user-generated applications for personal use. The proposed infrastructure consists of three layers [Grebner, 2009]. Figure 3 shows on the left these layers from an end-user perspective and shows on the right an architectural view on the same layers from an application developer’s perspective. In the following we explain each layer in detail.

#### 3.1 Pre-built and User-Generated Applications for Personal Use

First, facing the end-user, *functionally-oriented applications* each focus on supporting a particular type of a KWer’s activity. Figure 3 shows two different user interfaces



**Fig. 3.** Personal information cloud and domain-specific services serve as basis for user generated services and applications

targeting a KWer's task management activities in different situations. On the one hand, a task management (TM) sidebar [Grebner et al., 2008] shows the KWer's tasks and related information in a dedicated application focusing on task management. On the other hand, an email TM plug-in enables the KWer to conduct task management in the email client Microsoft Outlook while focusing on the tasks related to emails.

These two vastly different applications base on the same information, i.e., a task shown in one application can appear in the other application. Each application represents a distinct perspective on the common underlying unified personal information set. These applications can be implemented using multiple programming languages and technologies. The two here presented applications have been built in Java Swing and C#, respectively. Further applications for both the task management and meeting management domain can be found in [Grebner, 2009].

The vision fueled by the results of the here mentioned applications is that the KWer can leverage composition environments for situational applications to create user-generated applications for personal use. This happens by composing an application that addresses the KWer's individual activities and problems using the offered domain-specific services, see the next section for details. For example, a KWer composes a task management application following her own ideas on how to conduct task management. By using the proposed infrastructure with domain-specific services and the basic PIM system the task information produced by the composed application can be integrated with the KWer's personal information cloud and thus neatly works with other existing applications of the KWer.

### 3.2 Domain-Specific Services

Second, a set of *domain-specific services* handles the access to a particular perspective on a KWer's personal information from the application layer to the unified information model. The domain-specific service's functionality provides the needed parts of the personal information to domain-knowledgeable developers. This way, the user experience of each application can be designed solely relying on the domain and completely independent from the underlying (operating) system and associated metaphors and paradigms. Domain-specific services provide controlling business logic for applications on the application layer in the form of supporting functionality and commonly handling the access to a corresponding perspective of the KWer's unified information model. We implemented domain-specific services targeting the two personal supporting activities task management and meeting management, see Figure 3.

There are particular services for defined types of a KWer's activities. The services include functionality needed to support these activities. This includes, but is not limited to, the management of a corresponding perspective on a KWer's unified personal information. Typically focusing on a major entity of the information model, as for example tasks or people, the particular domain-specific service handles this perspective and the major entities and contains business logic needed to process these items. For example, for tasks there are service methods that create a new task, retrieve tasks by different criteria or methods to delegate a task.

The domain-specific service encapsulates the domain-specific business logic in a way that domain-knowledgeable KWers can apply these services without being experts in personal information management. As this domain-specific business logic is encapsulated in a service it can be re-used at every workspace or workspace-level application that needs to access an entity of the information model.

On the implementation side of the domain-specific services, their business logic interfaces core services provided by the basic PIM system such as, e.g., storage and communication. The domain-specific services' business logic interfaces the basic PIM system by invoking its services like, e.g., a PIM service to read and write the unified information model and a data wrapper for handling desktop information elements like files and emails. In our implementation with the Nepomuk semantic desktop as basic PIM system, they interface the core system services like for example PimoService [Sauermann&Klinkigt, 2009] to read and write the unified personal information model and the Aperture DataWrapper [Aduna B.V.&DFKI GmbH, 2005] as indexer for desktop information elements like files and emails.

### 3.3 Basic PIM System

Third, using a *basic PIM system* we can access numerous desktop services and thus re-use desktop functionality without replicating it in each workspace. Here we use the semantic desktop *Nepomuk* [Groza et al., 2007] as example for a service-oriented basic PIM system. The Nepomuk semantic desktop provides a service-oriented desktop architecture and services. The basic PIM system manages the unified personal information model representing the KWer's personal information cloud and offers a PIM Service providing generic read and write functionality for the unified personal information model. Nevertheless, it is user-owned but formally represented using a set

of ontologies. We use a formalized representation of this model, the Personal Information Model Ontology (PIMO) [Sauer mann et al., 2007] to enable machine processing on it by the domain-specific services. It features a unified representation through the RDF representation and it is integrated along the KWer's personal things. This represents the KWer's mental model, i.e., the personal, subjective view on the world. Furthermore, it encapsulates core operating functionality in services like, e.g., handling desktop information objects like for example, emails, websites and files.

## 4 Discussion of Benefits

Through using the proposed infrastructure, developers and end-users in a role to compose an application gain a number of benefits. First, it significantly reduces the complexity of implementing applications.

*Offering a domain-specific interface* reduces the complexity to build an application by opening the development of personal activity support applications to domain-knowledgeable developers while omitting the need for deep personal information management expertise. The domain-specific interface of the domain-specific services allows domain-knowledgeable developers to efficiently handle unified personal information. A domain-specific interface hides the underlying personal information management technology details. In comparison to a generic personal information management interface this saves domain-knowledgeable developers getting into the internals of personal information management and related technology.

*A common PIM system infrastructure* reduces the complexity to build a personal activity support applications by replacing the need for applications and the domain-specific services to each redundantly implement a proprietary stack for integrating desktop information objects. The common PIM system infrastructure underlies the applications and the domain-specific services. The PIM system, i.e., the semantic desktop, provides services and models to manage existing desktop information elements and to integrate them with the unified information model. This means that the PIM system infrastructure is re-usable instead of each application building an own management stack. This replaces the need for applications and the domain-specific services to each redundantly implement a proprietary stack for integrating desktop information objects, significantly reducing the complexity of implementing applications.

Second, using this reference architecture significantly improves the capabilities that developers can implement into personal activity support applications.

*An increased flexibility for building applications* can be realized as existing business logic for certain KWer activities can be re-used and only the user interface needs to be re-developed and designed, for example using the described composition environments for situational applications. The domain-specific services provide a number of services to realize the functionality needed for supporting a KWer's activities like, e.g., task delegation to send off a task to other people. As long as these services are available, developers can quicker develop new applications featuring an improved user interaction compared to when they need to re-develop the full personal information management stack. The service-oriented desktop architecture of the Nepomuk semantic desktop PIM system facilitates this re-use further as it keeps the available services in a service registry allowing other applications to discover and use these services. This enables multiple

applications that are implemented in different programming languages to invoke the provided services. By invoking services, multiple applications can, e.g., query and write to the common unified personal information model.

## 5 Summary and Outlook

To summarize, the here presented infrastructure offers service composition environments the ability to access a KWer's personal information in a consistent manner and thereby offers re-usable services that match a KWer's activities on a domain-specific level. This enables KWers to compose applications and services for personal use. Alternatively, it as well eases the work of developers developing respective components. At hand of task and meeting management services and two example applications we showed how tailored application "hats" can re-use common business logic to offer services and applications in the KWer's personal support domain.

Due to the service-based nature of this infrastructure, we want to enable service composition environments to consume the domain-specific services and thus offer application support for the personal use domain of KWers. The here presented principle of a personal information cloud and domain-specific services has been implemented for personal information on the desktop. The KWer's personal information is thus securely stored in an information repository that is accessible only to the one individual KWer owning it. To enable these services for use by the mentioned service composition environments, the KWer's personal information cloud, the used basic PIM system and the domain-specific services will be ported into web-based services and a multi-user environment while maintaining the privacy of the KWer's personal information cloud. In addition, the next implementation version will incorporate as well personal information already managed by web-based services, such as for example social networks, online storage providers as well as online email providers.

## References

- Aduna, B.V.: DFKI GmbH. Aperture (2005), <http://aperture.sourceforge.net> (Accessed 2009.10.31)
- Bergman, O., Beyth-Marom, R., Nachmias, R.: The project fragmentation problem in personal information management. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 271–274. ACM, New York (2006)
- Dragan, L., Möller, K., Handschuh, S., Ambrus, O., Trüg, S.: Converging Web and Desktop Data with Konduit. In: Proc. of Scripting and Development for the Semantic Web Workshop (2009), <http://CEUR-WS.org/Vol-449/Paper4.pdf> (Accessed 2009.10.31)
- Fisher, D., Nardi, B.: Soylent and ContactMap: Tools for Constructing the Social Workscape. In: Kaptelinin, V., Czerwinski, M. (eds.) Beyond the desktop metaphor: designing integrated digital work environments, pp. 171–190. MIT Press, Cambridge (2007)
- Grebner, O.: Using Unified Personal Information in Workspaces. PhD thesis. (2009), <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000012169> (Accessed 2009.10.31)

- Grebner, O., Ong, E., Riss, U.: KASIMIR - Work process embedded task management leveraging the Semantic Desktop. In: Bichler, M., Hess, T., Kremer, H., Matthes, U.L.F., Picot, A., Speitkamp, B., Wolf, P. (eds.) *Multikonferenz Wirtschaftsinformatik*, pp. 715–726. GITO-Verlag, Berlin (2008)
- Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R.: The Nepomuk project-on the way to the social semantic desktop. In: *Proceedings of I-Semantics*, vol. 7, pp. S201–S211 (2007)
- Jones, W., Teevan, J.: *Personal Information Management*. University of Washington Press (2007)
- Jones, W.: *Keeping Found Things Found*. Elsevier Inc., Amsterdam (2008)
- Moran, T.P., Zhai, S.: Beyond the Desktop Metaphor in Seven Dimensions. In: Kaptelinin, V., Czerwinski, M. (eds.) *Beyond the desktop metaphor: designing integrated digital work environments*, pp. 335–354. MIT Press, Cambridge (2007)
- Porter, M.E.: *Competitive advantage: Creating and sustaining superior performance*. Free Press (1998)
- Ravasio, P., Tschertner, V.: Users' Theories of the Desktop Metaphor, or Why We Should Seek Metaphor-Free Interfaces. In: Kaptelinin, V., Czerwinski, M. (eds.) *Beyond the desktop metaphor: designing integrated digital work environments*, pp. 265–294. MIT Press, Cambridge (2007)
- Ravasio, P., Schär, S.G., Krueger, H.: In pursuit of desktop evolution: User problems and practices with modern desktop systems. *ACM Trans. Comput.-Hum. Interact.* 11(2), 156–180 (2004)
- Sauermann, L., Klinkigt, M.: PIMO Service (2009), <http://dev.nepomuk.semanticdesktop.org/wiki/PimoService> (Accessed 2009.10.31)
- Sauermann, L., Van Elst, L., Dengel, A.: Pimo-a framework for representing personal information models. In: *Proceedings of I-Semantics*, pp. 270–277 (2007)
- Yahoo! Inc. Yahoo Pipes (2009), <http://pipes.yahoo.com/pipes/> (Accessed 2009.10.31)



# Towards Ontology Matching for Intelligent Gadgets

Oszkar Ambrus, Knud Möller, and Siegfried Handschuh

Digital Enterprise Research Institute (DERI)  
National University of Ireland, Galway (NUIG)  
{oszkar.ambrus, knud.moeller, siegfried.handschuh}@deri.org

**Abstract.** The FAST gadget development environment allows users to graphically compose intelligent, i.e., semantically annotated gadgets from predefined building blocks and deploy them on various mashup platforms, thus enabling the interconnection of different systems and services. In an environment where different parties use different ontologies to describe such building blocks, ontology matching is crucial. This paper discusses first steps in our effort to integrate ontology matching in an end-user-oriented environment such as FAST. We evaluate a number of tools and approaches for solving different levels of complexity in ontology matching and define the direction of integrating ontology matching into FAST.

**Keywords:** ontology matching, end-user, mashups, gadgets, widgets.

## 1 Introduction

FAST (Fast and Advanced Storyboard Tools) [1] is a visual programming platform allowing business users to build enterprise-class mashups, employing various underlying services and generating new ones. Resources in FAST are described semantically using different ontologies and vocabularies, and can therefore be combined to what we can call “intelligent gadgets”. These ontologies come from different parties, but will sometimes cover the same domain, making ontology matching necessary. The paper reports on early steps on how to integrate existing ontology matching approaches into an end-user-targeted tool such as the FAST platform. The focus of the paper is not on new methods and algorithms for ontology matching, but rather a survey and application of existing ones to our use case of ontology-matching for end-users. We will present automated solutions for simple cases as well as identify more problematic cases which require manual work, in which we want to support the ontology engineer.

In the remainder of this section we give an introduction to the FAST project, and a brief overview of ontology matching. In Sect. 2 we detail the requirements for ontology matching in FAST. Sect. 3 describes the alignment tool used and the rationale behind our choice. In Sect. 4 we describe a problem scenario that we want to give a solution for in FAST, which is the basis of this work, and in Sect. 5 we present the ontologies used for the different web services involved in the scenario. Finally, in Sect. 6 we present the testing procedure and the results, based on which we draw the conclusions in Sect. 7 and define future directions.

## 1.1 FAST

The main goal of the FAST Project [1] is to develop a web-based user-centric visual programming environment allowing users to build enterprise mashups [2] using so-called *gadgets* (see Fig. 1 for a screenshot exemplifying the composition of so-called *screens* to build such a gadget). The motivation behind FAST is to allow non-technical users to be involved in the development process of software applications based on their ad-hoc needs.

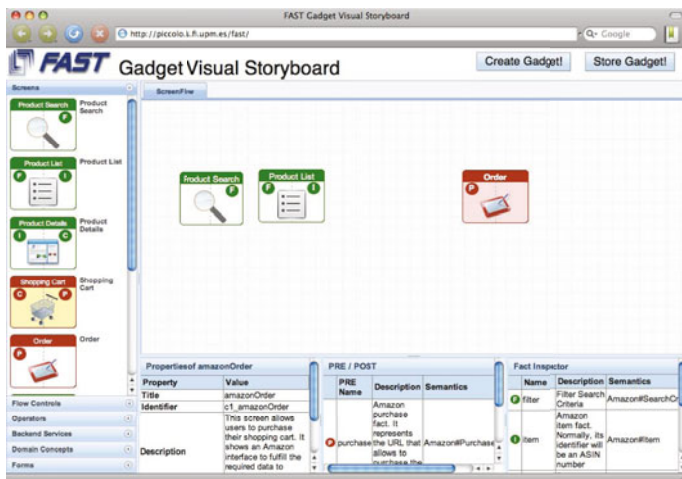


Fig. 1. Composing an intelligent gadget from screens in FAST

The relevant components of the enterprise mashup paradigm are *resources*, which are in the focus of this paper, representing all building-blocks of gadgets (such as the screens in Fig. 1), *gadgets* that provide a graphical interface and interaction mechanisms abstracting from the complexity of the underlying backend, and, *mashups* created by users using the provided gadgets.

## 1.2 Ontology Matching

FAST uses ontologies to conceptualise the underlying resources used by the different components. Ontologies embody the fundamental vehicle for conceptualising data on semantic systems; they describe the context and semantic background of data that should be known to all agents using it [3]. However, different ontologies are often used to describe the same domain or cover the same scenario. This is also true for FAST, where gadget building blocks can originate from different providers, who might use different ontologies to describe them, so the task of ontology matching is critical for interoperability.

Given two ontologies  $O$  and  $O'$  that need to be mapped to each other, we adopt the definition given in [4]: an ontology mapping element is a 5-tuple

$\langle id, e, e', n, R \rangle$ , where  $id$  is a unique identifier, identifying the mapping element,  $e$  and  $e'$  are entities (formulas, terms, classes, individuals) of the first and second ontology, respectively,  $n$  is a confidence measure holding the correspondence value between  $e$  and  $e'$ ,  $R$  is the correspondence relation holding between  $e$  and  $e'$  (e.g., **equivalence** (=) or **more general** ( $\sqsupseteq$ )). The alignment operation determines the mapping  $M'$  for a pair of ontologies  $O$  and  $O'$ . The alignment process can be extended by parameters, such as an input mapping, weights and thresholds and other external resources (dictionaries, thesauri, etc.). Different levels of mappings are defined: (a) A *level 0* mapping [5] is a set of the above mapping elements, when the entities are discreet (defined by URIs). See List. 2 for an example. (b) A *level 1* mapping is a slight refinement of level 0, replacing pairs of elements with pairs of sets of elements. (c) A *level 2* mapping can be more complex and defines correspondences in first order logic. It uses the ontology mapping language described in [6]. It can describe complex correspondences, such as the one detailed in Sect. 6.2.

## 2 Ontology Matching in FAST

The gadget life cycle in FAST has several phases and roles associated, as detailed in [1]. Here, we list the ones relevant for the ontology matching tasks, in decreasing order of the measure in which knowledge about ontologies is required. Note that several roles can be played by the same actor. (i) The *ontology engineer* creates the ontologies used to annotate services and data. This role also includes the process of ontology matching, either automated or manually, determining if the alignment is feasible and creating so-called *matching operator* building blocks, which are basic elements of the FAST screen building. The *resource developer* then uses these ontologies to annotate resources created in FAST. (ii) Ontology matching is needed by the *screen developer* at the design-time of a screen (a visual building block of a gadget). Screen developers have a dedicated UI component for building screens, in which they can use the matching operators to combine components annotated with different ontologies. No actual matching needs to be performed in this phase, but rather the possibility of matching needs to be determined (i.e., *can* two screens A and B be combined?). (iii) The *gadget developer* combines screens to screen-flows and gadgets, and only uses ontology matching implicitly. (iv) The *end-user* uses the final deployed gadget at run-time, but is unaware of the underlying resources and ontologies or the matching process. Only at run-time the actual mapping of instance data has to be performed. In this paper, we mainly consider the first two cases (i.e. ontology engineering and screen development).

## 3 Alignment Tool

An *ontology mapping* is a declarative specification of the semantic overlap between two ontologies [7], being represented as a set of axioms in a *mapping language*. It is the result of the *ontology alignment process* having has three

main phases: (1) discovering the mapping, (2) representing the mapping and (3) exploiting the mapping. Thus we need a tool to assist the ontology engineer in the alignment process. Based on the description given in Sect. 2 we identify the following requirements for an alignment tool, that we will take as the basis for an ontology matching component in FAST: (i) all three phases of the process need to be accessible, (ii) matching of OWL and RDFS ontologies must be supported, (iii) the tool should perform the alignment process with little or no user interference, since FAST is end-user oriented, (iv) the tool needs to be open source, allowing it to be integrated into the free and open FAST platform and (v) it should be well documented.

Based on these requirements, we compared three different tools. *MAFRA* [8] and *RDFT* [9] were found to be unsuitable for either being too restricted to their environment or being no longer available. *Alignment API* [5] is the tool best matching our requirements, satisfying all the desired conditions. It is still under active development, provides an API and its implementation, is open source (GPLv2 or above) and written in Java, providing an easy way to embed it into other programs. Alignment API can be extended by other representations and matching algorithms, it can be invoked through the command line interface (thus working without user interference) or one of the two available GUI implementations, or it can be exposed as an HTTP server. The tool allows for testing different alignment methods and can generate evaluation results based on a reference alignment. Alignment API can generate the mapping results in XSLT, therefore providing an easy way to integrate them into other systems.

## 4 Scenario Description

In our evaluation scenario, which is taken from the e-commerce domain, a user needs to build a gadget which combines data from major e-commerce services, allowing to aggregate item lists from all of them in a combined interface. As examples in our scenario, we consider the two most popular online shopping websites<sup>1</sup>, Amazon and eBay, along with the BestBuy site. The latter is an interesting case, because it exposes its data in RDF using the GoodRelations (GR) ontology [10], which has recently gained a lot of popularity. It is therefore one of the first major e-commerce sites to provide semantic metadata.

Figure 2 illustrates our scenario. There are three retrieval components that wrap the different e-commerce sites and provide data according to three different ontologies: the GR ontology, the Amazon ontology (A) and the eBay ontology (E). Another component displays GR items for display to the user, but not A or E items. If the gadget designer wants to aggregate data from all three services in the display, there will therefore have to be a mapping present between A and E on the one hand, and GR on the other.

---

<sup>1</sup> <http://alexa.com/topsites/category/Top/Shopping>, checked 01/11/2009

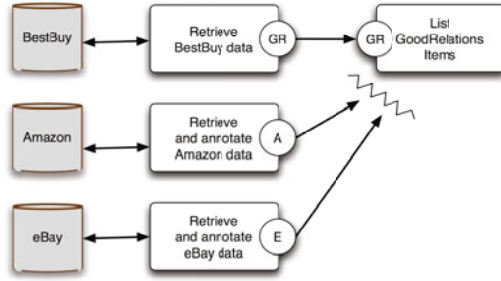


Fig. 2. Three gadget components retrieving incompatible data

## 5 Ontologies

Of the three ontologies used in our evaluation, only *GoodRelations* is a real-world, extensive ontology for e-commerce. The other two, i.e., the Amazon and the eBay ontologies were developed for simulation purposes as simplified versions of what would be used in the real-life scenarios. They were designed to showcase particular features of ontology mapping in our scenario.

*GoodRelations*: This ontology is aimed at annotating so-called “offerings” on the Web, which can be products or services. The ontology features support for ranges of units, measurements, currencies, shipping and payments, common business functions (sell, lease, etc.) and international standards (e.g. ISO 4217) and codes (e.g., EAN) in the field. The main class is *Offering*, which provides a *ProductOrService* with a given *BusinessFunction*. It may be constrained in terms of eligible business partner, countries, quantities, and other properties. It is also described by a given *PriceSpecification*. The super-class for all classes describing products or service types is *ProductOrService*, which is described by its title and description, manufacturer, make and model, etc.

*Amazon Ontology*: We have created a small Amazon ontology based on a subset of the datatypes supported by the web service exposed by Amazon to third-party agents. The ontology describes *Items* based on the *ItemAttributes* description given in the Amazon Product Advertising API documentation<sup>2</sup>. The ontology features three classes for describing a product. Example instance data is given in List. 1. (1) *Item* represents an Amazon item, defined by a title, a manufacturer, a product group (DVD, Book, etc.), an international EAN code, an ASIN (unique Amazon id), an author (for books) and a *ListPrice*. (2) *Company*, described by a legal name, is used for representing the manufacturer of an *Item*. (3) *ListPrice* has two properties: *hasCurrencyCode*, representing an ISO 4217 currency code (e.g. GBP or EUR), and *hasAmount* representing the price in the given currency.

<sup>2</sup> <http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/>

```

:Item_7590645 a amzn:Item ;
  amzn:hasASIN "B0012YA85A" ;
  amzn:hasManufacturer :Manufacturer_Canon ;
  amzn:hasModel "XSi Kit" ;
  amzn:hasPrice :Price_7590645_1 ;
  amzn:hasProductGroup "Electronics" ;
  amzn:hasTitle "Canon Digital Rebel XSi [...]" .
:Manufacturer_Canon a amzn:Company ;
  amzn:hasLegalName "Canon" .
:Price_7590645_1 a amzn:ListPrice ;
  amzn:hasAmount "575.55" ;
  amzn:hasCurrencyCode "GBP" .

```

**Listing 1.** Simplified Amazon ontology data in N3 notation

*eBay Ontology:* The eBay ontology was created based on the eBay Shopping API<sup>3</sup> and is supposed to annotate data retrieved through the web service described by the API. The ontology features three basic classes, (1) `SimpleItem` represents an eBay `Item`, that is sold by a `SimpleUser`. It is described by a title, a `CurrentPrice` (the highest bid or the fixed selling price), primary category name, manufacturer, model, EAN code, item ID (a unique eBay ID), bid count, end time of bid, country where the item is located, and a product ID (supporting major international product codes — based on the Finding API). (2) The `CurrentPrice` features a `hasAmountType` property (currency code), and a `hasAmount` property (amount of money for a price per unit). (3) `SimpleUser` contains information about eBay users, described by a user ID, about me URL and the seller's positive feedback score. This class will not be used for capturing information on goods for our scenario, but is an essential component of the eBay system, which was the reason for its inclusion in the ontology.

## 6 Testing and Results

We present the approach an ontology engineer has to take to discover and represent ontology mappings, and a means to exploit them after the they have been discovered and appropriately represented.

There is a major paradigm difference between the GR ontology and the other two ontologies (see Sect. 6.2 for details). After some initial testing, we concluded that automatic mapping from GR to A/E using the string-based methods employed by the Alignment API tool was not feasible. Therefore, the following sections report on *automatic mapping* for the A–E pair — which are similar enough to be suitable for level 0 mapping —, and *manual mapping* for the GR–A/E pairs.

### 6.1 Automatic Mapping

For automatic finding of level 0 mappings, we used a simple string distance-based algorithm provided by Alignment API [5], which computes the string distance

<sup>3</sup> <http://developer.ebay.com/DevZone/shopping/docs/CallRef/index.html>

between the names of the entities to find correspondences between them. Four methods have been used for computing the distance: (1) equality, which tests whether the names are identical, (2) Levenshtein distance (number of character operations needed), (3) SMOA distance (which is a specialised distance for matching ontology identifiers) and (4) a Wordnet-based [11] distance using the JWNL library with Wordnet.

The alignment description derived from these methods is given based on a simple vocabulary, containing a pair of ontologies and a set of correspondences, which express relations between entities of the two ontologies. We used the level 0 mapping representation for representing simple mappings, which map discrete entities of the two ontologies. Thus the representation of the correspondences is given with the five elements described (with the `id` being optional), as shown in List. 2. Similar mappings were also used for more complex, manually-created representations (level 2), as detailed in Sect. 6.2.

```
<level_0_mapping> a align : Cell ;
  align : entity1 amzn : hasCurrencyCode ;
  align : entity2 ebay : hasAmountType ;
  align : measure "1.0"^^xsd : float ;
  align : relation "=" .
```

**Listing 2.** Level 0 mapping element example

*Testing Procedure:* For the Amazon–eBay pair we set up a reference alignment, against which the results are evaluated. We then ran the matching process for all for methods: (1) equality, (2) Levenshtein distance with a confidence threshold of 0.33 (meaning that any correspondence having a smaller confidence measure will be excluded), (3) SMOA distance with a threshold of 0.5 and (4) Wordnet distance using a threshold of 0.5. To apply the results, we rendered an XSLT template to transform an example dataset.

*Results:* The results of automatically aligning the Amazon and eBay ontologies were quite favourable. As shown in Tab. 1, we captured the four main parameters used in information retrieval, as described in [12]. These four parameters are used for evaluating the performance of the alignment methods: (1) *Precision*, the fraction of results that are correct — the higher, the better, (2) *Recall*, the ratio of the correct results to the total number of correct correspondences — the higher, the better, (3) *Fallout*, the fraction of incorrect results - the lower the better, and (4) *F-measure*, which measures the overall effectiveness of the retrieval by a harmonic mean of precision and recall — the higher, the better. The first row (reference) shows the reference alignment, which, naturally, has both perfect precision and recall. We can observe what intuition has predicted, namely that pure string equality (equality) is far too simple and irrelevant, by only taking identical labels. By using string distances and giving certain thresholds (Levenshtein and SMOA), we can see that the results are much less precise, but have a better recall, since this allows for entities having similar names to be discovered, at the expense of having quite a few incorrect results

**Table 1.** Alignment results: Precision, Recall, Fallout and F-Measure

	precision	recall	fallout	f-measure
<b>reference</b>	1.00	1.00	0.00	1.00
<b>equality</b>	1.00	0.38	0.00	0.55
<b>SMA</b>	0.43	0.75	0.57	0.55
<b>Levenshtein</b>	0.40	0.75	0.60	0.52
<b>JWNL</b>	0.67	0.75	0.33	0.71

(lower precision); the thresholds allow for low-scored cases to be eliminated, although this results in the exclusion of some correct correspondences. The last column (JWNL) contains the results of the Wordnet-enabled method, which shows quite an improvement (precision of 0.67 and a recall of 0.75), due to the lexical analysis, which performs a much more relevant comparison of strings, giving a high number of correct results. The precision of the JWNL alignment shows only a tiny drop below the recall value, meaning that the number of incorrect correspondences discovered is small, and the main source of error is from the number of correspondences not discovered.

We can deduce that the results provided are satisfactory, even though the methods used were simple, string-based ones, and the process was completely automated without any user input. We are therefore confident that through some user assistance or an initial input alignment the tool can achieve 100% correct results.

## 6.2 Manual Mapping

The GoodRelations (GR) ontology employs a unique paradigm, different from the paradigms of Amazon (A) and eBay (E). In GR everything is centred around an instance of **Offering** and a graph of other instances attached to it, whereas for A (and similarly for E), the main class is **Item**, which holds all relevant properties. In principle, **Item** would correspond to **ProductOrService** in GR, but the properties of the **Item** class are reflected as properties of many different classes in GR.

Though the infeasibility of automating this alignment became obvious, we have represented the alignment in the mapping language supported by the tool, as a level 2 mapping (described in Sect. 1.2). This mapping description can later be used by the run-time gadget code. List. 3 shows an example mapping between two properties of the two ontologies, specifying that the relationship is **Equivalence** with a certainty degree of 1.0. This fragment does not show, but assumes the equivalence correspondence between the classes **Item** and **Offering**, which is a trivial level 0 mapping. This mapping specifies the relation

$$\forall v, z; hasEAN(v, z) \implies \exists x, y; includesObject(v, x) \wedge \\ typeOfGood(x, y) \wedge hasEAN\_UCC\_13(y, z),$$



```

<level_2_mapping> a align: Cell ;
  align: entity1 amzn:hasEAN;
  align: entity2
    [ a align: Property ;
      align: first gr:includesObject;
      align: next gr:hasEAN_UCC_13,
        gr:typeOfGood ];
  align: measure "1.0"^^xsd:float;
  align: relation "Equivalence" .

amzn:hasEAN a align: Property .
gr:hasEAN_UCC_13 a align: Property .
gr:includesObject a align: Relation .
gr:typeOfGood a align: Relation .

```

**Listing 3.** Fragment of the Amazon–GoodRelations mapping

meaning that the `hasEAN` property of  $v$  in the  $A$  ontology corresponds to the `hasEAN_UCC_13` property of the `typeOfGood` of the `includesObject` of  $v$  in GR. The domains and ranges of the properties are inferred, thus it is deduced, that in  $A$   $v$  is of type `Item` and  $z$  is `int`, and in GR  $v$ ,  $x$ ,  $y$  and  $z$  are instances of the classes `Offering`, `TypeAndQuantityNode`, `ProductOrService` and `int`, respectively.

Using this representation, complex correspondences can be modelled, using first order logic constructs.

## 7 Conclusion

The goal of this paper was to explore ontology matching in an environment such as the FAST platform, where building blocks described with different ontologies need to be integrated by end users. To this end, we have identified which roles in the FAST gadget development lifecycle need to be considered, and which kinds of ontology matching problems they might face. Based on an example scenario from the e-commerce domain, which includes real-world ontologies such as GoodRelations, we have evaluated different existing algorithms for level 0 matching problems. A wordnet-based approach for string matching performed best, giving results suitable for semi-automatic level 0 matching, thus enabling non-expert end users to perform such tasks in FAST. For more complex level 2 matching problems, manual definition of matching rules is still necessary. Additionally, we have evaluated three ontology matching tools based on the requirements given by the FAST platform, and established that the Alignment API tool suits our needs best. We use Alignment API both for the (semi-)automatic generation of level 0 matching rules, as well as for the syntactic representation of manually generated level 2 problems. Based on this format, ontology matching rules of all levels can be represented and executed in all relevant components of the FAST architecture.

As future work, we need to evaluate existing or develop new methods to aid users in complex ontology matching problems (level 2). Additionally, we will

integrate ontology matching into the running FAST platform, and evaluate its performance and usability there.

## Acknowledgments

The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Líon-2) and (in part) by the European project FAST No. FP7-ICT-2007-1.2 216048.

## References

1. Hoyer, V., Janner, T., Delchev, I., Lpez, J., Ortega, S., Fernández, R., Möller, K., Rivera, I., Reyes, M., Fradinho, M.: The FAST platform: An open and semantically-enriched platform for designing multi-channel and enterprise-class gadgets. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 316–330. Springer, Heidelberg (2009)
2. Hoyer, V., Stanoesvka-Slabeva, K., Janner, T., Schroth, C.: Enterprise mashups: Design principles towards the long tail of user needs. In: SCC 2008: Proceedings of the 2008 IEEE International Conference on Services Computing, pp. 601–602 (2008)
3. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, The Netherlands (1993)
4. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. *Journal on Data Semantics* 4, 146–171 (2005)
5. Euzenat, J.: An API for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
6. Scharffe, F., de Bruijn, J.: A language to specify mappings between ontologies. In: Proc. of the Internet Based Systems IEEE Conference, SITIS 2005 (2005)
7. de Bruijn, J., Lausen, H.: Web service modeling language (WSML). Member submission, W3C (June 2005)
8. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra — a mapping framework for distributed ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, p. 235. Springer, Heidelberg (2002)
9. Omelayenko, B.: RDFT: A mapping meta-ontology for business integration. In: Proceedings of the Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002), Lyon, France, pp. 76–83 (2002)
10. Hepp, M.: GoodRelations: An ontology for describing products and services offers on the web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 332–347. Springer, Heidelberg (2008)
11. Fellbaum, C., et al.: WordNet: An electronic lexical database. MIT Press, Cambridge (1998)
12. Olson, D., Delen, D.: Advanced data mining techniques. Springer, Heidelberg (2008)

# Author Index

- Addicks, Jan Stefan 62  
Aier, Stephan 35  
Alt, Rainer 145  
Altmann, Jörn 238  
Ambrus, Oszkar 570  
Andrikopoulos, Vasilios 283
- Birkmeier, Dominik 22  
Bos, Rik 48  
Brandic, Ivona 238  
Brinkkemper, Sjaak 48  
Bucchiarone, Antonio 467  
Buckl, Sabine 1
- Cappiello, Cinzia 467  
Carro, Manuel 414  
Casati, Fabio 208  
Chaparadza, Ranganai 303, 405  
Chee, Yi-Min 82  
Chi, Chi-Hung 197  
Clarke, Siobhán 314  
Comuzzi, Marco 187
- D'Andrea, Vincenzo 208  
Daniel, Florian 208  
Dannecker, Lars 520  
Davy, Alan 303  
Dejun, Jiang 197  
Delgado, Andrea 456  
Di Nitto, Elisabetta 467  
Doppstadt, Christian 134  
Dustdar, Schahram 176, 283
- Eberle, Hanna 358  
Ertlmaier, Thomas 370  
Etzion, Opher 370
- García, José M<sup>a</sup> 228  
García-Rodríguez de Guzmán, Ignacio 456  
Gilles, Florian 509  
Gomez, Jose Manuel 269  
Grajzer, Monika 335  
Grebner, Olaf 560
- Handschuh, Siegfried 570  
Hermenegildo, Manuel 414  
Hobson, Stacy 115  
Hoyer, Volker 509
- Ivanović, Dragan 414
- Janner, Till 509
- Kaldanis, Vassilios 405  
Karantonis, George 405  
Karastoyanova, Dimka 395  
Kassem, Gamal 293  
Katsaros, Giannis 405  
Kaufer, Frank 488  
Kazhamiakin, Raman 325, 395, 467  
Kett, Holger 385  
Khan, Ateeq 293  
Kim, Soo Dong 498  
Kloeckner, Sebastian 22  
Kofman, Alexander 370  
Koller, Bastian 165  
Köppen, Veit 293  
Kopp, Oliver 358  
Kotsiopoulos, Ioannis 165  
Kotsokalis, Constantinos 187, 248  
Kottke, Kristian 477  
Kruse, Steffen 62  
Kunkel, Marcel 134
- La, Hyun Jung 498  
Lago, Patricia 445  
Lamersdorf, Winfried 477  
Laredo, Jim 109  
Lausen, Holger 258  
Leitner, Philipp 176  
Leymann, Frank 176, 358, 395  
Liakopoulos, Athanassios 303  
Liu, Dong 269  
Liu, Guohua 96  
Liu, Xi 96  
Liu, Xuan 115  
Lodhi, Azeem 293  
Lopez, Mariana 109  
Loutas, Nikolaos 540

- Mahbub, Khaled 345  
 Manciozzi, Michele 435  
 Martín-Díaz, Octavio 228  
 Matthes, Florian 1  
 Mazza, Valentina 467  
 Mehandjiev, Nikolay 550  
 Meiners, Matthias 477  
 Michlmayr, Anton 176  
 Micsik, András 165  
 Miede, André 72  
 Möller, Knud 570  
 Mora, Juan 165  
 Mos, Adrian 269  
 Müller, Carlos 256  
 Müller, Jürgen 145  
 Muñoz, Henar 165  
 Muth, Marcel 425  
  
 Namoun, Abdallah 550  
 Naumann, Felix 488  
 Nedyalkov, Nedislav 72  
 Nestler, Tobias 520  
  
 Oppenheim, Daniel 82  
  
 Paik, Incheon 218  
 Papazoglou, Mike P. 435  
 Patil, Sameer 115  
 Paulus, Thomas 370  
 Pedrinaci, Carlos 269  
 Perepletchikov, Mikhail 435  
 Peristeras, Vassilios 540  
 Piattini, Mario 456  
 Pierre, Guillaume 197  
 Pires, Luís Ferreira 530  
 Pistore, Marco 325, 395, 467  
 Popescu, Razvan 314  
 Postina, Matthias 62  
 Pursche, Andreas 520  
  
 Qin, Haihuan 96  
 Quaireau, Samuel 269  
  
 Ratakonda, Krishna 82  
 Rathfelder, Christoph 187  
 Razavian, Maryam 445  
 Renner, Thomas 385  
 Repp, Nicolas 72  
 Resinas, Manuel 156  
 Rey, Guillermo Alvaro 269  
  
 Risch, Marcel 238  
 Rodríguez, Carlos 208  
 Rosenberg, Florian 176  
 Ruiz-Cortés, Antonio 156, 228  
 Ruiz, Francisco 456  
 Ryan, Caspar 435  
  
 Saake, Gunter 293  
 Schelp, Joachim 35  
 Schill, Alexander 123  
 Schipper, Jurjen 48  
 Schuller, Dieter 72  
 Schweda, Christian M. 1  
 Schwind, Michael 134  
 Silva, Eduardo 530  
 Silveira, Patrícia 208  
 Springer, Thomas 123  
 Staikopoulos, Athanasios 314  
 Stanoevska-Slabeva, Katarina 509  
 Steffens, Ulrike 62  
 Steinmetz, Nathalie 258  
 Steinmetz, Ralf 72  
 Stelzer, Dirk 12  
 Stollberg, Michael 425  
 Su, Jianwen 96  
  
 Taheri, Zouhair 208  
 Takada, Haruhiko 218  
 Tarabanis, Konstantinos 540  
 Tcholtchev, Nikolay 335  
 Theilmann, Wolfgang 187  
 Toro, Miguel 228  
 Treiber, Martin 283  
 Trigos, Edmundo David 123  
 Tröger, Ralph 145  
  
 Unger, Tobias 358  
  
 van den Heuvel, Willem-Jan 435  
 van Sinderen, Marten 530  
 van Steenberghe, Marlies 48  
 Vaudaux-Ruth, Guillaume 269  
 Vidalenc, Bruno 335  
 Vidačković, Krešimir 385  
 Vogel, Tobias 488  
 von Ammon, Rainer 370  
 Vukovic, Maja 109  
  
 Wajid, Usman 550  
 Weiner, Nico 385

- Wetzstein, Branimir 176, 395  
Winkler, Matthias 123  
Winkler, Ulrich 187, 248  
Worledge, Claire 208
- Yan, Zhimin 96
- Zacco, Gabriele 187  
Zafeiropoulos, Anastasios 303  
Zaplata, Sonja 477  
Zeier, Alexander 145  
Zengin, Asli 325  
Zhang, Liang 96  
Zisman, Andrea 345