Gesellschaft für Informatik e.V. (GI)

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

Broken down into
- seminars
- proceedings
- dissertations
- thematics

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

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-628-2

EMISA 2014 is the sixth international workshop in a series that provides a key forum for researchers and practitioners in the fields of enterprise modelling and the design of information system (IS) architectures. The workshop series emphasizes a holistic view on these fields, fostering integrated approaches that address and relate business processes, business people and information technology. EMISA'14 will provide an international forum to explore new avenues in enterprise modelling and the design of IS architectures by combining the contributions of different schools of Information Systems, Business Informatics, and Computer Science.

F. Feltz, B. Mutschler, B. Otjacques (Eds.): EMISA 2014

234

GI-Edition

Lecture Notes in Informatics

**Fernand Feltz,
Bela Mutschler,
Benoît Otjacques (Eds.)**

Enterprise Modelling and Information Systems Architectures (EMISA 2014)

**Luxembourg
September 25-26, 2014**

Proceedings

Fernand Feltz, Bela Mutschler, Benoît Otjacques (Eds.)

Enterprise Modelling and Information Systems Architectures (EMISA 2014)

Sixth International Workshop on Enterprise Modelling and Information Systems Architectures

September 25 - 26, 2014
Luxembourg

Gesellschaft für Informatik e.V. (GI)

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

Volume P-234

ISBN 978-3-88579-628-2
ISSN 1617-5468

Volume Editors
Fernand Feltz
Benoît Otjacques
Centre de Recherche Public - Gabriel Lippmann
ISC Department
4422 Belvaux, Luxembourg
E-Mail: { feltz | otjacque } @ lippmann.lu
Bela Mutschler
Hochschule Ravensburg-Weingarten
Fakultät Elektrotechnik und Informatik
88241 Weingarten, Germany
E-Mail: bela.mutschler@hs-weingarten.de

printed by Köllen Druck+Verlag GmbH, Bonn

Organizer
Centre de Recherche Public - Gabriel Lippmann
ISC Department
4422 Belvaux, Luxembourg

The workshop is jointly organised by the Centre de Recherche Public – Gabriel Lippmann (Luxembourg) and the GI Special Interest Group on Design Methods for Information Systems (GI-SIG EMISA)

Programme Co-Chairs
Fernand Feltz (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)
Bela Mutschler (University of Applied Sciences Ravensburg-Weingarten)
Benoît Otjacques (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)

Programme Committee
Witold Abramowicz (University of Poznan)
Antonia Albani (University of St.Gallen)
Thomas Allweyer (University of Applied Sciences Kaiserslautern)
Colin Atkinson (University of Mannheim)
Lars Bækgaard (University of Aarhus)
Ulrike Baumoel (University of Hagen)
Khalid Benali (University of Lorraine)
Martin Bertram (Commerzbank, Frankfurt)
Patrick Delfmann (European Research Center for Information Systems (ERCIS))
Jörg Desel (University of Hagen)
Eric Dubois (Centre de Recherche Public - Henri Tudor, Luxembourg)
Thomas Engel (University of Luxembourg)
Werner Esswein (Technical University of Dresden)
Ulrich Frank (University of Duisburg-Essen)
Andreas Gadatsch (University of Applied Sciences Bonn-Rhein-Sieg)
Norbert Gronau (University of Potsdam)
Heinrich Jasper (Technical University of Freiberg)
Reinhard Jung (University of St.Gallen)
Dimitris Karagiannis (University of Vienna)
Stefan Klink (University of Karlsruhe)
Ralf Klischewski (German University of Kairo)
Horst Kremers (CODATA, Germany)
John Krogstie (University of Trondheim)
Susanne Leist (University of Regensburg)
Heinrich C. Mayr (Alpen-Adria-University of Klagenfurt)
Jan Mendling (University of Vienna)
Mirjam Minor (University of Frankfurt)
Bernd Müller (University of Applied Sciences Wolfenbüttel)
Volker Nissen (Technical University of Ilmenau)
Markus Nüttgens (University of Hamburg)
Andreas Oberweis (Karlsruhe Institute of Technology)

Sven Overhage (University of Augsburg)
Hansjuergen Paul (Institute for Work and Technology)
Erik Proper (Centre de Recherche Public - Henri Tudor, Luxembourg)
Michael Rebstock (University of Applied Sciences Darmstadt)
Manfred Reichert (University of Ulm)
Stefanie Rinderle-Ma (University of Vienna)
Peter Rittgen (University of Boras)
Michael Rosemann (Queensland University of Technology)
Matti Rossi (Aalto University)
Frank Rump (University of Applied Sciences Emden/Leer)
Gunter Saake (University of Magdeburg)
Eike Schallehn (University of Magdeburg)
Carlo Simon (Provadis University of Applied Sciences)
Elmar J. Sinz (University of Bamberg)
Stefan Strecker (University of Hagen)
Oliver Thomas (University of Osnabrueck)
Klaus Turowski (University of Magdeburg)
Marten van Sinderen (University of Twente)
Gottfried Vossen (University of Münster)
Barbara Weber (University of Innsbruck)
Hans Weigand (University of Tilburg)
Mathias Weske (University of Potsdam)

Organising Committee

Patrik Hitzelberger (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)
Guy Simon (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)

Doctoral Workshop Organisation

Mohammad Ghoniem (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)
Thomas Tamisier (Centre de Recherche Public - Gabriel Lippmann, Luxembourg)

Preface

At the beginning of the 3rd millennium, digital Information Systems (IS) rule our economies, and have reshaped our social and everyday life, especially with the advent of the Internet and the cheap and ubiquitous access to its countless services – services that require well-designed, cost-efficient IS at the provider side.

EMISA has been focusing very successfully on IS design for 35 years now. This year's anniversary workshop will be an excellent opportunity to draw a balance on what has been achieved (and not), and what is state-of-the-art in the domain of enterprise modelling and IS architectures. Based on this "where are we?" consideration, the question of "where do we go?" will be equally interesting. What are current and foreseeable future topics and trends in IS research, and what is the impact on modelling? Considering on-going technology changes like mobile and cloud computing, and still evolving new methodological approaches like service oriented and model-driven architectures, the main question is: Do we have the right methods and tools to design the new and emerging systems?

EMISA 2014 is the 6th International Workshop in a series that provides a key forum for researchers and practitioners in the fields of enterprise modelling and the design of IS architectures. The workshop series emphasizes a holistic view on these fields, fostering integrated approaches that address and relate business processes, business people and information technology. It provides an international forum to explore new avenues in enterprise modelling and the design of IS architectures by combining the contributions of different schools of Information Systems, Business Informatics, and Computer Science. have asked for contributions covering the following topics:

- Enterprise modelling: languages, methods, tools
- Patterns for enterprise modelling and IS architectures
- Model life cycle management and model evolution
- Model configuration and management of model variants
- Model quality: metrics, case studies, experiments
- Process modelling and process-aware IS
- Collaborative enterprise modelling
- Model-driven architectures and model-driven IS development
- Component- and service-oriented software architectures
- Service engineering and evolution
- Service composition, orchestration and choreography
- Complex event processing and event-driven architectures
- Human factors in enterprise modelling and IS
- Modelling social information and enterprise innovation networks
- Cloud computing infrastructures and their influence on IS engineering
- Security and trust aspects
- Mobile enterprise services
- Individual-based IS strategies
- Software product line architectures and modelling
- Visual aspects of modelling and modelling languages

- IS for large scale data and related design questions

We invited papers that outline research in progress as well as completed research papers. Submitted papers were reviewed by at least three members of the program committee, and were evaluated on the basis of significance, originality, technical quality, and exposition.

The Workshop Papers

In 2014, nine papers were accepted for presentation at the workshop. The selected articles provide a snapshot of current examples for how IS research can be conducted, and what insights such research can uncover.

[Session 1: Enterprise Modelling]

In their position paper "Towards An Analysis Driven Approach for Adapting Enterprise Architecture Languages", De Kinderen and Ma argue for an approach for assisting language engineers in adapting enterprise architecture languages in a controlled manner.

The paper "Outlining a Graphical Model Query Approach Based on Graph Matching" by Breuker et. al outlines a graphical model query approach based on graph matching. It consists of a graphical query specification language and a matching algorithm based on graph matching that takes the query as input and returns all matches found in a model to be searched. The graphical query specification language can be used to draw model queries much like a model would be constructed.

[Session 2: Process Modelling]

The paper "Suggested Guidelines for choosing an adequate Level of Detail" by Nissen et. al deals with the challenge to determine an adequate level of detail in process modelling. In literature, only few and rather unspecific recommendations exist how to solve this problem. In their paper, the authors look at which measurable factors influence the adequate detail level, and on this basis make proposals for guidelines how it can be determined in a specific application situation.

The paper "Designing and Implementing a Framework for Event-based Predictive Modelling of Business Processes" by Becker et. al deals with predictive modelling techniques to event data collected during the execution of business processes. Specifically, the paper presents a framework developed to support real-time prediction for business processes. After fitting a probabilistic model to historical event data, the framework can predict how running process instances will behave in the near future, based on the behavior seen so far.

The paper "BPMN Extension for Business Process Monitoring" by Baumgraß et. al suggests an extension to BPMN, which implements the connection between process models and events. Bridging this gap is an important challenge. Generally, the execution of business processes generates a lot of data, which can be utilized for process monitoring and analysis. In manual executing business process environments, however, i.e., in environments not driven by a process engine, the correlation of occurring events to activities of the corresponding process model is far from being trivial. Typically, process event monitoring points (PEMPs) are utilized to specify the locations, where particular events are expected, in the process model at design-time. Therewith, process execution information can be assigned to a process during run-time.

[Session 3: Process Management Technology]

The paper “Towards Schema Evolution in Object-aware Process Management Systems” by Chiao et. al presents fundamental requirements for enabling schema evolution in the context of object-aware processes. These requirements are then discussed along PHILharmonicFlows, a framework that targets at comprehensive support of object-aware processes.

In her paper “On the Usability of Business Process Modelling Tools – a Review and Research Agenda” Maria Shitkova picks up the challenge to select a business process modelling (BPM) tool with respect to the aspect of usability. She conducts a literature review to find out the current state of research on the usability in the BPM area. The results of the literature review show, that although a number of research papers mention the importance of usability for BPM, real usability evaluation studies have rarely been undertaken. Based on the results of the literature analysis the possible research directions in the field of usability of BPM tools are suggested.

[Session 4: Process Implementation]

The paper “Visual Analytics for Supporting Manufacturers and Distributors in Online Sales” by Parisot et. al presents basic concepts of OPTOSA, a Visual Analytics solution for the optimization of online sales, designed to support manufacturers in all phases of the online sales process from the product specification to the price fixing and more. OPTOSA combines a data processing module that builds and constantly updates operational knowledge related to sales positioning with a decision assistant that uses relevant aspects of the knowledge for helping the tasks of the different teams along the integrated chain of the sales.

Finally, in their paper “Business Process as a Service – Status and Architecture” Barton and Seel deal with business process as a service (BPaaS) as next level of abstraction. Specifically, they present a literature analysis of the current state-of-the-art in BPaaS. In order to investigate how a process can be built on top of a cloud service, a prototpye of an external application is presented, which is built on top of a cloud service using a RESTful API.

We hope you find these papers stimulating and the presentations interesting. We would like to thank the authors for their efforts, and also the program committee in dedicating their time to evaluating and selecting these papers.

Luxembourg, September 2014
Fernand Feltz, Bela Mutschler, Benoît Otjacques

Directory

Process Implementation

Keynotes

Flexibility and Evolution in Process-Aware Information Systems: All Problems Solved?

Stefanie Rinderle-Ma

University of Vienna
Faculty of Computer Science
Research Group Workflow Systems and Technology
stefanie.rinderle-ma@univie.ac.at

Flexibility and evolution in Process-Aware Information Systems (PAIS) have been intensively investigated for almost two decades and mature solutions [RW12], academic prototypes, e.g., the CPEE [MRM14], and even commercial systems, e.g., AristaFlow [LKRD10] have been developed. Starting from this statement, one could ask the following questions:

- Are there still open challenges and questions?
- Is the adoption of flexible PAIS still behind expectations in practice? And if yes, why is this?

The talk tries to answer these questions along the following building blocks:

1. *Current situation and state of the art.* As surveys [RRD04a, SMR$^+$08] and books [RW12] show, flexibility and evolution in PAIS cover several dimensions ranging from design time flexibility (by underspecification or based on declarative models), runtime flexibility where we can distinguish between "foreseen" exceptions (to be dealt with by compensation or rollback) and "unforeseen" exceptions (dealt with by, e.g., ad-hoc changes of single process instances) to process evolution (meaning the migration of running process instances after changing the process schema). In addition, different kinds of flexibility might arise in interplay [RRD04b]. But not only process models and instances might be subject to change, also other aspects of the PAIS can undergo adaptations such as the organizational structures and access rules [RR07].

2. *Challenges and requirements from practical projects.* Insights from developing flexible process technology for the manufacturing domain (cf. ADVENTURE[1] project), the care domain (cf. ACaPlan[2] project), and collaborative process scenarios (cf.

[1] `http://www.fp7-adventure.eu/`
[2] `http://cs.univie.ac.at/project/acaplan`

3. *Challenges and directions in research and technology transfer.* One important conclusion that hence can be drawn is that flexibility and evolution in PAIS cannot be considered in isolation. This insight has been already gained when stating that different aspects of the PAIS might be subject to changes and changing one aspect might have more or less severe effects on the other aspects as well [RR07]. Specifically, if we understand flexibility in PAIS as a non-functional requirement, it cannot be considered in isolation from other non-functional requirements such as compliance and security, interoperability, or usability. Figure 1 sketches some of these requirements that might coincide with flexibility in PAIS. It it is obvious, for example, that without providing users with some understanding of what a change means and what effects it might have, the adoption of flexible process technology might be low [KWRM13, RWRW05]. Moreover, violating existing compliance or security requirements by changing a process model or instance is not constructive as well [LRM14]. Finally, providing flexibility only for centralized process scenarios ("in-house processes") is not enough. In turn, interoperable process scenarios between different partners or organizations can be also subject to change and it becomes even more important to be able to control the change effects potentially spreading over the collaboration [FRMR12].

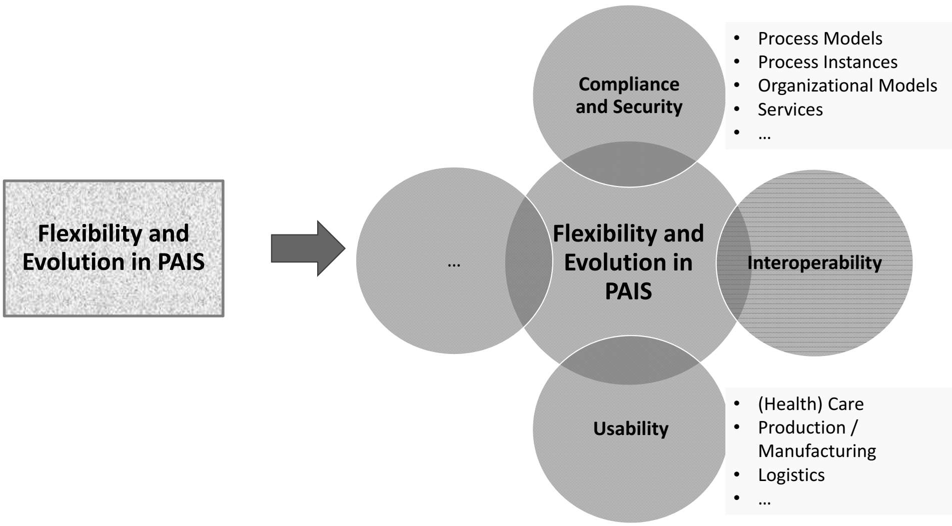

Figure 1: Flexibility in PAIS: Requirements, Aspects, and Applications

In summary, the talk will raise the claim that flexibility and evolution in PAIS are still "en vogue", i.e.., crucial in practical applications and still posing many challenges questions

[3] http://www.wst.univie.ac.at/communities/c3pro/

and research directions, particularly, at the interfaces and combinations of different aspects and requirements.

References

[FRMR12] Walid Fdhila, Stefanie Rinderle-Ma, and Manfred Reichert. Change propagation in collaborative processes scenarios. In *CollaborateCom*, 2012.

[KWRM13] Simone Kriglstein, Günter Wallner, and Stefanie Rinderle-Ma. A visualization approach for difference analysis of process models and instance traffic. In *Business Process Management*, pages 219–226. 2013.

[LKRD10] Andreas Lanz, Ulrich Kreher, Manfred Reichert, and Peter Dadam. Enabling Process Support for Advanced Applications with the AristaFlow BPM Suite. In *Business Process Management Demos*, 2010.

[LRM14] Maria Leitner and Stefanie Rinderle-Ma. A systematic review on security in Process-Aware Information Systems – Constitution, challenges, and future directions. *Information and Software Technology*, 56(3):273–293, 2014.

[MRM14] Jürgen Mangler and Stefanie Rinderle-Ma. CPEE - Cloud Process Exection Engine. In *Business Process Management Demos*, 2014. (accepted).

[RR07] Stefanie Rinderle and Manfred Reichert. A formal framework for adaptive access control models. pages 82–112, 2007.

[RRD04a] Stefanie Rinderle, Manfred Reichert, and Peter Dadam. Correctness criteria for dynamic changes in workflow systems – a survey. *Data & Knowledge Engineering*, 2004.

[RRD04b] Stefanie Rinderle, Manfred Reichert, and Peter Dadam. Disjoint and overlapping process changes: Challenges, solutions, applications. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 101–120. 2004.

[RW12] Manfred Reichert and Barbara Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, 2012.

[RWRW05] Stefanie Rinderle, Barbara Weber, Manfred Reichert, and Werner Wild. Integrating process learning and process evolution–a semantics based approach. In *Business Process Management*, pages 252–267. 2005.

[SMR+08] Helen Schonenberg, Ronny Mans, Nick Russell, Nataliya Mulyar, and Wil van der Aalst. Process flexibility: A survey of contemporary approaches. pages 16–30. 2008.

On the Role of Process Models in Risk and Disaster Information Management

Horst Kremers

CODATA-Germany
PO Box 200548
13515 Berlin, Germany
office@horst-kremers.de

There is considerable effort to cover the broad variety of types of risk including natural disaster risk, technical disaster risk, biological disaster risk, transport and energy, research and innovation, critical infrastructure protection, cross-border health threats, implications for the health service systems, environmental impact assessment, green infrastructure, integrated coastal management, agriculture, food and nutrition security, water, flood risk management, major industrial accident prevention etc.

The complexity of actors and their information needs in all phases of risk and disaster management (from alarm and first aid through prevention measures) is in due need to be structured not only semantically by appropriate methods of meta-information but at the same time, pragmatics structures (workflows, service orchestration, etc.) need an appropriate corresponding framework of business process models.

Currently, the strategic programs of the Hyogo Framework of Action [HFA2] are under global discussion and the corresponding discussions at EU level (Union Civil Protection Mechanism (2013), Manging Risk to Achieve Resilience (2014)) also clearly indicate on the methodological and technological challenges that are faced in this domain.

An overall strategy for information exchange for and between sectors of information society members (including private business, industry, the health domain, agro- [CL01] and food supply chains, lawyers, chambers of professions, etc. etc.) is not developed.

In the future, data flows should be much more specified on the basis of user decision process requirements engineering, centering on actors' information requests [KRE10].

The development (modeling, specification, and implementation, tests etc.) of information standards in these domains is a process of broad involvement of competences from all actors' domains. "Big Data in RISK" (massive and at the same time structurally very complex and heterogeneous data) is in due need for not just "being available" but instead being interoperable for detailed analysis and decision support in a technically controlled way to allow measures of information services quality and contribute essentially to the requested high level of transparency and accountability.

Compared to the current situation of making data of different type and different analysis available for use in risk and disaster management, it is necessary to anticipate and model the broad effects of risk situations and potential disaster on population and economy in an actor-oriented much more differentiated manner and to support the appropriate alternatives of responses based on extensive informational and communicational principles.

The application of complex process modeling will allow control and analysis of information services for decision making in humanitarian situations in a much more reliable way and especially would lead to document and control alternative fact sets as well as contexts and thus contribute essentially to transperancy and accountability in all phases of the disaster cycle.

References

[CHL01] M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.): IT-Standards in der Agrar- und Ernährungswirtschaft - Fokus: Risiko- und Krisenmangement, 24.-25. Februar 2014 in Bonn GI-Edition - Lecture Notes in Informatics (LNI), P-226

[COM216] The post 2015 Hyogo Framework for Action: Managing risks to achieve resilience, http://ec.europa.eu/echo/files/news/post_hyogo_managing_risks_en.pdf

[HFA2] HYOGO FRAMEWORK FOR ACTION 2005-2015, Building the Resilience of Nations and Communities to Disasters, MID-TERM REVIEW 2010-2011

[KRE10] Horst Kremers, Alberto Susini, Victoria Bakhtina, Nina Frolova, Petr Kubicek, Alexandre Tavares, Els Verfaillie: LNIS Vol. 4, Lecture Notes in Information Sciences, CODATA-Germany 2010, Report pp. 179-190, http://www.codata-germany.org/RISK_2009/2009_Berlin_Risk_Management_Workshop_Report_.pdf

[PRW12] Gertraud Peinel, Thomas Rose, Alexander Wollert: The Myth of Business Process Modelling for Emergency Management Planning, Proceedings of the 9th International ISCRAM Conference – Vancouver, Canada, April 2012, L. Rothkrantz, J. Ristvej and Z. Franco, eds.

Keywords

risk information management, risk information models, risk information interoperability, standards development, risk information processes modelling and applications, services and service composition, natural,technical, chemical risks from local to international level, risk information system structure, components, risk-related databases and information system components, risk and risk-model change in time and space, risk modeling issues for infrastructure (e.g. factories, railways, highways, pipelines, maritime traffic etc.), disaster management and emergency preparedness, prevention, alert, response and mitigation, health and biological risks issues for humans, and the environment, risk communication, decisionmaking, actors, accountability

Enterprise Modelling

Towards An Analysis Driven Approach for Adapting Enterprise Architecture Languages

Sybren de Kinderen[1,3], Qin Ma[2,3]

[1] University of Luxembourg, Luxembourg
[2] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
[3] EE-Team, Luxembourg, Luxembourg*
sybren.dekinderen@uni.lu, qin.ma@tudor.lu

Abstract: Enterprise Architecture (EA) modeling languages are increasingly used for various enterprise wide analyses.

In most cases one needs to adapt EA languages to an appropriate level of detail. However such an adaptation is not straightforward. Language engineers currently deal with analysis driven language adaptation in an ad-hoc manner, adapting languages from scratch. This introduces various problems, such as a tendency to add uninteresting and/or unnecessary details to languages, while important enterprise details are not documented. Moreover, adding detail increases the complexity of languages, which in turn inhibits a language's communication capabilities. Yet experience from practice shows that architects often are communicators, next to analysts. As a result, one needs to find a balance between a model's communication and analysis capabilities.

In this position paper we argue for an approach for assisting language engineers in adapting, in a controlled manner, EA languages for model-driven enterprise analyses. Furthermore, we present the key ingredients of such an approach, and use these as a starting point for a research outlook.

1 Introduction

Enterprise Architecture (EA) is increasingly recognized as a steering instrument that covers the complete business-to-IT stack of an enterprise [AW09, OPW+08, Lea13], interrelating an enterprise's products and services, business processes, IT applications and physical IT infrastructure. By emphasizing such a holistic perspective on an enterprise [Lea13], EA can act as an instrument for various *enterprise-wide analyses* (briefly enterprise analyses) to support decision making, such as the enterprise wide impact of access control concerns [FDP+12], the modifiability of an enterprise-wide IT system [LJH10], cost management [Lea13], and more. Two recent surveys among practitioners [MLM+13, LJJ+06] also show a need from industry for such analyses.

*The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, The University of Luxembourg, Radboud University Nijmegen and HAN University of Applied Sciences (www.ee-team.eu)

EA modeling languages, prominently the Open Group standard ArchiMate [IJLP12, Lea13] provide a model driven approach to capture enterprise architectures. Because of the holistic nature, these languages are often on purpose designed for expressing an enterprise at a high level of abstraction [IJLP12]. As a consequence, resulting EA models of such languages are more used to facilitate communication among stakeholders than to support enterprise analyses [IJLP12, MLM+13].

Yet to do a proper enterprise analysis (be it for cost management, security concerns, or otherwise), we also need *domain-specific details* that provide us with a detailed expression of analysis concerns. Thus, to perform an enterprise-wide analysis we essentially need to perform both (1) inter-layer analysis, whereby models with a high level of abstraction allow us to connect different enterprise layers, e.g., business, application and infrastructure layers as defined in ArchiMate, and (2) intra-layer analysis, whereby detailed domain specific models can express analysis concerns to *a level of detail* that is sufficiently amenable for analysis purposes.

A case supporting this argument is the ArchiMate language. As illustrated by the authors of [Lea13, p. 189 - 219], ArchiMate is enriched with domain specific details to enable enterprise analyses, namely:

- either extra attributes assigned to concepts and relations of ArchiMate to capture measures relevant for analysis, e.g., response time of a service and utilization of a resource for performance analysis [JI09], and e.g., importance of a business process and effectiveness of an information system for portfolio analysis [QSL12] (more syntactical details);
- or more details about the meaning of these attributes and the inter-relation between their values [JI09, QSL12] (more semantical details);
- or a higher degree of formality by translating ArchiMate models into mathematical formalisms to support static impact-of-change analysis [dBBJ+04, BBJ+05] (more formal).

As hinted above, it is impractical, if not impossible, to have a universal language that caters to all types of analyses [MLM+13, LPJ10, KGP12], because the diversity of types of analysis and their distinct requirement on the level of detail. Therefore, a tailored solution pertaining to the level of detail for each type of analysis is a better option. Note here that Sect. 3 elaborates further on what we mean by "analysis" and "level of detail".

The right level of detail is not straightforward to achieve. Currently, it is mainly done in an ad-hoc manner, where for each analysis task, a language engineer basically has to start from scratch. This gap is also reflected in literature. The authors of [MLM+13, p.18] state that, for analysis purposes, architects call for extending current languages with extra properties to enhance their expressiveness for a particular domain. Yet this language extension is not considered a trivial task as one may tend to add excessive detail to a language [MLM+13, p.13], describing aspects of a system that are trivial or uninteresting, while the most interesting discussions are not documented.

Furthermore, [MLM+13] observes a tension between the architect's dual roles as an analyst (the "introvert" architect) and as a communicator (the "extrovert" architect). On the one hand, as stated, architects as analysts call for extending current languages with extra (e.g. domain specific) properties. On the other hand, for communication purposes architects prefer a language that is "simple enough to communicate the right message to stakeholders"[MLM+13, p.18]. To this end, [MLM+13] states that architectural languages should be generic and semi-formal, rather than domain-specific, and detailed.

The above two challenges further emphasize that, for model-driven enterprise analysis, dealing with the model's level detail is a non-trivial issue. In this position paper, we argue for an approach that tackles the challenges from a model-driven, language-based perspective. We envision a generic analysis-driven EA language adaptation approach to assist language engineers in evaluating and adapting, in a controlled manner, EA languages for model-driven enterprise analyses.

As elaborated in Sect. 4, such an approach will address the problem from the following two aspects: firstly, a framework will be established to evaluate the fitness of a candidate EA modeling language for the targeted analysis purpose; secondly, language customization and model integration techniques will be studied to realize the suggested adaptation.

As a starting point for the evaluation part, we can take inspiration from existing work such as model quality [KSJ06, Moo05] to assess the level of detail. For the adaptation part, we can exploit model integration techniques [ZKK07, KM10] to make languages fit for analysis purposes. Each of these elements is a valuable component for creating our language adaptation approach. However, these elements by themselves are not sufficient for our research purposes. First, model quality work is generic, thus lacking a capability to specifically assess the capacity of models for dealing with a particular analysis. Second, how to specialize and combine these individual works in an effective approach to indeed support model-driven enterprise analyses has not received much research attention yet.

Meanwhile, in developing the approach, we will give special care to achieve a balance between using EA models for communication and analysis purposes. We require such attention since the addition of details adds complexity that may inhibit communication. As a starting point we can use literature on (1) the design principles behind the ArchiMate language [LPJ10], which are explicitly aimed at model complexity reduction through, e.g., conceptual integrity principles, (2) model (de-composition), which subdivides models into smaller relevant models to aid communication [MKG13], and (3) model complexity management [Moo09], which provides techniques to construct a visual notation that does not overload the human mind.

As such, the main contribution of this paper is twofold (1) to argue for an approach that can adapt, in a systematic way, EA languages to cope with various enterprise analyses, (2) a first impression of what we consider to be the key elements for such an approach.

This paper is structured as follows. Sect. 2 discusses how the state of the art forms useful input for analysis-driven language adaptation, and where it falls short. Thereafter Sect. 3 argues for systematic analysis-driven language adaptation, whereas Sect. 4 provides a first impression of what we consider to be important elements for such language adapation. Sect. 5 concludes, and provides a research outlook.

2 Background

2.1 Enterprise Architecture modeling

Various enterprise architecture frameworks provide ingredients for enterprise analysis. To name a few: ARIS [SN00, STA05], CIMOSA [KVZ99], DoDAF/MoDAF, ArchiMate [Lea13], MEMO [Fra02], and UPDM [OMG13].

UPDM (Unified Profile for DoDAF/MoDAF) is an OMG standard language that unifies concepts and viewpoints from the enterprise architecture frameworks DoDAF and MoDAF [1]. It provides a standard UML profile for expressing DoDAF/MoDAF concepts [OMG13, p.17]. Furthermore, UPDM provides a mapping to SysML, which provides a starting point for model driven analysis of EA models created with UPDM.

CIMOSA and ARIS are frameworks for creating and managing enterprise architectures, which both provide a process-oriented modeling language for expressing enterprise architectures. Particularly, ARIS provides the well known Event Process Chain (EPC) language [STA05].

While the above mentioned frameworks and languages may provide a starting point for different analyses, they provide few guidelines for adaptation to a particular EA analysis. Dealing with different levels of detail, or how to instantiate SysML models for analyses, is not further specified. This is also true for EPC. While various formalizations of EPC exist in academic literature, they typically remain on a detailed workflow level. Thus, how to deal with the different enterprise perspectives required to do an enterprise wide analysis (e.g. strategic goals or computational resources) , or how to deal with different levels of detail, remains - to the best of our knowledge - underresearched for EPC models.

In summary: while various EA modeling languages provide a good starting point for analysis they remain just that: *a starting point*.

2.2 Model quality

Work on model quality provides us with hints on how to assess the fitness of a language for a particular purpose. Here frameworks evaluate the general quality of a model [KSJ06], for example if the syntax of a model is appropriate for the modeling task at hand. Furthermore, some model quality frameworks focus on evaluating a particular type of model, such as UML activity diagrams [GFSN$^+$11].

However, as emphasized by [Moo05], the field of model quality is still immature. In particular, a multitude of academic propositions for model quality exist, but few have been extensively validated in practice [Moo05]. Furthermore, there is no agreement on what different model quality characteristics mean. The latter is more recently reflected in [HFL12], who try to build a consensus around the model quality attribute “understandability”. Fur-

[1] DoDAF and MoDAF are frameworks that provide a standard way for planning and managing enterprise architectures, but are not languages.

thermore, to the best of our knowledge, no model quality work exists pertaining to assessing the fitness of a model for analysis purposes. Yet we expect at least some characteristics to be specific for model based analysis, such as the level of formality of a model.

2.3 Enterprise analysis

As mentioned before there exist several instances of model driven enterprise wide analysis, e.g. [JIV+14, JI09, dBBJ+04, QSL12]. These analyses often require models to express the enterprise at different levels of detail. Particularly, this is illustrated by the profitability analysis exposed in [JIV+14].

[JIV+14] introduces an approach for reasoning under uncertainty about profitability analysis of to-be business networks, by extending the net present value calculations from the e^3value modeling language with probabilities on the occurrence of future scenarios.

In line with the e^3value language, [JIV+14] performs a profitability analysis based on the value exchanges of the actors participating in the business network. Thus they remain at a *high level of abstraction*. However, [JIV+14, p.25] admits that many of the details to do a in-depth, substantial profitability calculation require them to "zoom in" on various modeled elements of the business network. They then go on to argue that such details could typically be obtained by relating their business network profitability approach to approaches for enterprise architecture cost analysis and prediction, prominently [JI09].

Yet, in contrast to [JIV+14], [JI09] remains at a *very detailed level of abstraction*. For example, their insurance case analyzes business processes such as "store damage report", as supported by a "report scanning application" (see [JI09, p.66]). Thus here there is an apparent gap in level of detail between the two types of analyses, which needs to be addressed by language engineers if one wants to perform an in-depth profitability analysis.

Dealing with such differing levels of detail in a controlled manner is not a straightforward task to achieve, as mentioned in the introduction. We actually observed the problem of dealing with different levels of abstraction in our own work, particularly regarding an experiment on bridging the value modeling technique e^3value with ArchiMate via the transaction modeling technique DEMO [KGP12]. Here, a key idea behind this experiment is to use DEMO transaction patterns to analyze what business process steps are required to realize economic transactions stemming from e^3value, and to subsequently use these business processes as a starting point for ArchiMate modeling. Yet, applying the DEMO transaction patterns yielded detailed process models focused on communication acts, such as "send an acknowledgment receipt". As a result the produced process models were not fully suitable for ArchiMate, which typically expresses process models at a high level of abstraction. As a result, as part of future work, [KGP12] suggests to assess the fitness of connecting DEMO and ArchiMate due to their differing level of detail, and how to deal with this connection. This is actually one instance of the more general problem statement described in this paper.

2.4 Model driven language engineering

Model Driven Engineering (MDE) is an engineering discipline where *models* are systematically used as the primary artifacts throughout the engineering lifecycle. A model is a sound abstraction of an original, being a software system or an enterprise for example, allowing predictions or inferences to be made [Küh06]. Models are expressed in modeling languages. The definition of a modeling language consists of the specification of the following components: abstract syntax, concrete syntax and semantics, as well as mappings between them. Model Driven Language Engineering (MDLE) applies MDE to language engineering [Kle09]. More specifically, models are exploited to capture all the components of a language specification. The mapping between these artifacts are established by model transformations. Models used to define languages are referred to as metamodels, namely, models of models [Küh06, OMG03].

The main contributions of the proposed approach, namely (1) the definition of the notion of level of detail and the evaluation of a candidate EA modeling language for a targeted analysis purpose, (2) the techniques to adapt languages towards the right level of detail, and (3) the support to balance EA model communication capability and the presence of extra complexity, will be largely following the mindset from MDLE. Various types of (meta-)models and model transformation techniques will be identified, defined, and exploited for achieving our goals.

2.5 Language adaptation

Adapting a candidate EA modeling language towards the right level of detail to serve a given analysis purpose involves two directions of manipulations: to remove unnecessary details, and to introduce missing details.

For the former, existing works on metamodel pruning provide inspiration. The idea is to eliminate unnecessary details of a modeling language and obtain a minimal set of modeling elements containing a required subset of elements of interest. Metamodel pruning techniques have been investigated for various purposes such as the construction of model transformations [SMBJ09]. In our own previous work, we developed a generic (meta-)model decomposition technique and applied it to the Eclipse Modeling Framework to improve language comprehension [MKG13]. However, analysis oriented purposes that we will address in adapting languages have not yet been considered by pruning techniques.

For the latter, language integration techniques provide methods to *actually enrich* current EA modeling languages with analysis capabilities. Such techniques allow for capitalizing on the complementary strengths of languages by (1) merging two languages, or subsets thereof, into a new language. [KBJK03, ZKK07] propose example techniques for this; (2) keeping a federated set of languages, thus establishing links between metamodels of individual languages, but leaving the original metamodels untouched. An exemplar set of federated enterprise models is MEMO [Fra02], which consists of a set of models that each express a relevant perspective on the enterprise; (3) having an intermediate enterprise modeling language (a “hub”), through which different enterprise modeling languages (the “spokes”) are linked. The Unified Enterprise Modeling Language (UEML) [Ver02] is a prominent example of this strategy.

Furthermore, given the model-driven nature of the proposed approach, we can also exploit existing model composition techniques and aspect-oriented modeling techniques when missing details need to be introduced. Examples of such techniques include [BBDF$^+$06, WS08, ODPRK08, SSK$^+$07, KM10].

Yet, current language integration techniques do not sufficiently deal with tensions between languages existing at differing levels of detail, neither do they take care not to sacrifice the communication capability of a language while integrating it with others.

3 Research Objectives

Our objective is to develop an approach for adapting the level of detail of EA modeling languages to cope with different enterprise analyses in a controlled manner.

This section rationalizes such as an approach. We do so by breaking down our main objective into four sub objectives, that we subsequently discuss in further detail.

1. Define the level of detail of a modeling language.

   Different persons may have a different interpretation of abstract vs detailed granularity. Consider two example languages: the Business Process Modeling Notation (BPMN) and ArchiMate. Here, one may argue that ArchiMate is more detailed than BPMN because, *syntactically*, it allows one to express the IT applications and physical IT infrastructure, in addition to business processes. However, one may also argue that BPMN is more detailed than ArchiMate because, *semantically*, the processes captured in BPMN express more specific temporal dependencies (e.g. task A finishes before task B starts).

   The above example illustrates that there is a need to clarify the term “level of detail” so that we can assess languages accordingly. Moreover, we consider that at least syntax and semantics are two important dimensions.

2. Provide a diagnosis of the level of detail of EA languages along different dimensions with respect to an analysis task.

   A language might not be detailed enough for one type of analysis, but sufficient for another. Consider cost management for two stakeholders: an enterprise architect, and a process manager. On the one hand, the enterprise architect concerns himself with having a global overview of costs. To arrive at the global costs overview, he requires syntactical details to capture an enterprise holistically. ArchiMate is an example language having the required level of detail. On the other hand, the process manager is concerned with finding out the costs of each step in a process, and for computing the cost of a process within a time frame. To achieve these, he requires both syntactical details to capture the structure of the business process and semantical details to capture the dynamic behavior of the process. Languages such as BPMN would be more appropriate candidates to consider than ArchiMate.

The above example shows a need to assess the fitness of a language for performing a particular analysis in terms of level of detail. Furthermore, in case of mismatches, one should also pinpoint discrepancies.

3. Adapt the level of detail of EA languages pertaining to a particular type of analysis.

   Following up on the diagnosis of fitness of a language for an analysis, we need a systematic approach to language adaptation. As we observed from the existing work, languages adapted in an ad-hoc manner have the following shortcomings [MLM+13]: (1) unnecessary details might be introduced which makes the resulting models too complex; (2) necessary details might be overlooked which prevents the provisioning of analysis results relevant to end users; (3) inconsistency might emerge in the adapted language as a result of introducing concepts that overlap and/or conflict with existing ones. Hence, we need techniques to adapt EA languages in a controlled manner towards the right level of detail and meanwhile following guidelines to avoid unnecessary complexity.

4. Balance extra level of detail required for analysis with communication.

   In line with [MLM+13] facilitating communication is deemed important for industry uptake and use of architecture modeling languages. Yet, [MLM+13] also shows that the focus on analysis for languages - as predominant in academia - has at least partly hindered their communicability. Hence, while designing our approach for currently used EA modeling languages we should take care not to sacrifice communicability of the models for the sake of analysis.

4 How to adapt

Now that we have discussed the objectives for and rationale behind an analysis driven approach for adapting enterprise architecture languages, we discuss a first version of the envisioned approach. Particularly, we envision that our approach will consist of the following three parts: (1) a granularity scale framework, (2) an analysis-driven language adaptation method, (3) techniques and tools for balancing between a model's level of detail and its communication capability.

1. Granularity scale framework We define a framework to clarify the notion of level of detail of EA modeling languages and to assess the fitness of an EA modeling language with respect to an analysis task.

Based upon previous work on enriching enterprise modeling languages with analysis capabilities, e.g., [JI09, QSL12, BBJ+05], we initially identify three dimensions along which EA modeling languages can be adjusted, namely the syntax, the semantics and the level of formality. The first two dimensions are aligned with the components of a language

Stakeholder	Concerns	Analysis Description	Required Information	Language Evaluation	
				ArchiMate	BPMN
Enterprise Architect	To find out the costs of all enterprise level components	Examples of enterprise level components include: products and services, business processes, IT applications, and physical infrastructure. The cost of a component can influence the cost of another due to the interdependency between them. For example, the cost of purchasing and maintaining an IT asset should find its way to the costs of business processes supported by the asset, proportionally to their execution.	Products and services, and their costs	1	0
			Business processes, and their costs	1	1
			IT applications, and their costs	1	0
			Physical infrastructure, and their costs	1	0
			Interdependency among enterprise level components	1	0
Process Manager	To find out the cost of a process within a time frame	The cost of a process depends upon the costs of each step in the process, and how often they are executed, which is influenced by the frequency of process execution and probability of branch choice.	Fine-grained business process model	1	2
			Business process execution semantics	0	1
			Cost of carrying out a process activity	0	0

Table 1: Granularity scale for cost management analysis with ArchiMate and BPMN. Scoring: 0) no support, 1) partial support, 2) full support.

specification in model driven language engineering, and the last dimension determines the analysis capability of a language as witnessed by [MLM+13]. These three dimensions together constitute a so called 3D "granularity scale" that will be used by the framework to frame the level of detail of EA modeling languages.

With the granularity scale, the framework will follow a method to assess the fitness of EA languages along different dimensions with respect to an analysis task. For the design of such a method, the procedural methods proposed for assessing model quality [KSJ06, Moo05] provide us with a useful starting point. Furthermore, we seek inspiration from guidelines proposed by ontology mapping literature [CSH06] to identify syntactical and semantical heterogeneity between ontologies[2] for the purpose of pinpointing various types of mismatches along all the dimensions.

The envisioned framework takes as input an EA modeling language and a given analysis task, and produces a qualitative "fitness for purpose" diagnosis elaborated along the three dimensions. Note that, since we aim at focusing our effort on the level of detail, we consider a language's concrete syntax out of scope for our approach.

An early version of such a granularity scale framework is presented in Table 1. Here we see the cost management analysis example for two languages, ArchiMate and BPMN, discussed in Sect. 3, analyzed on: (1) the involved stakeholders, (2) their analysis concerns, (3) the information required for addressing the concerns, and (4) the fitness of a language with respect to the required information. At this point in time, we aggregate the fitness status on our three dimensions into a single score ranging from 0 (no support) to 2 (full support). For example: we see that an enterprise architect requires information on different enterprise components, and that ArchiMate is more suitable in expressing this information than BPMN. This is reflected in the scores: 1 versus 0 (note: ArchiMate natively does not support expression of costs, hence we score it as 1 instead of 2).

[2]Here ontology refers to a formal ontology: "a formal specification of a shared conceptualization" [BAT97]. Similar to a modeling language, ontologies are usually specified in terms of concepts and their interrelations, and are formalized to the point that a computer can process them. Furthermore, discrepancies between ontologies are analyzed in typical modeling language terms, prominently syntax and semantics.

2. Analysis-driven language adaptation In line with the "fitness for purpose" diagnosis, we develop adaptation techniques to integrate EA modeling languages at different levels of detail.

We mainly consider two types of language integration techniques: loosely coupled, i.e., federating a set of languages in coherence by mapping the concepts from different languages; tightly coupled, i.e., decompose existing languages into language fragments then compose the fragments into a Domain Specific Language (DSL) with the right level of detail. An exemplar set of federated enterprise models is MEMO [Fra02], which consists of a set of models that each express a relevant perspective on the enterprise. The Unified Enterprise Modeling Language (UEML) [Ver02] is another prominent example of this strategy. For the DSL based approach, we need techniques in two directions, namely: to remove unnecessary details, and to introduce missing details. For the former, we explore techniques such as metamodel pruning [SMBJ09], model slicing [BLC08, BCBB11] and model decomposition [MKG13] which help in identifying the part of the language relevant for the analysis at hand. For the latter, we explore techniques such as language merging [KBJK03, ZKK07] and model composition [KM10].

We reason that the federation based approach is light-weight in the sense that instead of cutting them off, unnecessary details are simply hidden and the original languages remain untouched. As a consequence, existing tools and models of these languages can be reused. Moreover, in cases where the users are already familiar with the individual languages (which remain untouched), the learning curve of the adapted language might be less steep. However, these advantages come at a price. For example, for automated analysis, extra efforts in terms of model transformations are needed to filter and gather relevant information from original models.

On the contrary, we posit that the DSL based approach calls for more efforts at the language adaptation and EA modeling stage. However, it enjoys all the advantages a DSL brings about compared to a general purpose language [MHS05], being gains in domain expressiveness, ease of use, etc. Moreover, by definition, once created, the DSLs are precisely at the right level of detail for the targeted analyses.

In order to be generic, we will support both approaches and provide guidelines in selecting the appropriate techniques.

3. Balancing communication and level of detail We develop guidelines to control model complexity, and model visualization techniques, implemented into a software tool, to facilitate communication of EA models.

On the one hand, the proposed guidelines aid language engineers during the language adaptation process. Particularly, we aim at controlling model complexity when enriching a language with analysis capabilities. To this end, we can capitalize on literature pertaining to (1) the design principles behind the ArchiMate language [LPJ10], which are explicitly aimed at model complexity reduction through, e.g., conceptual integrity principles, (2) more generally, design principles for engineering complex systems (e.g. [Bro87]) such as "do not introduce what is irrelevant".

On the other hand, the model visualization techniques, and corresponding tools, aid in hiding a model's complexity while communicating. More specifically, we aim to exploit (1) model de-composition, which subdivides models into smaller relevant models to hide unnecessary complexity [MKG13], (2) model complexity management techniques [Moo09], which provides a means to construct a visual notation that does not overload the human mind.

5 Conclusions and outlook

In this paper, we have argued for an approach that assists language engineers in systematically adapting EA languages to make them fit for various enterprise wide analysis. Furthermore we provided an overview of what we consider to be the key elements for such an approach.

We are aware of the ambition level of this research effort. In line with this we foresee the following more concrete research challenges for each of our three key language adaptation elements:

Concerning the *granularity scale framework* we have to clarify further *how* we actually assess the fitness of a language for a particular analysis purpose. One challenge here is to decide on objective versus subjective measuring, the difference being that (1) with objective measuring, one assesses language fitness by analyzing a language specification in the light of the analysis purpose, whereas (2) with subjective measuring, one assesses language fitness by eliciting stakeholder opinions on the fitness of a language for the analysis purpose at hand. Furthermore, we should elaborate on the grading scale compared to the initial version in Table 1, in particular regarding the derivation of scores along each of the three granularity scale dimensions: syntax, semantics, and level of formality.

Concerning the *analysis-driven language adaptation*, we should exhaustively and systematically compare existing language enriching approaches, so as to provide for a road map for selecting a suitable language enrichment approach. To the best of our knowledge such a systematic overview, that covers *multiple fields* (enterprise modeling, model driven engineering, ontology mapping), does not yet exist.

Finally, concerning the *balancing of analysis and communication*, we should further analyze literature on the design of complex systems. In addition, in testing a model's communicability, we should ultimately involve model end users, such as enterprise architects with a modeling background. We plan on involving them in case studies, which will be the primary means for practical validation of our proposed approach.

References

[AW09] Stephan Aier and Robert Winter. Virtual decoupling for IT/business alignment–conceptual foundations, architecture design and implementation example. *Business & Information Systems Engineering*, 1(2):150–163, 2009.

[BAT97] Pim Borst, Hans Akkermans, and Jan Top. Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2):365–406, 1997.

[BBDF+06] Jean Bézivin, Salim Bouzitouna, Marcos Del Fabro, Marie P. Gervais, Frédéric Jouault, Dimitrios Kolovos, Ivan Kurtev, and Richard F. Paige. A Canonical Scheme for Model Composition. In *Proceedings of the 2nd European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA 2006)*, volume 4066 of *Lecture Notes in Computer Science*, pages 346–360, 2006.

[BBJ+05] F. S. de Boer, M. M. Bonsangue, J. Jacob, A. Stam, and L. van der Torre. Enterprise Architecture Analysis with XML. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences - Volume 08*, HICSS '05, pages 222.2–, Washington, DC, USA, 2005. IEEE Computer Society.

[BCBB11] Arnaud Blouin, Benoît Combemale, Benoit Baudry, and Olivier Beaudoux. Modeling Model Slicers. In *Proceedings of the ACM/IEEE 14th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2011)*, pages 62–76, 2011.

[BLC08] Jung Ho Bae, KwangMin Lee, and Heung Seok Chae. Modularization of the UML Metamodel Using Model Slicing. *Fifth International Conference on Information Technology: New Generations*, 0:1253–1254, 2008.

[Bro87] F.P. Brooks Jr. No silver bullet: essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, April 1987.

[CSH06] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A Survey on Ontology Mapping. *SIGMOD Rec.*, 35(3):34–41, September 2006.

[dBBJ+04] Frank S de Boer, Marcello M Bonsangue, Joost Jacob, Andries Stam, and L Van der Torre. A logical viewpoint on architectures. In *Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*, pages 73–83. IEEE, 2004.

[FDP+12] Christophe Feltus, Eric Dubois, Erik Proper, Iver Band, and Michaël Petit. Enhancing the ArchiMate Standard with a Responsibility Modeling Language for Access Rights Management. In *Proceedings of the Fifth International Conference on Security of Information and Networks*, SIN '12, pages 12–19, New York, NY, USA, 2012. ACM.

[Fra02] Ulrich Frank. Multi-perspective enterprise modeling (MEMO) conceptual framework and modeling languages. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 1258–1267. IEEE, 2002.

[GFSN+11] Marcela Genero, Ana M Fernández-Saez, H James Nelson, Geert Poels, and Mario Piattini. Research review: a systematic literature review on the quality of UML models. *Journal of Database Management (JDM)*, 22(3):46–70, 2011.

[HFL12] Constantin Houy, Peter Fettke, and Peter Loos. Understanding Understandability of Conceptual Models. What Are We Actually Talking about? In Paolo Atzeni, David Cheung, and Sudha Ram, editors, *Conceptual Modeling*, volume 7532 of *Lecture Notes in Computer Science*, pages 64–77. Springer Berlin Heidelberg, 2012.

[IJLP12] M.-E. Iacob, H. Jonkers, Mark M. Lankhorst, and Henderik A. Proper. *ArchiMate 2.0 Specification*. The Open Group, 2012.

[JI09] Henk Jonkers and Maria-Eugenia Iacob. *Performance and cost analysis of service-oriented enterprise architectures*. IGI Global, Hershey, PA, 2009.

[JIV+14] Pontus Johnson, MariaEugenia Iacob, Margus Välja, Marten Sinderen, Christer Magnusson, and Tobias Ladhe. A method for predicting the probability of business network profitability. *Information Systems and e-Business Management*, pages 1–27, 2014.

[KBJK03] Harald Kühn, Franz Bayer, Stefan Junginger, and Dimitris Karagiannis. Enterprise Model Integration. In Kurt Bauknecht, AMin Tjoa, and Gerald Quirchmayr, editors, *E-Commerce and Web Technologies*, volume 2738 of *Lecture Notes in Computer Science*, pages 379–392. Springer Berlin Heidelberg, 2003.

[KGP12] Sybren de Kinderen, Khaled Gaaloul, and Henderik A. Proper. Bridging value modelling to ArchiMate via transaction modelling. *Software & Systems Modeling*, pages 1–15, 2012.

[Kle09] Anneke Kleppe. *Software Language Engineering: Creating Domain-specific Languages Using Metamodels*. Addison-Wesley, 2009.

[KM10] Pierre Kelsen and Qin Ma. A Modular Model Composition Technique. In David S. Rosenblum and Gabriele Taentzer, editors, *FASE*, volume 6013 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2010.

[KSJ06] John Krogstie, Guttorm Sindre, and Håvard Jørgensen. Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems*, 15(1):91–102, 2006.

[Küh06] Thomas Kühne. Matters of (Meta-)Modeling. *Software and System Modeling*, 5(4):369–385, 2006.

[KVZ99] Kurt Kosanke, F Vernadat, and Martin Zelm. CIMOSA: enterprise engineering and integration. *Computers in industry*, 40(2):83–97, 1999.

[Lea13] Marc Lankhorst and et al. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2013.

[LJH10] Robert Lagerström, Pontus Johnson, and David Höök. Architecture analysis of enterprise systems modifiability–models, analysis, and validation. *Journal of Systems and Software*, 83(8):1387–1403, 2010.

[LJJ+06] Åsa Lindström, Pontus Johnson, Erik Johansson, Mathias Ekstedt, and Mårten Simonsson. A survey on CIO concerns - do enterprise architecture frameworks support them? *Information Systems Frontiers*, 8(2):81–90, 2006.

[LPJ10] M.M. Lankhorst, H.A. Proper, and H. Jonkers. The Anatomy of the ArchiMate Language. *International Journal of Information System Modeling and Design (IJISMD)*, 1(1):1–32, 2010.

[MHS05] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and How to Develop Domain-specific Languages. *ACM Comput. Surv.*, 37(4):316–344, December 2005.

[MKG13] Qin Ma, Pierre Kelsen, and Christian Glodt. A Generic Model Decomposition Technique and its Application to the Eclipse Modeling Framework. *Software & Systems Modeling*, 2013.

[MLM+13] Ivano Malavolta, Patricia Lago, Henry Muccini, Patrizio Pelliccione, and Antony Tang. What industry needs from architectural languages: A survey. *Software Engineering, IEEE Transactions on*, 39(6):869–891, 2013.

[Moo05] Daniel L Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.

[Moo09] Daniel L Moody. The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *Software Engineering, IEEE Transactions on*, 35(6):756–779, 2009.

[ODPRK08] Audrey Occello, Anne-Marie Dery-Pinna, Michel Riveill, and Günter Kniesel. Managing Model Evolution Using the CCBM Approach. In *Proceedings of 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS-MBD workshop)*, pages 453–462. IEEE Computer Society, 2008.

[OMG03] OMG. MDA Guide v1.0.1, June 2003.

[OMG13] OMG. Unified Profile for DoDAF and MoDAF (UPDM), version 2.1, August 2013.

[OPW+08] M. Op ’t Land, H.A. Proper, M. Waage, J. Cloo, and C. Steghuis. *Enterprise Architecture – Creating Value by Informed Governance*. Enterprise Engineering Series. Springer, Berlin, Germany, 2008.

[QSL12] Dick Quartel, Maarten WA Steen, and Marc M Lankhorst. Application and project portfolio valuation using enterprise architecture and business requirements modelling. *Enterprise Information Systems*, 6(2):189–213, 2012.

[SMBJ09] Sagar Sen, Naouel Moha, Benoit Baudry, and Jean-Marc Jézéquel. Meta-model Pruning. In Andy Schürr and Bran Selic, editors, *MoDELS*, volume 5795 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2009.

[SN00] August-Wilhelm Scheer and Markus Nüttgens. ARIS Architecture and Reference Models for Business Process Management. In Wil Aalst et al., editor, *Business Process Management*, volume 1806 of *Lecture Notes in Computer Science*, pages 376–389. Springer Berlin Heidelberg, 2000.

[SSK+07] Andrea Schauerhuber, Wieland Schwinger, Elisabeth Kapsammer, Werner Retschitzegger, Manuel Wimmer, and Gerti Kappel. A Survey on Aspect-Oriented Modeling Approaches. Technical report, E188 - Institut für Softwaretechnik und Interaktive Systeme; Technische Universität Wien, 2007.

[STA05] August-Wilhelm Scheer, Oliver Thomas, and Otmar Adam. Process modeling using event-driven process chains. *Process-Aware Information Systems*, pages 119–146, 2005.

[Ver02] F. Vernadat. UEML: Towards a unified enterprise modelling language. *International Journal of Production Research*, 40(17):4309–4321, 2002.

[WS08] Ingo Weisemöller and Andy Schürr. Formal Definition of MOF 2.0 Metamodel Components and Composition. In *MoDELS ’08: Proceedings of the 11th international conference on Model Driven Engineering Languages and Systems*, pages 386–400, Berlin, Heidelberg, 2008. Springer-Verlag.

[ZKK07] S. Zivkovic, H. Kuhn, and D. Karagiannis. Facilitate modelling using method integration: An approach using mappings and integration rules. In *proceedings of the 15th European Conference on Information Systems (ECIS)*, 2007.

Outlining a Graphical Model Query Approach Based on Graph Matching

Dominic Breuker, Hanns-Alexander Dietrich, Matthias Steinhorst, Patrick Delfmann

European Research Center for Information Systems (ERCIS)
University of Münster
Leonardo-Campus 3, 48149 Münster, Germany
{breuker|dietrich|steinhorst|delfmann}@ercis.uni-muenster.de

Abstract: This paper outlines a graphical model query approach based on graph matching. It consists of a graphical query specification language and a matching algorithm based on graph matching that takes the query as input and returns all matches found in a model to be searched. The graphical query specification language can be used to draw model queries much like a model would be constructed. To achieve applicability in many different model analysis scenarios, the query approach provides structurally exact and structurally similar pattern matching as well as semantic comparison of model node and edge contents. Following a design science research process, we derive functional requirements for the query language and matching algorithm from the literature, outline its syntax, formally specify its matching principle, and demonstrate its functionality by providing a working prototype implementing previously identified requirements.

1 Introduction

Conceptual models are labeled and typed graphs that can be used to describe and analyze an organization and its information systems. Examples of conceptual models are process models that describe the order in which a set of business activities are executed, data models that capture the data necessary to execute business tasks, or organizational charts representing the relationships of employees and departments of a company. Conceptual models do not only serve to document but also to analyze specific aspects of corporate reality. In this context, we understand "analysis" as the task of extracting relevant structural and semantic (i.e., label) information out of conceptual models for a given task. In many cases, such an analysis results in querying a collection of models to identify model fragments with (partly) given structural characteristics and (partly) given contents. Such an analysis may serve various business objectives (for a detailed discussion on model analysis purposes, including different kinds of conceptual models, cf. [De14]):

- One reason for analyzing conceptual models is *compliance checking* against laws and regulations [Aw07]. Compliance management has become an important management task that – if done incorrectly or not at all – can be very costly for an organization. Compliance checking requires identifying model fragments, either in process models only or in combination with other types of models (e.g., data models, organizational charts, etc.), whose structure and contents indicate a compliance violation [Kn10].

- *Identifying weaknesses in process models* serves to improve business processes according to efficiency, effectiveness, or quality. It aims at avoiding double work or unnecessary manual processing, for instance [Bec10]. Identifying such weaknesses requires finding process model fragments whose structure and contents typically indicate a shortcoming of a business process.
- Model translation is about *transforming conceptual models* from one notation into another one, for instance from a conceptual into an executable specification. Notable examples discussed in the literature include translating BPMN models into BPEL code [ODA08, Ga08] or transforming data models into relational schemas. Model translation requires identifying model fragments of a source notation that are to be translated to model or code fragments of a target notation.
- A further purpose of model analysis is *checking models for structural or behavioral conflicts* to ensure their syntactical correctness and – in case of process models – their proper execution semantics [Me08]. Syntax errors or improper execution semantics may disrupt runtime execution. Hence, identifying such problems can help assuring proper model execution in the case of automation. Identifying structural or behavioral conflicts requires identifying model fragments that indicate such conflicts.

Querying conceptual models to identify fragments with particular characteristics thus serves an abundance of analysis tasks that are performed to design, redesign, or improve corporate information systems in different ways. Hence, many companies have started to create large collections of conceptual models [DRR12]. Such collections mostly include process models but may also include other types of conceptual models like data models, organizational charts, or ontologies. They may contain hundreds to thousands of models and, in turn, each model may consist of hundreds to thousands of elements [Di11]. Given the complexity of these collections, the task of analyzing conceptual models is becoming increasingly difficult [Di11]. Analyzing conceptual models is even more complex considering that such an analysis may serve different business objectives as described above. The scientific community has put forth a great number of different model query approaches (cf. Section 6). They allow for querying a model collection automatically in order to identify particular pattern matches in the models. A pattern match in this context refers to a model fragment that complies with a predefined query pattern defining the fragment's structure and labels.

A common characteristic of many recent model query approaches is that they are specifically designed to support one particular business objective (e.g., [YDG12]). In addition, some of these approaches are designed for analyzing models developed in a particular modeling language (e.g., [Aw07] or [Be08]). Instead of developing such customized query approaches, we follow the argument of [Aa13] arguing that it is more beneficial to develop a query approach that can be used to support multiple business objectives. Furthermore, we argue that it is beneficial to develop an approach that is by default applicable to multiple types of conceptual models (i.e., process models, data models, etc.) and models of any modeling language. Companies that wish to automatically analyze their models may not be able to use a given, specialized query approach if it does not fit their use case or modeling language. Furthermore, for reasons of economy and internal conventions, they may not be willing to change their preferred modeling language for every new analysis purpose to render some specialized analysis approach applicable [Aa13].

A previous utility evaluation of a multi-purpose and language independent model query language [De10] revealed that process managers perceive such a mechanism to be highly beneficial to support model analysis [Be11]. The query language presented in this study, however, uses formal set operations to define pattern queries. We argue that the ease of use of this language is rather low as pattern specification is cumbersome (cf. application examples in [Be11]). We argue that a query language is much easier to use if it allows for graphically modeling a pattern in much the same way as a model is developed.

In this paper, we thus present a multi-purpose and modeling language independent model query approach that allows for specifying patterns graphically. The approach consists of a graphical pattern editor and a matching algorithm. The theoretical foundations of our approach are based on graph theory, because any conceptual model is essentially a graph consisting of nodes and edges, no matter what modeling language is used. These nodes and edges are usually attributed with a type and a label, sometimes multiple labels, such as a name, cost, time, etc. A BPMN model [BPM13], for instance, contains nodes of type "task" and may have a name containing the value "grant credit".

In graph theory, the problem of pattern matching is known as the problem of subgraph isomorphism [Ul78]. Corresponding algorithms are able to detect exact pattern matches (i.e., the pattern match and the pattern have to be structurally and semantically identical). In the context of analyzing conceptual models however, it is often necessary to identify paths of model elements that are of previously unknown length (cf. requirements described in Section 2). Consequently, a matching mechanism is required that allows for identifying subgraphs in a model that are not strictly identical to a predefined pattern, but may contain paths of previously unknown length. In graph theory, this kind of pattern matching is known as the problem of subgraph homeomorphism [LW09]. Unfortunately, corresponding algorithms produce a huge number of pattern matches because by default all pattern edges are mapped to paths of all possible lengths in the model. This, however, is often not necessary, because only few pattern edges may need to be mapped to paths in the model. The resulting huge number of potentially irrelevant matching results leads to increased runtimes [LW09]. We therefore introduce a new type of graph problem that we call relaxed subgraph isomorphism, in which a node in the pattern graph has exactly one equivalent node in the model, but an edge in the pattern graph may – if so specified – be mapped to a path of elements in the model.

The purpose of this paper is thus to formally specify, implement, and evaluate a graph-based query approach for conceptual models using relaxed subgraph isomorphism detection. It takes into account that nodes and edges of conceptual model graphs can be annotated with various attributes as described above and that these attributes may have to be checked within the pattern matching process. Furthermore, we take into account that conceptual models may contain both directed and undirected edges, as well as more than one edge between two given nodes (e.g., hierarchy structures in Entity-Relationship Models (ERM), [Ch76]). Summarizing, the contribution of this paper is as follows:

- From a theoretical point of view, we introduce a new type of graph problem called relaxed subgraph isomorphism designed to address the particular requirements of pattern matching in conceptual models. We furthermore present a novel approach to

solve this problem, including specifics of conceptual models such as mixed directed/undirected edges as well as node and edge semantics. Up to now, only related work on subgraph isomorphism and subgraph homeomorphism exists. It does not take into account configurable edge-path mappings as described above. Also, it is restricted to either undirected or directed edges and mostly ignores node and edge semantics, which are both of utmost importance when analyzing conceptual models.
- From a practical point of view, the query approach supports a wide variety of different model analysis objectives involving pattern matching (cf. examples above). It is thus not restricted to supporting one objective alone.
- The query language is furthermore not restricted to finding pattern matches in conceptual models of a particular type or modeling language, but can be used on models of any type or modeling notation – as long as they are based on graphs.
- The query language allows to graphically model a pattern in much the same way as a model is developed. We argue that it is thus easier to use than query languages that rely on text-based pattern specification and follow the evaluation results of [Be11].

The remainder of the paper is structured as follows: In Section 2, we formulate functional requirements for pattern matching in conceptual models that were derived from typical patterns coming from literature on model analysis. In Section 3, we formally introduce the notions of a conceptual model, of subgraph isomorphism, and of relaxed subgraph isomorphism as a basis for the matching process. In a next step, we describe the graphical specification of patterns and briefly outline how our matching algorithm realizes relaxed subgraph isomorphism (Section 4). Section 5 presents a prototypical implementation of the model query approach using a meta-modeling tool. In Section 6, we evaluate our solution by arguing for its utility in comparison to existing model analysis approaches. The paper concludes with a summary of its contribution, limitations and an outlook to future research in Section 7.

2 Requirements of Pattern Matching in Conceptual Models

Several patterns that are relevant for model analysis purposes have been discussed in the literature and stem from research fields already named in the introduction. In particular, in order to identify model sections that match such patterns, a model query approach should comply with the following requirements (for a detailed discussion related to these requirements and an empirical derivation cf. [De14]):

- Requirement 1 (R1): The matching algorithm of the query approach should be able to return model subgraphs that structurally exactly match a predefined pattern graph. For example, this is needed to identify neighbored model elements, as it is necessary, for instance, for syntax checking (e.g., if two nodes of different types are not allowed to be connected by edges).
- Requirement 2 (R2): The matching algorithm of the query approach should be able to return model elements that have a particular type like “BPMN task”, “EPC function”, etc., whenever this is specified in the pattern. If such types are not specified, elements of any type should be returned. This is necessary for all of the business tasks

outlined in Section 1. For instance, for compliance checking, we need to know if two activities follow each other in a process model.

- Requirement 3 (R3): The matching algorithm of the approach should be able to return model elements that have a particular label like “Check invoice” whenever this is specified in the pattern. If such labels are not specified, the algorithm should return elements with any label. For instance, we need to be able to evaluate labels if we search for weaknesses (e.g., a “print” activity followed by a “scan” activity).
- Requirement 4 (R4): The matching algorithm of the query approach should be able to return model fragments containing a path of previously unknown length. This means that it should be possible to specify a pattern that contains simple edges and edges that are mapped to a path in the model. For example, compliance incorporates rules that prescribe that before an activity is performed in a process, another one must have been performed before, but it is not necessary that these both activities directly follow each other. So, we must be able to check whether there is a path between them.

These requirements have been derived from the structure and the label semantics such patterns typically have. In the following, we provide a brief example for each requirement. For a comprehensive list of patterns, we refer to the literature (e.g., [Bec10], [Ga08], [ODA08], [Me08], [Be11], [ADW08], [Me07], [PGD12], [Ba05], and [De14]). In addition to these requirements, we argue that a query language for conceptual models should be applicable to not only process models, but also to any other type of conceptual model like data models, organizational charts, ontologies, etc. It should furthermore not be limited to analyzing models developed in a particular modeling language [Aa13]. Also, the approach should provide a graphical model editor (cf. [SA13] for an additional discussion on the benefit of graphical pattern editors compared to text-based approaches). We therefore add two additional requirements as follows:

- Requirement 5 (R5): The query language should be applicable to conceptual models of multiple types and languages.
- Requirement 6 (R6): The query language should provide a graphical pattern editor that allows for modeling a pattern graph according to R1 to R5. This pattern graph is then augmented to include modeling language-specific type and label information.

3 Formal Specification

In this section, we formalize the functional requirements identified above. To do so, we first introduce the concepts of a conceptual model in terms of a graph and then proceed with defining the problem of subgraph isomorphism (cf. R1). Finally, we note that subgraph isomorphism must be relaxed to account for the requirements we identified (cf. R4). To account for the variety of conceptual modeling languages, we keep our definition of a conceptual model as abstract as possible. The goal is to ensure that the model query language can be used flexibly, no matter what modeling language is used or in which way a language was adapted (cf. R5). We assume that conceptual models consist of nodes representing any domain object of interest, and of edges describing relationships between them. Additional information regarding the nature of objects and relationships are conceptualized as attributes annotated to nodes and edges (cf. R2, R3).

Definition 1 (conceptual model): A conceptual model is a tuple $M=(O,R,A,\alpha)$, with O being a non-empty set of objects (nodes) and R being a non-empty set of relationships (edges). We write $E=O\cup R$ to denote the set of all model elements. $R=R_D\cup R_U$ consists of directed and undirected relationships. As in many modeling languages, multiple relationships are allowed between the same two nodes (like, e.g., hierarchy structures in ERMs), we use *multi-edges* to define relationships as follows: $R_D\subseteq O\times O\times \mathbf{N}$ is the set of directed relationships between the objects of a conceptual model (i.e., directed edges of the model graph). $R_U\subseteq\{\{o_x,o_y,n\} \mid o_x,o_y\in O,\ n\in \mathbf{N},\ o_x\neq o_y\}$ is the set of undirected relationships between the objects of a conceptual model (i.e., undirected edges of the model graph). $\mathbf{N}$ is the set of natural numbers used to number multi-edges. Numbering of multi-edges always starts at $n=1$ and increases by one for each additional edge. A is a non-empty-set of attributes carrying all information that can be associated with elements of E. It can be used, for example, to assign an element a type (e.g., a BPMN "task" or an EPC "function", cf. R2) or a label (e.g., "receive goods" to describe an activity in a process, cf. R3). In general, elements from A can be high-dimensional vectors of attributes assigned to either objects or relations via the function $\alpha: E\rightarrow A$.

As the goal of the model query language is to map elements of one conceptual model (a pattern) to those of another one (the model), we must define which elements are compatible with each other. Clearly, this depends on the context of application (cf. Section 2). It may be a simple equality check of element types (cf. R2) or a full-fledged check applying similarity measures to textual descriptions (cf. R3). Again, to keep things general, we define only an equivalence relation $\sim\subseteq A\times A$ on attributes. As an example, this equivalence relation could be implemented as a simple equality check on types. More complicated equivalence relations taking into account multiple attribute dimensions and based on, for instance, string similarity, linguistic [DHL09] or ontological [TF09] similarity measures are easily conceivable.

We use the notation $M'\leq M$ to denote that M' is a model that can be obtained from M by removing objects from O, relationships from R, and by reducing the domain of α accordingly. Formally, this reduction can be described as $O'\subseteq O$, $R'\subseteq R$, and $\alpha'=\alpha|_{E'}$. In terms of graph theory, M' is called a subgraph of M and exactly matches a fragment of M. We define a pattern query P as a model that is searched for in another one M ($|O_P|\leq|O_M|$ and $|R_P|\leq|R_M|$). Given a model graph M and a pattern graph P, P is isomorphic to a subgraph $M'\leq M$ if there is a structure preserving one-to-one mapping φ between all elements of P and all elements of M' (cf. R1) satisfying the equivalence relation $\alpha(e_P)\sim\alpha(\varphi(e_P))$. The following definition formalizes this subgraph isomorphism relation.

Definition 2 (subgraph isomorphism): Given a pattern graph $P=(O_P,R_P,A,\alpha_P)$, a model graph $M=(O_M,R_M,A,\alpha_M)$, and an equivalence relation $\sim\subseteq A\times A$, P is subgraph-isomorphic to M if and only if it exists an $M'\subseteq M$ such that there exists a bijection $\varphi: E_P\rightarrow E_{M'}$ satisfying the isomorphism condition:

$$(o_x,o_y,n_P)\in R_P \Leftrightarrow (\varphi(o_x),\varphi(o_y),n_M)\in R_{M'};\ \{o_x,o_y,n_P\}\in R_P \Leftrightarrow \{\varphi(o_x),\varphi(o_y),n_M\}\in R_{M'};$$
$$\alpha(o_{Pj})\sim\alpha(\varphi(o_{Pj})),\ \alpha(r_{Pj})\sim\alpha(\varphi(r_{Pj})),\ n_P,n_M\in \mathbf{N}$$

To extend this definition towards relaxed subgraph isomorphism, we first have to introduce the notion of a path. A path in a model graph M can be understood as a sequence of objects $<o_1,\ldots,o_n>$ such that $(o_i,o_{i+1},z_i)\in R \vee (o_{i+1},o_i,z_i)\in R \vee \{o_i,o_{i+1},z_i\}\in R \;\; \forall i\in\{1..n\text{-}1\}$, $z_i\in \boldsymbol{N}$. We write $paths(o_x,o_y)$ to denote the set of all paths between nodes o_x and o_y. Special types of paths are those obeying certain constraints on the directions of their edges. $paths_d(o_x,o_y)$ shall denote the set of all directed paths from o_x to o_y, meaning all sequences of objects $<o_x,\ldots,o_y>$ such that $(o_i,o_{i+1},z_i)\in R \;\; \forall\; i\in\{x..y\text{-}1\},\; z_i\in \boldsymbol{N}$. Conversely, $paths_u(o_x,o_y)$ shall denote the set of all undirected paths between o_x to o_y, meaning all sequences of objects $<o_x,\ldots,o_y>$ such that $\{o_i,o_{i+1},z_i\}\in R \;\; \forall\; i\in\{x..y\text{-}1\},\; z_i\in \boldsymbol{N}$. Finally, $paths_a(o_x,o_y)$ shall denote the set of all paths between o_x and o_y, ignoring the direction of the contained edges, that is, all sequences of objects $<o_x,\ldots,o_y>$ such that $\{o_i,o_{i+1},z_i\}\in R \vee (o_i,o_{i+1},z_i)\in R \vee (o_{i+1},o_i,z_i)\in R \;\; \forall i\in\{x..y\text{-}1\},\; z_i\in \boldsymbol{N}$.

Based on paths, we define the notion of relaxed subgraph isomorphism (cf. R4). In the subgraph isomorphism definition, any pair of model nodes must be connected directly via an edge (i.e., a path of length one) whenever the two pattern graph nodes that map on them are connected by an edge. In the relaxed notion, these model graph nodes may be connected via paths of any length. Effectively, edges in the pattern graph *can* be mapped to paths instead of edges in the model graph, but only when this is explicitly specified in the pattern. To choose for which pattern edges this generalization shall be used, and which kind of path definition should be applied, we define a function $p\colon R_P \rightarrow \{edge, normalPath, mixedPath\}$. It indicates whether an edge of the pattern graph shall correspond to an edge in the model graph (edge), a directed or undirected path (normalPath), or any path regardless of the directions of its edges (mixedPath). The value must be specified by the user for each pattern edge. In the following, we call edges of the pattern to be mapped to paths in the model *path-edges*.

Definition 3 (relaxed subgraph isomorphism): Given a pattern graph $P=(O_P,R_P,A,\alpha_P)$, a model graph $M=(O_M,R_M,A,\alpha_M)$, an equivalence relation $\sim\subseteq A\times A$, and a function $p\colon R_P \rightarrow \{edge, normalPath, mixedPath\}$, P is relaxedly subgraph-isomorphic to M if and only if it exists an $M'\subseteq M$ such that there exists a left-total relation $\psi\colon E_P \rightarrow \boldsymbol{P}(E_{M'})$ satisfying the relaxed isomorphism condition:

$$(o_x,o_y,n_P)\in R_P \wedge p((o_x,o_y,n_P))=edge \Leftrightarrow (\psi(o_x),\psi(o_y),n_M)\in R_{M'},\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ \alpha(r_{Pj})\sim\alpha(\psi(r_{Pj})),\ n_P,n_M\in \boldsymbol{N};\ \{o_x,o_y,n_P\}\in R_P \wedge p(\{o_x,o_y,n_P\})=edge \Leftrightarrow \{\psi(o_x),\psi(o_y),n_M\}\in R_{M'},\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ \alpha(r_{Pj})\sim\alpha(\psi(r_{Pj})),\ n_P,n_M\in \boldsymbol{N};\ (o_x,o_y,n_P)\in R_P \wedge p((o_x,o_y,n_P))=normalPath \Leftrightarrow |paths_d(\psi(o_x),\psi(o_y))|=max(n_P),\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ n_P\in \boldsymbol{N};\ \{o_x,o_y,n_P\}\in R_P \wedge p(\{o_x,o_y,n_P\})=normalPath \Leftrightarrow |paths_u(\psi(o_x),\psi(o_y))|=max(n_P),\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ n_P\in \boldsymbol{N};\ (o_x,o_y,n_P)\in R_P \wedge p((o_x,o_y,n_P))=mixedPath \Leftrightarrow |paths_a(\psi(o_x),\psi(o_y))|=max(n_P),\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ n_P\in \boldsymbol{N};\ \{o_x,o_y,n_P\}\in R_P \wedge p(\{o_x,o_y,n_P\})=mixedPath \Leftrightarrow |paths_a(\psi(o_x),\psi(o_y))|=max(n_P),\ \alpha(o_{Pj})\sim\alpha(\psi(o_{Pj})),\ n_P\in \boldsymbol{N}$$

The definition assures that, in one mapping, each node of the pattern is matched to exactly one node in the model, every simple edge of the pattern is matched exactly to one edge in the model, and every path-edge of the pattern is matched to exactly one path in the model. Whenever two nodes of the pattern are connected by more than one (n) path-edge, the definition assures that these *n* path-edges are always mapped to *n different* paths in the model (cf. the $max(n_P)$ condition).

4 The Query Language

Based on the definitions above, pattern matching always starts with the definition of a pattern that should be searched for in conceptual models. To that end, we introduce the syntax of a pattern query. A pattern query is essentially a graph consisting of nodes and edges as described in *Definition 1*. To define a pattern query, we propose not to have it specified formally, but have it drawn and transformed automatically into a formal representation complying with the pattern definition afterwards. For example, consider the pattern query shown in Figure 1. It represents a behavioral conflict in EPCs. In particular, when the triggering event fires, the succeeding AND connector may never fire when the process instance was routed to the other path by the XOR connector [Me07].

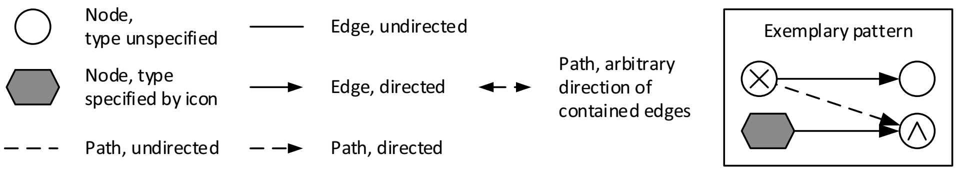

Figure 1: Pattern specification in the graphical concrete syntax

Nodes o_P of such a visual pattern query are represented by circles and edges/paths r_P by lines. If a node is assigned a node type (e.g., "entity type") as part of $\alpha(o_P)$, we propose to attach an icon according to the representation of a corresponding node type in a model. In the example, three nodes are typed, so they appear as a red hexagon, a circle containing an "x" and a circle containing a "∧", standing for "event", "XOR connector", and "AND connector". One node is not typed, so it can be mapped to any node of a model. Labels as further parts of $\alpha(o_P)$ should appear within the boundary of a node. Further attributes of $\alpha(o_P)$ should not appear as a visual part of the pattern query. In an implementation, they should be specified by opening a context menu. Pattern edges to be mapped to edges in a model r_P are represented by solid lines with one attached arrow if they are directed. Pattern edges to be mapped to paths in a model r_P are represented by dashed lines with an arrow attached if they are directed. Without an arrow, they are undirected. With arrows at both ends, they should be mapped to paths containing edges of any direction. As we cannot identify the type of an edge only from its representation in many modeling languages, we propose to attach the type of the edge – if specified – textually. Any other attributes of $\alpha(r_P)$ should be handled like those of $\alpha(o_P)$.

5 Implementation

To demonstrate the feasibility of our model query language, we implemented it as a plugin for a meta-modeling tool that was available from a previous research project.

Being a meta-modeling tool, it allows for specifying modeling languages at runtime. Conceptually, it is based on the idea that any modeling language essentially provides a set of object types and relationship types. To create a model these types are instantiated into concrete objects and relationships. The query language we propose is thus based on the same constructs that are used on meta-level to define a modeling language. This is why the query language is essentially modeling language-independent.

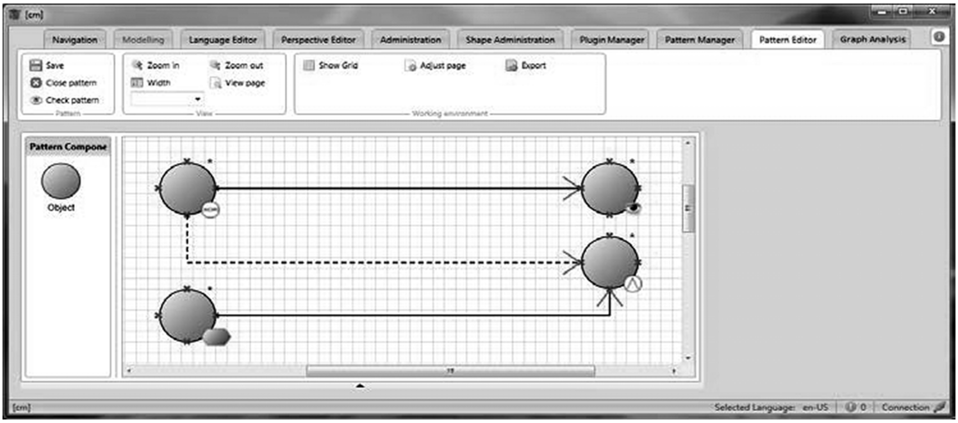

Figure 2: Pattern editor

The implementation of the query language includes a pattern editor as depicted in Figure 2. This editor allows for drawing a pattern graph complying with the Requirements R1 to R5 derived above (R6). The analyst can specify a name for the pattern query and choose the modeling language the pattern query is supposed to be valid for (R5). The type of a node or edge can be customized according to the corresponding modeling language. The plugin accesses the modeling language specification in order to get all type information that is required during the matching process. The exemplary pattern query in Figure 2 matches the example in Figure 1.

After having specified a pattern query or choosing a previously specified one, the user can specify which models should be analyzed. We implemented an algorithm that matches the pattern to the models according to the formal definitions in Section 3. The algorithm determines all pattern matches in all input models. The plugin returns a list of models that contain pattern matches. By selecting an entry from this list, the model is loaded in the modeling environment of the meta-modeling tool. All returned pattern matches are highlighted in different colors to allow for retracing which pattern node was mapped to which model node and which pattern edge was mapped to which model edge or path (cf. Figure 3). In the example, a pattern match was found that contains a path between the XOR split and the AND join. If a model contains more than one pattern match, the user can browse through the matches, meaning that the highlighting of the model changes to the corresponding places for every match.

To assure the applicability of the model query approach and its implementation, we performed a preliminary runtime experiment, in which we searched for fourteen specific patterns. Seven of these patterns were EPC patterns, and seven were ERM patterns. We applied them to 53 EPCs (sizes from 15 to 294 elements) and 42 ERMs (sizes from 16 to 97 elements) coming from the retail industry (for details on the models and patterns, please see [De14], where we used the same patterns and models to test another query language). We conducted the performance evaluation on an Intel® Core™ 2 Duo CPU E8400 3.0 GHZ with 4 GB RAM and Windows 7 (62-Bit edition). We disabled the energy saving settings in Windows and executed the application as a 32-bit real-time process to avoid any unnecessary hardware slow down or process switching. As a result, we observed runtimes for searching one pattern in one model ranging from fractions of a millisecond to just under five seconds. In more than 90% of the pattern matching cases, results could be obtained in less than 100 milliseconds for ERMs (in less than 10 milliseconds for EPCs). Note that the (few) long runtimes were due to path searches with most of the restrictions like type, label etc. turned off. For instance, one ERM search that took 2130 milliseconds returned 22856 matches. Hence, we evaluate the overall performance of the approach satisfactory.

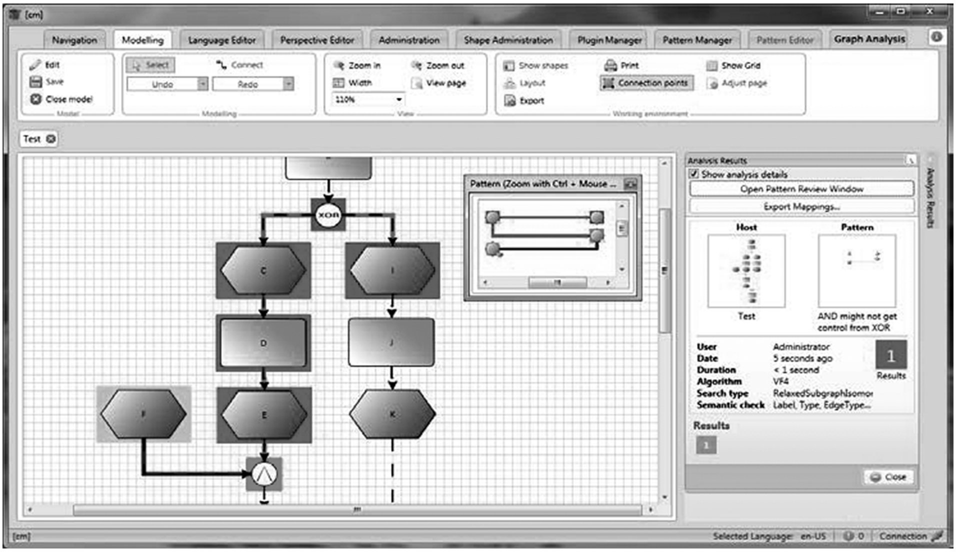

Figure 3: Pattern match visualization

6 Related Work

Table contains a detailed comparison of our work to other model query languages proposed in the literature. We draw on the requirements for a graph-based model query language presented in Section 2 to compare our work with existing approaches. A query language thus has to be able to find arbitrary isomorphic substructures in a model graph (R1), consider type (R2) and (R3) label information in its matching process and find paths of arbitrary length (R4). A query language should furthermore support querying conceptual models of any type or graph-based modeling language (R5) and provide a graphical pattern editor (R6). The table contains a "+" if the query language fulfills a given requirement and a "-" otherwise. The table contains a "0" if the given requirement is only partly fulfilled (see details below).

Three classes of query languages can be distinguished. *The first class* contains all process model query languages. aFSA [MW06], APQL [So11], BeehiveZ [Ji10], BPMN VQL [FT09], BPMN-Q [Aw07], BPQL [MS04], BP-QL [Be08], IPM-PQL [CKJ07], PPSL [Fö07] and an approach based on indexing and untanglings [PRH14] belong to this class. These query languages can only be applied to process models. Some of these languages were designed to query models of a particular process modeling language. BPMN-Q [Aw07] is a prominent example. Other authors define a new process modeling language and propose a query approach for this language (cf. BPQL and BP-QL). The *second class* of query languages contains those approaches that are specific to UML models. EMF-IncQuery [Ber10] and OCL [OCL13] fall into this category. As with process model query languages, these approaches are thus designed to query models of a particular type or modeling languages. Pattern queries are created by means of a declarative programming language. The *third class* contains general graph query languages. SPARQL [W3C13] as well as the Neo4j query language Cypher [Ne13] are prominent examples that fall into this category. They provide functionality that is similar to the feature set of our query language. These approaches are essentially declarative programming languages. A graphical pattern editor is not provided.

Query Language / Requirement	R1	R2	R3	R4	R5	R6
aFSA	+	+	+	-	-	+
APQL	+	+	+	+	-	-
BeehiveZ	+	+	+	-	-	+
BPMN VQL	+	+	+	+	-	+
BPMN-Q	+	+	+	+	-	+
BPQL	+	+	+	-	-	-
BP-QL	+	+	+	+	-	+
Cypher (Neo4j)	+	+	+	+	+	-
EMF-IncQuery	+	+	+	+	-	-
IPM-PQL	+	+	+	+	-	-
OCL	+	+	+	-	-	-
PPSL	+	+	+	+	-	+
SPARQL	+	+	+	+	+	-
Untanglings	-	0	+	+	0	-
VMQL	+	+	+	+	0	0
Our work	+	+	+	+	+	+

Table 1: Comparison of our work to other model query languages

VMQL [St11] is a visual model query language providing functionality that is similar to that of our approach. The language is also intended to be applicable on models developed in any modeling language. However, this has been demonstrated for UML as well as BPMN models only (i.e., R5="0") [SA13]. Extending VMQL to additional modeling languages however requires developing a new pattern editor that provides the element types of that particular modeling language (i.e., R6="0").

7 Contributions, limitations, and outlook

The purpose of this paper was to introduce a query language for conceptual models that is applicable to models of any type or modeling language. Our work can furthermore support multiple business objectives of analyzing models and is thus expected to be broadly applicable. Hence, the contribution of the paper can be summarized as follows:

- The query language we propose includes a matching algorithm that is based on a new kind of graph search problem called *relaxed subgraph isomorphism*, which is particularly related to the analysis of conceptual models. Therefore, we contribute to algorithmic graph theory by introducing a new class of graph search problem. With our work, we expect to trigger further research on this problem from a graph theoretical perspective. For instance, novel efficiency-enhanced algorithms would contribute both to graph theory and – in turn – to conceptual model analysis.
- In practice, only few multi-purpose and language-independent model query languages exist up to now. With our work we hence contribute to a variety of different model analysis objectives requiring pattern matching. Furthermore, our approach can be used on models of any graph-based modeling language. Therefore, we expect wide-spread application, as model analysis and model analysts are no longer restricted to particular modeling languages or few analysis objectives.
- Another important characteristic of our work is its graphical pattern editor easing the usage of the query language.

However, our work reveals some limitations resulting from its broad applicability:

- Defining pattern queries for a particular application scenario is the responsibility of an analyst. Furthermore, a pattern query has to be defined according to the modeling languages of the models to be analyzed. Naturally, although searching for ERM patterns in EPCs is possible using our query language, it will not return any results.
- Another restriction of our query language also results from its applicability to multiple modeling languages. It is by design not able to analyze special characteristics of special model types, for instance execution semantics of process models.
- Our approach has not yet been subject to a comprehensive utility evaluation. Despite this, we expect analysts to appreciate it, as related works have already proven to be highly relevant for model analysis purposes [Be11]. Moreover, due to the possibility to specify patterns graphically, we expect an even higher utility of our approach. Comprehensive utility evaluations are subject of short-term research. In particular, we plan to apply our work in financial institutions to check model repositories containing integrated business process models, data models and organizational charts for regulatory compliance violations and weaknesses. This will furthermore carve out additional functional requirements that have to be included in the query language.

Although we already conducted performance experiments suggesting satisfactory runtime of the query approach (not included in this paper), medium-term research will focus on performing additional runtime experiments on extremely large models with the aim of further increasing execution speed and applying the algorithm to further model analysis scenarios in practice. At the moment, we investigate graph-theoretical structural characteristics of conceptual models that can speed up pattern matching. In particular, we expect bounded treewidth and planarity of conceptual model graphs to be very promising characteristics to increase matching performance significantly.

References

[Aa13] van der Aalst, W.M.P.: Business Process Management: A Comprehensive Survey. ISRN Software Engineering. 2013 (2013), pp. 1-37.

[ADW08] Awad, A.; Decker, G.; Weske, M: Efficient compliance checking using bpmn-q and temporal logic. In: Business Process Management. Berlin 2008, pp. 326-341.

[Aw07] Awad, A.: BPMN-Q: A Language to Query Business Processes. In: Proceedings of the EMISA 2007, pp. 115-128.

[Ba05] Batra, D.: Conceptual data modeling patterns: Representation and validation. Journal of Database Management (JDM), 16 (2005) 2, pp. 84-106.

[Be08] Beeri, C.; Eyal, A.; Kamenkovich, S.; Milo, T.: Querying business processes with BP-QL. Information Systems, 33 (2008) 6, pp. 477-507.

[Be11] Becker, J.; Bergener, P.; Delfmann, P.; Weiss, B. (2011). Modeling and Checking Business Process Compliance Rules in the Financial Sector. In: Proceedings of the ICIS 2011.

[Bec10] Becker, J.; Bergener, P.; Räckers, M.; Weiß, B.; Winkelmann, A.: Pattern-Based Semi-Automatic Analysis of Weaknesses in Semantic Business Process Models in the Banking Sector. In: Proceedings of the ECIS 2010. Pretoria 2010.

[Ber10] Bergmann, G.; Horváth, Á.; Ráth, I.; Varró, D.; Balogh, A.; Balogh, Z.; Ökrös, A.: Incremental Evaluation of Model Queries over EMF Models. In: Proceedings of the International Conference MODELS 2010. Oslo 2010, pp. 76-90.

[BPM13] Object Management Group, Business Process Model and Notation 2.0, 2013. (http://www.omg.org/spec/BPMN/2.0/).

[Ch76] Chen, P.P.-S.: The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems. 1 (1976) 1, pp. 9-36.

[CKJ07] Choi, I.; Kim, K.; Jang, M.: An XML-Based Process Repository and Process Query Language for Integrated Process Management. Knowledge and Process Management 14 (2007) 4, pp. 303-316.

[De10] Delfmann, P.; Herwig, S.; Lis, L.; Stein, A.; Tent, K.; Becker, J.: Pattern Specification and Matching in Conceptual Models. A Generic Approach Based on Set Operations. Enterprise Modelling and Information Systems Architectures 5 (2010) 3, S. 24-43.

[De14] Delfmann, P.; Steinhorst, M.; Dietrich, H.-A.; Becker, J.: The Generic Model Query Language GMQL – Conceptual Specification, Implementation, and Runtime Evaluation. Information Systems. Accepted for publication, DOI: 10.1016/j.is.2014.06.003.

[Di11] Dijkman, R.; Dumas, M.; Van Dongen, B.; Käärik, R.; Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems. 36 (2011) 2, pp. 498-516.

[DHL09] Delfmann, P.; Herwig, S.; Lis, L.: Unified Enterprise Knowledge Representation with Conceptual Models – Capturing Corporate Language in Naming Conventions. In: Proceedings of the ICIS 2009. Phoenix 2009.

[DRR12] Dijkman, R. M.; La Rosa, M.; Reijers, H. A.: Managing Large Collections of Business Process Models – Current Techniques and Challenges. Computers in Industry 63 (2012) 2, pp. 91-97.

[Fö07] Förster, A.; Engels, G.; Schattkowsky, T.; Van Der Straeten, R.: Verification of business process quality constraints based on visual process patterns. In: Proceeding sof the 1st

Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering 2007, pp. 197-208.
[FT09] Di Francescomarino, C.; Tonella, P.: Crosscutting concern documentation by visual query of business processes. In: BPM Workshops 2009. Berlin 2009, pp. 18-31.
[Ga08] García-Bañuelos, L.: Pattern Identification and Classification in the Translation from BPMN to BPEL. In: On the Move to Meaningful Internet Systems: OTM 2008. Berlin 2008, pp. 436-444.
[Ji10] Jin, T.; Wang, J.; Wu, N.; La Rosa, M.; Ter Hofstede, A. H. (2010). Efficient and accurate retrieval of business process models through indexing. In: On the Move to Meaningful Internet Systems: OTM 2010. Berlin 2010, pp. 402-409.
[Kn10] Knuplesch, D.; Ly, L. T.; Rinderle-Ma, S.; Pfeifer, H.; Dadam, P.: On Enabling Data-Aware Compliance Checking of Business Process Models. In: Proceedings of the 29th International Conference on Conceptual Modeling (ER 2010). Vancouver 2010, pp. 332-346.
[LW09] Lingas, A.; Wahlen, M.: An exact algorithm for subgraph homeomorphism. Journal of Discrete Algorithms 7 (2009) 4, pp. 464-468.
[Me07] Mendling, J.: Detection and Prediction of Errors in EPC Business Process Models, Vienna University of Economics and Business Administration. Vienna 2007.
[Me08] Mendling, J.; Verbeek, H. M. W.; van Dongen, B. F.; van der Aalst, W. M. P.; Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model. Data & Knowledge Engineering 64 (2008) 1, pp. 312-329.
[MS04] Momotko, M.; Subieta, K.: Process Query Language: A Way to Make Workflow Processes More Flexible. In: Proceedings of the 8th East European Conference on Advances in Databases and Information Systems (ADBIS 2004). Budapest 2004, pp. 306-321.
[MW06] Mahleko, B.; Wombacher, A.: Indexing Business Processes based on Annotated Finite State Automata. In: Proceedings of the 2006 IEEE International Conference on Web Services (ICWS'06). Washington, DC 2006, pp. 303-311.
[Ne13] Neo4j: Cyper Query Language. 2013. (http://docs.neo4j.org/chunked/stable/cypher-query-lang.html).
[OCL13] Object Management Group: Object Constraint Language, 2013. (http://www.omg.org/spec/OCL/2.3.1/).
[ODA08] Ouyang, C.; Dumas, M.; Van Der Aalst, W.M.P.: Pattern-based translation of BPMN process models to BPEL web services. International Journal of Web Services Research (IJWSR), 5 (2008) 1, pp. 42-62.
[PGD12] Polyvyanyy, A.; García-Bañuelos, L.; Dumas, M.: Structuring acyclic process models. Information Systems 37 (2012) 6, pp. 518-538.
[PRH14] Polyvyanyy, A.; La Rosa, M.; ter Hofstede, A. H. M.: Indexing and Efficient Instance-based Retrieval of Process Models Using Untanglings. In: Proceedings of the 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014). Thessaloniki, Greece, pp. 439-456.
[SA13] Störrle, H.; Acretoaie, V.: Querying business process models with VMQL. In: Proceedings of the 5th ACM SIGCHI Annual International Workshop on Behaviour Modelling – Foundations and Applications (BMFA 2013). New York 2013, pp. 1-10.
[So11] Song, L.; Jianmin, W.; ter Hofstede, A. H. M.; La Rosa, M.; Ouyang, C.; Wen, L.: A Semantics-based Approach to Querying Process Model Repositories. QUT eprints 2011.
[St11] Störrle, H.: VMQL: A visual language for ad-hoc model querying, Journal of Visual Languages & Computing. 22 (2011) 1, pp. 3-29.
[TF09] Thomas, O.; M. Fellmann M.: Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes. Business & Information Systems Engineering. 1 (2009) 6, pp. 438-451.
[Ul78] Ullmann, J. R.: An Algorithm for Subgraph Isomorphism. Journal of the ACM 23 (1976) 1, pp. 31-42.

[W3C13] World Wide Web Consortium: SPARQL Query Language for RDF, 2013. (http://www.w3.org/TR/rdf-sparql-query/).

[YDG12] Yan, Z.; Dijkman, R.; Grefen, P.: Fast business process similarity search. Distributed and Parallel Databases, 30 (2012) 2, pp. 105-144.

Process Modelling

Choosing an adequate level of detail in business process modelling

Volker Nissen, Christian Heyn, Frank Termer

University of Technology Ilmenau, Chair of Business & Information Systems Engineering in Services, PB 10 05 65, D-98684 Ilmenau, Germany
volker.nissen@tu-ilmenau.de

Abstract: It is a basic matter of business process modelling to determine an adequate level of model detail, a problem which also can be derived from the guidelines of modelling by Becker et al. [BRU00]. In literature, only few recommendations how to solve this problem can be found. In addition they are quite unspecific and have gaps. In this paper, we investigate which measurable factors influence the adequate detail level, and on this basis make proposals for guidelines how it can be determined in a specific application situation.

1 Motivation

According to Becker et al. [BKR03, 4] a process is „a completely closed, timely and logical sequence of activities which are required to work on a process-oriented business object. (…) A business process is a special process that is directed by the business objectives of a company and by the business environment.“ A process model helps in abstracting the real business matter for certain modelling purposes [Ro96, 17]. To model processes, a number of graphic modelling techniques are available, having a fixed defined set of rules and regulations concerning the design of the models each. The most important are the event-driven process chain (EPC), the extended event-driven process chain (eEPC), the Business Process Modelling Notation (BPMN), the activity diagrams of the Unified Modelling Language (UML), Petri nets and the PICTURE method [Ga10, 71]. By using these methods a modelling can be carried out at different levels of detail (LOD). As a conclusion, there is the question how detailed a process should be modelled. This question also results from the guidelines of business process modelling as defined by Becker et al. [BRU00]. The principle of relevance leads to the question, to which extent the modelling in relation to the modelling purpose in mind is to be carried out. Here, the principle of efficiency stands in the way as a limiting factor, which says that the necessary efforts to create a model have to be in an appropriate relation to the benefits of the model. Becker et al. [BRU00] clearly state that their modelling guidelines are rather general and require further refinement.

In the literature concerning process modelling an adequate LOD is often demanded without giving a concrete recommendation how this LOD should be determined [TNW12, 54]. In most of the cases only information about the hierarchical subdivision of processes into refined sub-processes can be found. The terms ‘detail level’ and ‘hierarchy of process models’ are often used synonymously. Other possible aspects of detailing, such as additional information objects or the enrichment of process elements with attribute information are rarely subject of discussion. To the best of our knowledge, a comprehensive work dealing with the detailing problem is not available. Today it is one of the central problems of process modelling to choose the adequate level of model detail, a fact that is also confirmed by a study investigating current problems and future challenges in business process management [IRR09, 9]. Therefore, this contribution deals with the following two research questions: 1. Which factors may influence the adequate LOD of business process models (BPMod)? 2. How can the adequate LOD of BPMod be determined?

2 Data and methodology

A model is an artefact in the sense of Hevner [Hev04]. In our research process we refer to the Design Science Research (DSR) Methodology Process Model (Fig. 1) of Peffers et al. [PTR07, 54], which is widely accepted. By means of the artefact created it shall be possible to choose the correct detail level of the process modelling in an application situation. We demonstrate the applicability and the usability of the artefact by using it in an exemplary scenario. The broader evaluation of our proposal will have to be left to future work. This article is our intention to communicate the so far results to the scientific community.

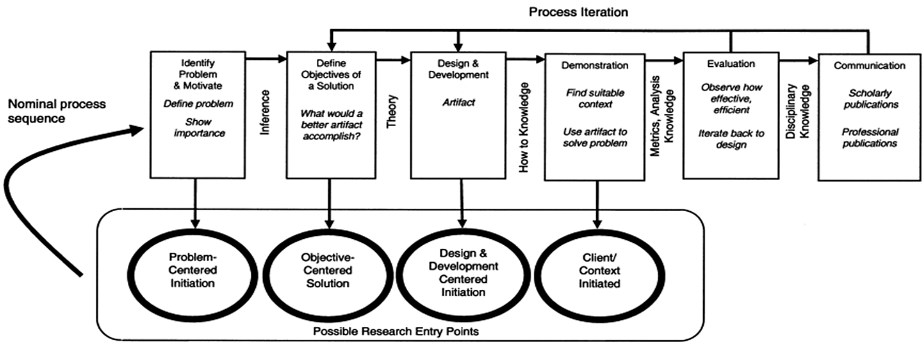

Fig. 1: DSR Methodology Process Model after Peffers et al. [2007, 54].

A literature analysis according to the procedure proposed by Fettke [Fe06] was carried out as a basis for our design plan. For this, books, journal and conference articles with a focus on process modelling, respectively process management, were analyzed. As a starting literature, relevant textbooks for process management such as Allweyer [Al09], Becker et al. [BMW09] and Gadatsch [Ga10] and for business information technology such as Ferstl and Sinz [FS13] and Heinrich et al. [HHR11] were chosen. Furthermore, the keywords „detail level“, „abstraction level“, „granularity“, „detailing“ and „abstraction“ as well as „business process“, „business process modelling“, „process“ and „process modelling“ where taken for a detailed search in data bases and scientific search engines. The German equivalents were used as well. Next to journals and proceedings being A-classified in the business information systems rankings of VHB and WKWI, the following databases and search engines were consulted as well: ACM Digital Library, AISeL, Bielefeld Academic Search Engine, EBSCO, Gemeinsamer Verbundkatalog (GVB), Google-Scholar, IEEE Xplore, Ilmenauer Discovery Tool, catalogue of the library of the University of Ilmenau, Scirus, Springer Link, Web of Knowledge. In this phase, at first only the headlines and, if necessary, the abstracts were used in order to limit the result to relevant hits only. The then following reading of the table of contents, abstracts, summaries or even complete texts further narrowed down the results. In addition, a forward and backward search according to Webster & Watson [WW08] was carried out with the aim to find further sources by using the literature already identified. Then the relevant sources found were analysed and interpreted.

3 Preliminary considerations concerning the level of model detail

A generally accepted definition of the term LOD cannot be found in literature [BT96, 5]. None of the currently existing definitions of the LOD covers it completely. Here, the detail level is to be understood as a characteristic of BPMod, which shows the extent, respectively the granularity of the modelling. A practical visualisation of different LODs is the process hierarchy, which however should have a vertical content-related consistency. Characteristics of the LOD are the process depth, the process width, the process length, the content of information and the attribution [Ro96, 71-84, 132-133].

The process depth describes to how many levels a process model was disaggregated and specified until the desired detailing is reached. The process width describes which states of the system are covered by the model. If the number of modelled special cases increases, the process width rises. The process length is the size of a process model under the condition that process width and process length remain constant. If the process length is too big, it may happen that the process model shows too many details and therefore becomes confusing. But if the process length is too small, important information may not be included. Furthermore, the information content in process models can be further specified. Herewith process models can be further refined with information objects such as organisational units. By means of attributes, process models as well as the information objects can be further specified.

Not all aspects of this definition are equally relevant for the development of a model to determine the adequate LOD. Only the process depth and the process width are completely relevant. A statement about the information content, the process length and the attributes of a model cannot be given precisely. Information objects for example can contain organisational units and by means of attributes these may be enriched. Attributes can be attached to information objects as well as complete BPMod [Ro96, 133]. The investigated objects are not build on one another and therefore a gradation of them is not recommended [TNW12, 63]. For these reasons it is only possible to give hints to the potential information objects and attributes.

Furthermore, the LOD has to be limited adequately, for example dependent on the purpose of modelling [TNW12] [SS08] [BRU00]. An adequate LOD is the solution of a certain process modelling task with the help of a suitable refinement level. To determine the adequate LOD, a top-down approach is useful. This means that models have to be further specified until the LOD needed is reached. The reason why a bottom-up approach is not taken, is that in this case all existing BP of a company have to be modelled on a high LOD first [SS13, 140-142], which could already exceed the LOD needed. Furthermore, the vertical content-related consistency can better be ensured by a top-down approach, as the detailing of the BP is raised gradually. In addition, the relevance and the relation of the respective BPMod to the company are easier to comprehend by means of a gradual refinement.

In literature there are already approaches that intend to identify the correct LOD [Ga83, 65-75] [BR98, 212-219]. The 'problem-oriented approach' concentrates on those areas where problems arise and tries to solve them with the help of BP modelling. The existing problems are initially too complex for a solution approach. Therefore they have to be detailed into sub-problems and to be narrowed down. The detailing has to be continued until concrete measures to solve the problem are possible to take. This approach has the disadvantage that the measures developed to solve the problem may perhaps not cover some parts of the BP and therefore a complete examination and modelling of the BP is not performed. The 'goal- oriented approach' focussed on business goals. Such goals can for example be a reduction of lead time, an optimization of capacities or a reduction of the productions costs. These goals are further specified until concrete measures how to achieve them can be developed with the help of BPMod. In case of the goal-oriented approach only those parts of BP are covered that contribute to the achievement of the goals. As a conclusion, this approach as well as the problem oriented approach involves the risk that BP are modelled incompletely.

4 Factors influencing the adequate level of detail

4.1 Professional criteria

The following professional, respectively content-related criteria were identified to have an impact on the adequate LOD of BPMod:

Structuring: Processes are foreseeable at different degrees, and the resulting models vary in size. Highly structurable BP make it possible to outline the sequence with a high LOD. Weakly structurable BP make it necessary to model the BPMod with a rather lower LOD, as deviations and uncertainties exist [Al09, 65-66].

Repetitiveness: For processes that are carried out often, a rather higher LOD can be strived for than for processes that are carried out rarely. It is easier to find for example improvement potentials when using a high LOD in case of a high repetitiveness [FL97, 29-30] [Al09, 67-68].

Automation: It is necessary to have a high LOD of the modelling, when an automated running of a BP is intended [TNW12, 61]. The more human interventions are necessary, the more this professional criterion tends to be manual and as a conclusion the adequate LOD connected to it decreases.

Modification frequency: Models may have to be changed, for example due to new technologies, changing customer demands or new laws. A BPMod with a high modification frequency should be modelled on a rather low LOD [TNW12, 61].

Flexibility: The necessity for flexibility arises, when uncertainty factors are there or certain liberties for the process handling are necessary. An uncertainty factor can for example arise, when the input of a BP can be diverse and the objects to handle therefore may vary. If the necessary flexibility for a BP is high, it should be rather modelled on a lower LOD. The detailing of the process width may be an exemption, as it shows the different states of the system [Ro96, 133]. In this case a high flexibility can make a high detailing rather necessary [GS05, 137-141] [FL97, 29-30].

Knowledge intensity: According to Schmelzer and Sesselmann [SS13, 70-72] the LOD of BPMod is also dependent on knowledge intensity of the process. Knowledge may be available implicitly as well as explicitly and contains the totality of know how and abilities being necessary to solve a problem [HS11, 286]. Especially the implicit knowledge is relevant here, which is bound to the knowledge carriers. The explicit knowledge contributes to the criterion of data and information intensity. If a BP is dependent on implicit knowledge and therefore has a high knowledge intensity, a low LOD is recommended as often only abstract processing specifications can be made.

Data and information intensity: This criterion is connected to knowledge intensity, with explicit information and data being in the centre of attention here, which can be modelled easily. In case of a high data and information intensity, a high LOD is therefore recommended. As a result, the information and data needed can be set for the processes [Al09, 66-72].

Security aspects: Security aspects may, for instance, concern the protection of internal company data against unauthorised access or the reduction of the accident risk during production processes. For the case that relevant security aspects exist, the LOD should be rather high to show these completely and comprehensible. However, if there are no relevant security aspects, a low LOD can be adequate [LWS08, 287].

Process KPIs: If data for calculating key performance indicators (KPI) has to be collected in a BP, the LOD has to be chosen in such a way that it is possible to determine the KPIs. Thus, usually a rather high LOD is necessary, as to later automatically collect the data through information and communication systems. Furthermore, in case of a high detailing the reasons for possible deviations from the predicted, respectively expected KPI values can be found faster. As a conclusion, the necessary improvements of the BPMod, respectively the BP can be carried out target-oriented and promptly [GSV94,64-66] [HS98, 168-169].

Number of triggering and provisioning events: Triggering events can be seen as an input that starts the BP. Provisioning events are an output of an BP and are able to trigger other BP. If the number of triggering and provisioning events is too big, a detailing should be carried out, otherwise for example uncertainties in handling the BP may arise or the search for the causes of errors is complicated. Therefore the LOD should be rather high, when a BP contains many triggering and provisioning events [SS08, 206].

Resources in the process: Conflicts may arise, when for example a machine is needed for a BP, but the occupancy and release events were modelled in one function only. In this case, the modelling should be more detailed to show the exact use of the resource. A BPMod with an unclear use of resources suggests a higher LOD [Ga83, 81].

Process length: It contains the size of a BPMod on an elected detail level. If the process length is too big, the BPMod may be too complex and loose its clarity [Ro96, 133]. To provide clarity in this case, for example parts of the BP can be summarized to explain them in the next detailing step. Or the BPMod can be divided to raise the clarity. If the process length is too small, there is the risk that not all relevant contents are included. But the process length can only be seen after modelling and the correct process length has to be determined for each model individually. As a conclusion, as general recommendations concerning the process length with reference to the correct LOD are not possible, this criterion is not included into our model. The adequate process length has to be determined after modelling and to be adapted if necessary.

4.2 Model purpose

This article bases on the classification of the process modelling purposes into the categories organisation design and application system design of [RSD08, 50-58]. For the organisation design illustrative BPMod are needed, especially to show the processes in the organisation. In application system design rather formal models are used, as the objectives here have a relation to the application systems [Ro96, 45-46]. The LOD of BPMod for the purpose of organisation design is usually rather smaller than for the purpose of application system design [SS08, 206]. The two purposes mentioned above can be further subdivided on a second level. So on one hand BPMod can be taken as the fulfilment of the purpose and on the other hand as a means to an end. As a result for the modelling purpose we have a matrix consisting of 2x2 arrays. When a BPMod is already the result of a purpose, for example of an organisation design, it may have been created e.g. with the objective of an organisation documentation or of a certification project. When a BPMod serves as a means to an end, the result of the modelling is not simultaneously the fulfilment of the purpose. Herewith the BPMod are used to contribute to the result. To choose a standard software with the help of BPMod, the existing reference models of the software are compared to the company-specific BPMod [RSD08, 54-55]. In this case the BPMod are only needed to support the choice of standard software.

4.3 Basic conditions of process modelling

Basic conditions include factors that may influence the BPMod from the outside. Herewith the external conditions of the BPMod are considered to determine the adequate level of detail.

If the employees taking part in the BP have a sufficient qualification, the LOD should be rather lower than with employees having a low level of qualification. If the LOD is too high in case of a sufficient qualification, this may lead to a limitation of liberties and also to a reduction of motivation. Then, improvement potentials or innovations may not be uncovered or individual creativeness is curbed. But if the qualification of the employees is low, the LOD should be high, as by means of detailed instructions the possibility of processing errors can be reduced [SS08, 206] [Ro96, 139].

BP are influenced by regulations, laws and rules that must be followed by the company and the employees (Compliance). As a conclusion they have an influence on the LOD during BP modelling. It is not possible to give the probable LOD for this criterion, as it has to be chosen depending on the compliance requirements in question.

A further basic condition for the determination of the adequate LOD is the relevance that BP, respectively parts of BP, have on the achievement of the objectives of the company or of a higher-order BP. It makes sense to model a BP in detail, if it is able to contribute a considerable share to the achievement of the objectives. Herewith the handling of this BP becomes easier and improvement potentials can be discovered.

Furthermore, the urgency of a BPMod to be made can have an influence on the LOD. Urgency here means the time horizon, until which a BP is to be modelled [Hä00, 61]. If a BPMod has to be created contemporarily, the LOD should be rather low. But it has to be stated that it may be a preliminary model, which should be revised, specified and improved later. In case of high urgency, the actual LOD necessary can also be attained by increasing the number of employees being involved in modelling.

„Best practices“ is the course of action which seems to be the best to solve certain problems [HHR04, 116]. An existing „best practice“, consisting of already existing models or parts of models which may be used for BP modelling, can be taken as an orientation aid for choosing the adequate LOD.

5 Determination of the adequate detail level

In the following, a methodology to determine the adequate LOD for BPMod is outlined that uses the relevant factors identified above. Then, in section 6, this methodology is applied in order to illustrate its practical use.

5.1 Differentiation according to process depth and process width

The question how detailed a process should be modelled can be answered with view to the process depth and to the process width. Therefore it is at first necessary to describe in greater detail the classification of the adequate detail level along these two dimensions.

In this work a division of the process depth into five levels is taken, which follows [Hü03, 88-93], who differentiates between main (end-to-end) process, process, process step, operation and elementary activity, with the process depth and therefore also the detailing of the BPMod rising in this order starting at main process and going to elementary activity. Main processes as the first hierarchy level of process depth are the value creating primary processes, which may reach across the whole company. The levels below specify only excerpts of the respective higher level of the process depth down to those elementary operations that cannot be further divided in a practical way. Fig. 2 illustrates the different levels of process depth using the example of customer order processing.

The process width shows the states of the system, which are covered by the BPMod and rises with the number of modelled states of the system [Ro96, 133]. In literature only one approach [TNW12, 63] dealing with the grading of the process width could be found. Here the process width is dependent on the execution frequencies of the individual process alternatives as well as on the totality of the possible process variations. Our model refers to this logic, but carries out a five steps differentiation of the process width.

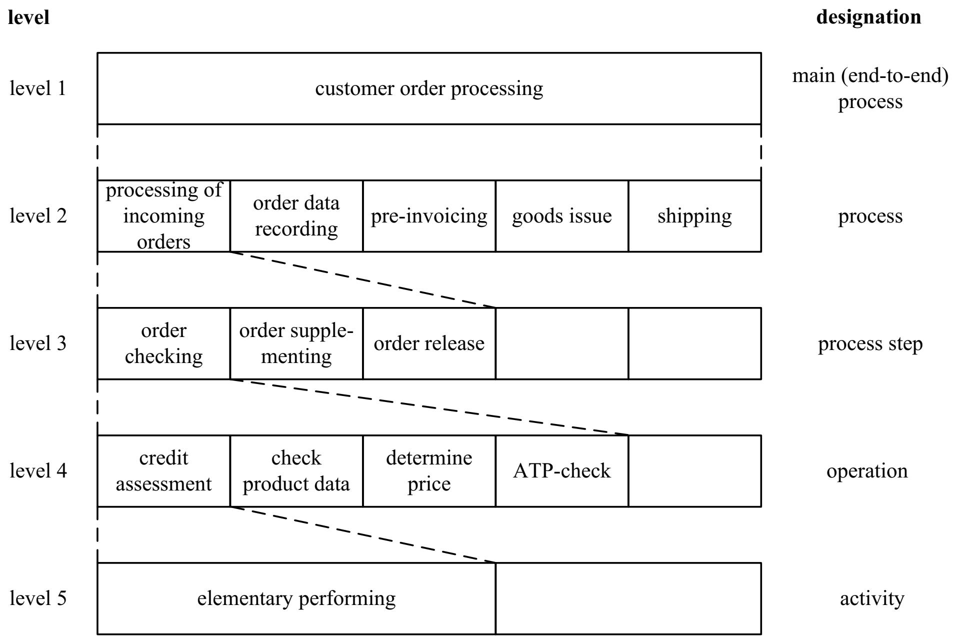

Fig. 2: The different levels of process depth illustrated for main process ‘customer order processing’

The process width and therefore the detailing of the BPMod rises from step one to five. The first step is called „Happy Path“ [FRH10, 131] and shows the best case of the process, with only those variations recorded, that are necessary to carry out the best case. Generally the execution frequency of this best case should be higher than that of the other variations. On the second step further variations of the BP come on board, which is called „Standard Case“ and should cover approx. 65 % of the cases carried out in the system. The third step in the process width is called „Special Case“ and contains more variations than the previous steps. It covers approx. 80 % of the cases carried out in the system. The fourth step is called „Extensive“ and covers approx. 95 % of the cases. The fifth and highest specified step of the process width covers up to 100 % of the possible alternatives and is called „Complete“.

On the basis of these two aspects the adequate LOD of a BPMod can be shown clearly in short form like in the example in Fig. 3. A name field can be used to identify the BPMod as well as a further field for the purpose of the model. The LOD determined is to be entered in the fields process depth and process width. For this, the scale from one to five is to be used and has to be marked until the determined practical LOD. The minimum LOD values recommended are marked darker in the scale. Contrary to that, the lighter colourings are a marking for the recommended maximum detailing. The annotation field can be used for remarks like relevant information objects or attributes.

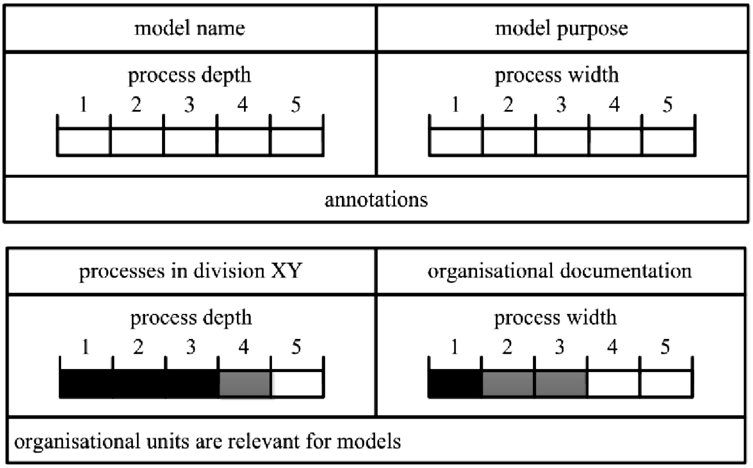

Fig. 3: Template and example for a brief description of the adequate detail level

5.2 Determination of the adequate detail level

To design the model for the determination of the adequate LOD it is necessary to seize on the factors involved shown before and to further structure them. Incompatible combinations of the individual factors are excluded by rules. The individual factors should be weighed out, so that their differing relevance for the LOD can be included.

Rules show probable relations between variations of different professional criteria (Tab. 1) and help to efficiently and consistently determine the variations, if no other information are given. Each line in Tab. 1 represents a rule. The rules put two professional criteria into an „if-then-relation“ and are set according to the following scheme: If criterion one is there in its respective variation, then criterion two tends to have the variation cited. In rule one for example a rather weak structuring is given and therefore the automation criterion should tend to manual. For reasons of space we go without a more precise presentation of the interrelations of the rules.

Tab. 1: Rules concerning the variations of the professional criteria

Rule	If	Then
1	Structuring (weak)	Automation (manual)
2	Structuring (strong)	Knowledge intensity (low)
3	Structuring (weak)	Knowledge intensity (high)
4	Structuring (strong)	Data / Information intensity (high)
5	Structuring (weak)	Data / Information intensity (low)
6	Knowledge intensity (high)	Automation (manual)
7	Automation (automated)	Data / Information intensity (high)
8	Automation (automated)	Modification frequency (rare)
9	Automation (automated)	Flexibility (low)
10	Automation (automated)	Use of resources (unambiguous)

Our suggestion for weighting of the professional criteria can be seen in Fig. 4. This suggestion is open to discussion. We are currently conducting a Delphi study to identify the best weightings. Here it also has to be taken into account that the criterion of flexibility has a deviating weighting concerning process depth and process width, which is made clear by the naming in brackets (W, D).

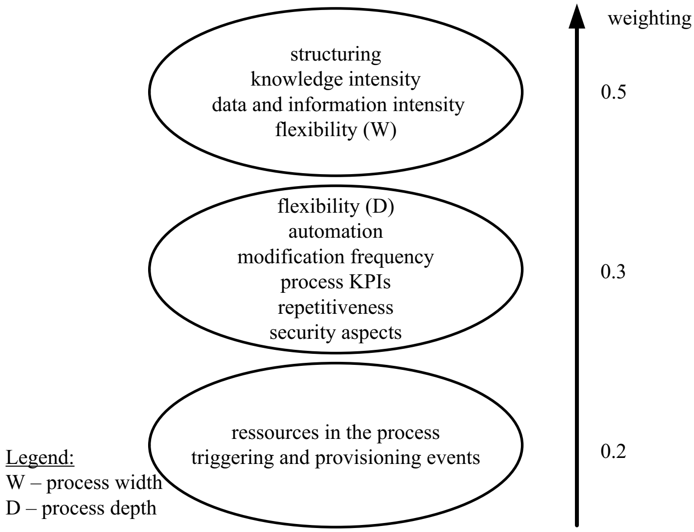

Fig. 4: Weighting of the professional criteria

The highest weighting have the professional criteria structuring, knowledge intensity as well as information and data intensity. As the criteria influence each other, the importance of these criteria can be derived by means of the rules set above. In addition, they are very important for the adequate LOD. Contrary to that, the use of the resources as well as the triggering and provisioning events have to be put into the lowest weighting category, as they only have small influence on the LOD and should only show a tendency for the finding of the adequate LOD.

For the basic conditions of modelling, no weighting or interrelations of rules among each other are assumed. But it is supposed that compliance always has to be fulfilled. Models need to contain all details that are required for meeting the defined compliance requirements. As a conclusion, the subsumed aspects indicate a minimum LOD for the entire BPMod. A second assumption deals with the handling of best practices, which should only serve as a clue for the LOD and not as a fixed default. An adaption of the LOD has to be carried out for example to adapt the best practices to the qualification of the existing employees or to the urgency of completion.

Concerning the relation between modelling purpose and LOD the assumption, taken from literature, is made that BPMod for the purpose of organisation design have to be rather less detailed than the BPMod for application system design [SS08, 193]. It is assumed that BPMod for the purpose of application system design should have a minimum LOD of 3. This detailing refers to the process depth as well as process width.

In Fig. 5 the meanwhile derived model how to determine the adequate LOD is shown. This model consists of the three groups of factors involved: professional criteria, basic conditions and model purpose deliver a LOD. The recommended LOD for the respective BPMod can be derived from them. Furthermore, this model has to be completed two times, to be able to determine the appropriate process depth and process width for a BPMod. In the following demonstration example (referring to [KR12, 535-539]) the application of the model is explained.

6 Application example

For a mail-order company the adequate LOD for the modelling of the BP order processing is to be determined. In addition, this BP is to be supported by a workflow management system in the future with the BPMod being automatically transferred into workflows then. As a conclusion, the purpose of this BPModng can be assigned to application system design, which already says that we have a minimum LOD of three for the process depth and the process width.

In case of the order processing in a mail-order company mostly all steps can be fixed before the processing starts and the arising uncertainty is rather low. For these reasons it can be assumed that the value of the professional criterion structuring is ‘high’. Furthermore it can be assumed that for this BP only few knowledge is needed and that employees after a short introduction are able to carry out this BP as all relevant information for the processing of the BP are available.

As a conclusion it can be said that this BP has a low knowledge intensity and a high data and information intensity. As the support by a workflow management system should be possible, the value of the automation may be considered as high, but not fully automated. Due to the rather few uncertainties it may be deduced that there do not exist many triggering and provisioning events for this BP. Furthermore, due to the tenth rule of the professional criteria, the use of the resources tends to be unambiguous. Concerning the professional criterion flexibility, a value differentiation between process width and depth has to be set. In Tab. 2 the values of the professional criteria for this example are given in an overview, in Tab. 3 the basic conditions for the modelling.

In Fig. 6 the model for the determination of the adequate LOD for the process depth is filled with these exemplary data of the BP. The same table should be made for the process width as well, but differences concern only the criterion of flexibility and the overall condition of best practices, so this table is left out here for reasons of space.

professional criteria	lowest value	LOD 1 ¦ 2 ¦ 3 ¦ 4 ¦ 5	highest value	weighting	calculation	weighted LOD	LOD	highest value	LOD 5 ¦ 4 ¦ 3 ¦ 2 ¦ 1	lowest value	basic conditions
structuring	weak		strong	0.5	0.5 * LOD			low		high	employee qualification (I)
repetitiveness	low		high	0.3	0.3 * LOD			high		low	compliance
automation	manual		automated	0.3	0.3 * LOD			important		unimportant	relevance
modification frequency (I)	frequent		rare	0.3	0.3 * DetLev			low		high	urgency (I)
flexibility (I for D)	high (D) low (W)		low (D) high (W)	0.3 (D) 0.5 (W)	0.3 or 0.5 * LOD			high		low	LOD of best practice
knowledge intensity (I)	high		low	0.5	0.5 * LOD						
information and data intensity	low		high	0.5	0.5 * LOD						
security aspects	low		high	0.3	0.3 * LOD						
process KPIs	few		many	0.3	0.3 * LOD						
triggering and provisioning events	few		many	0.2	0.2 * LOD						
ressources in process (I)	non-ambiguous		ambiguous	0.2	0.2 * LOD						
			sum	3.7 (D) 3.9 (W)	sum			sum			
			sum weighted LOD / sum weighting					sum LOD / 5			
					model purpose:						
					required level of detail:						

Legend:
W - process width
D - process depth
I - inverse scale to standardize calculation

Fig. 5: Scheme for the determination of the adequate detail level

professional criteria	lowest value	LOD 1 ¦ 2 ¦ 3 ¦ 4 ¦ 5	highest value	weighting	calculation	weighted LOD
structuring	weak		strong	0.5	0.5 * LOD	2.5
repetitiveness	low		high	0.3	0.3 * LOD	1.5
automation	manual		automated	0.3	0.3 * LOD	1.2
modification frequency (I)	frequent		rare	0.3	0.3 * LOD	1.2
flexibility (I for D)	high (D) low (W)		low (D) high (W)	0.3 (D) 0.5 (W)	0.3 or 0.5 * LOD	1.2
knowledge intensity (I)	high		low	0.5	0.5 * LOD	2.0
information and data intensity	low		high	0.5	0.5 * LOD	2.0
security aspects	low		high	0.3	0.3 * LOD	0.9
process KPIs	few		many	0.3	0.3 * LOD	1.2
triggering and provisioning events	few		many	0.2	0.2 * LOD	0.4
ressources in process (I)	non-ambiguous		ambiguous	0.2	0.2 * LOD	0.2
			sum	3.7 (D) 3.9 (W)	sum	14.3
			sum weighted LOD / sum weighting			3.9

LOD	highest value	LOD 5 ¦ 4 ¦ 3 ¦ 2 ¦ 1	lowest value	basic conditions
4	low		high	employee qualification (I)
3	high		low	compliance
5	important		unimportant	relevance
4	low		high	urgency (I)
4	high		low	LOD of best practice
20	sum			
4	sum LOD / 5			

model purpose:	order processing (application system design)
required level of detail:	minimal LOD of **3** required

Legend:
W - process width
D - process depth
I - inverse scale to standardize calculation

Fig. 6: Recommended detailing of the process depth for the application example (to be made analogous for the process width, left out for reasons of space)

Tab. 2: Content-related criteria in the demonstration example

Professional criterion	Value	Process depth	Process width
Structuring	High	5 of 5	5 of 5
Repetitiveness	Often	5 of 5	5 of 5
Automation	Automated	4 of 5	4 of 5
Modification frequency	Rare	4 of 5	4 of 5
Flexibility	Low	4 of 5	2 of 5
Knowledge intensity	Low	4 of 5	4 of 5
Information intensity and data intensity	High	4 of 5	4 of 5
Security aspects	Medium	3 of 5	3 of 5
Indexes	High	4 of 5	4 of 5
Provisional and triggering events	Few	2 of 5	2 of 5
Use of resources	Clear	1 of 5	1 of 5

Tab. 3: Basic modelling conditions in the demonstration example

Condition	Value	Process depth	Process width
Employee qualification	Low	4 of 5	4 of 5
Compliance	Medium	3 of 5	3 of 5
Relevance	Important	5 of 5	5 of 5
Urgency	Low	4 of 5	4 of 5
LOD of Best Practices	High	4 of 5	3 of 5

As can be seen in Fig. 6, the LOD should be taken rather high. According to the modelling conditions and the professional criteria, the LOD in the process depth (and process width) should be four (the values determined of the professional criteria are 3.9 (respectively 3.6) and those of the conditions 4.0 (respectively 3.8), they were rounded up to four. Concerning the process depth, this BP should be modelled until the hierarchy level of the operations. And the process width should be modelled until the step „Extensive“, which means that up to 95 % of the possible system cases are covered. Due to the rules and assumptions defined, the minimum LOD for the process depth and the process width is three. This is mainly caused by the main purpose of the application system design and also by the compliance requirements given. In this example, the upper limit of the LOD is defined by the professional criteria and modelling conditions and the lower limit by the model purpose.

In Fig. 7 the relevant LOD can be found as a summary. The minimum LOD recommended has a dark marking, the maximum LOD recommended a lighter marking. Furthermore, the annotation field suggests some information objects for the model. Organisational units may for example show, who is responsible for which step in the BP. Attributes for example can support the automated recording and derivation of KPI data and the introduction of a workflow management system.

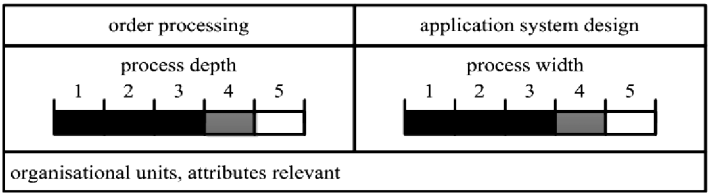

Fig. 7: Recommended LOD for the demonstration example (overview)

7 Limitations and future work

In this work we suggested a model for the determination of the adequate LOD in process modelling and with the help of a demonstration example its practical applicability was shown. A detailed evaluation, for example by means of a case study or by expert interviews, is left to future research. In addition, rules and weightings concerning the relevant factors involved were drawn up, which may be incomplete and would have to be verified in the future as well. A weighting of the model purposes and the basic modelling conditions was left out, a procedure that seems to be justified here, but which also needs further verification. And, the model purpose currently is considered only in a very rough differentiation, which should be refined in the future, too.

Another interesting avenue of research in our context would be the LOD analysis of existing process models in the light of the individual model purpose. This inductive approach would complement our own considerations and might be helpful in the evaluation.

At the moment the efforts necessary to model a BP are not considered. It can be assumed that a rising LOD leads to a rising modelling effort. As a conclusion it is currently not clear, at which LOD of the BPMod the relation between the efforts arising and the benefit is optimal. Finally, it might make sense to implement the proposed model to carry out the assumptions and rules in an automated way and to herewith facilitate the determination of the adequate LOD.

References

[Al09] Allweyer, T.: BPMN 2.0 – Einführung in den Standard für die Geschäftsprozess-modellierung. 2. Aufl., Books on Demand, Norderstedt, 2009.

[BKR03] Becker, J.; Kugeler, M.; Rosemann, M. (eds.): Process Management. Springer, Berlin, 2003.

[BMW09] Becker, J.; Mathas, C.; Winkelmann, A.: Geschäftsprozessmanagement. Springer, Berlin, 2009.

[BRU00] Becker, J.; Rosemann, M.; v. Uthmann, C.: Guidelines of Business Process Modelling. In (v.d. Aalst, Desel, J.; Oberweis, A. eds.): Business Process Management, LNCS 1806, Springer, Berlin, 2000, pp. 30 – 49.

[BR98] Bogaschewsky, R.; Rollberg, R.:Prozeßorientiertes Management. Springer, Berlin, 1998.

[BT96] Brooks, R.; Tobias, A.: Choosing the best Model. In: Mathematical and Computer Modelling 24 (1996)4, 1-14.

[FS13] Ferstl, O; Sinz, E.J.: Grundlagen der Wirtschaftsinformatik. 7. ed., Oldenbourg, München, 2013.

[Fet06] Fettke, P.: State-of-the-Art des State-of-the-Art. Wirtschaftsinformatik, 48 (5), 2006, S. 257–266.

[FL97 Fischermanns, G.; Liebelt, W.: Grundlagen der Prozeßorganisation. 4. ed., Götz Schmidt, Gießen, 1997.

[FRH10] Freund, .J; Rücker, B.; Henninger, T.: Praxishandbuch BPMN. Hanser, München, 2010.

[Ga10] Gadatsch, A.: Grundkurs Geschäftsprozess-Management. 6. Ed., Vieweg+ Teubner / GWV, Wiesbaden, 2010.

[Ga83] Gaitanides, M.: Prozeßorganisation – Entwicklung, Ansätze und Programme prozeß-orientierter Organisationsgestaltung. Vahlen, München, 1983.

[GSV94] Gaitanides, M.; Scholz, R.; Vrohlings, A; Raster, M.: Prozeßmanagement. Hanser, München, 1994.

[GS05] Grief, J.; Seidlmeier, H.: Modellierung von Flexibilität mit Ereignisgesteuerten Prozessketten. In (Nüttgens, M.; Rump, F. eds.): Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, GI e.V., Hamburg, 2005, 137-155.

[Hä00] Hässig, K.: Prozessmanagement. Versus, Zürich, 2000.

[HS98] Hagemeyer, J.; Striemer, R.: Anforderungen an die Erweiterung von Metamodellen für die Geschäftsprozeßmodellierung. In (Herrmann, T; Scheer, A-W; Weber, H. eds.): Verbesserung von Geschäftsprozessen mit flexiblen Workflow-Management-Systemen 1. Physica, Heidelberg, 1998, 161-180.

[HHR04] Heinrich, L.J.; Heinzl, A.; Roithmayr, F.: Wirtschaftsinformatik-Lexikon. 7. ed., Oldenbourg, München, 2004.

[HHR11] Heinrich, L.J.; Heinzl, A.; Riedl, R.: Wirtschaftsinformatik – Einführung und Grundlegung. 4. ed., Springer, Heidelberg, 2011.

[HS11] Heinrich, L.; Stelzer, D.: Informationsmanagement – Grundlagen, Aufgaben, Methoden. 10. ed., Oldenbourg, München, 2011.

[Hü03] Hüsselmann, C.: Fuzzy-Geschäftsprozessmanagement. EUL, Lohmar-Köln, 2003.

[IRR09] Indulska, M.; Recker, J. C.; Rosemann, M.; Green, P.: Business Process Modelling : Current Issues and Future Challenges. In (van Eck, P.; Gordijn, J.; Wieringa, R., Hrsg.) Proc. 21st CAiSE. Springer, Berlin, 2009, S. 501-514.

[KR12] Köster, C; Ribbert, M.: Prozessorientiertes Risikomanagement und dessen Prüfung. In (Becker, J.; Kugeler, M.; Rosemann, M. eds.): Prozessmanagement. 7. ed., Springer, Berlin, 2012, S. 515-539.

[LWS08] Lehner, F.; Wildner, S.; Scholz, M.: Wirtschaftsinformatik. Hanser, München, 2008

[PTR07] Peffers, K.; Tuunanen, T.; Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems 24 (2007) 3, pp. 45 – 77.

[Ro96] Rosemann, M.: Komplexitätsmanagement in Prozessmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Dissertation, Univ. of Münster, 1996.

[Ro08] Rosemann, M.; Schwegmann, A.; Delfmann, P.: Vorbereitung der Prozessmodellierung. In (Becker, J.; Kugeler, M., Rosemann, M., eds.) Prozessmanagement. 6. ed., Springer, Berlin, 2008, S. 45-104.

[SS13] Schmelzer, H.; Sesselmann, W.: Geschäftsprozessmanagement in der Praxis. 8. ed., Hanser, München, 2013.

[SS08] Speck, M.; Schnetgöke, N.: Sollmodellierung und Prozessoptimierung. In (Becker, J.; Kugeler, M.; Rosemann, M., Hrsg.): Prozessmanagement, 6. ed., Springer, Berlin, 2008, 185-220.

[TNW12] Termer, F., Nissen, V.; Wessels, S: Grundlegende Überlegungen zum adäquaten Detaillierungsgrad von Geschäftsprozessmodellen. In (Rinderle-Ma S., Weske M. eds.): Proceedings EMISA, LNI 206, Bonn: GI, 2012, 53-65.

[WW08] Webster, J.; Watson, R.T.: Analyzing the past to prepare for the future. Writing a literature review.: In: MIS Quarterly 26 (2002) 2, Xiii- Xxiii.

Designing and Implementing a Framework for Event-based Predictive Modelling of Business Processes

Jörg Becker, Dominic Breuker, Patrick Delfmann, Martin Matzner

Department of Information Systems
University of Muenster - ERICS
Leonardo-Campus 3
48149 Münster
{becker,breuker,Delfmann,matzner}@ercis.uni-muenster.de

Abstract: Applying predictive modelling techniques to event data collected during business process execution is receiving increasing attention in the literature. In this paper, we present a framework supporting real-time prediction for business processes. After fitting a probabilistic model to historical event data, the framework can predict how running process instances will behave in the near future, based on the behaviour seen so far. The probabilistic modelling approach is carefully designed to deliver comprehensible results that can be visualized. Thus, domain experts can judge the predictive models by comparing the visualizations to their experience. Model analysis techniques can be applied if visualizations are too complex to be understood entirely. We evaluate the framework's predictive modelling component on real-world data and demonstrate how the visualization and analysis techniques can be applied.

1 Motivation

Enterprises increasingly invest in data analysis capabilities to gain competitive advantage [PF13]. Areas of application are manifold and numerous fields surrounding this topic emerged [CS12]. Not surprisingly, business process management (BPM) scholars have recognized the merits of data analysis long ago. Most importantly, process mining [Aa11] has emerged as a discipline developing methods for the analysis of business process event data collected from an organization's information systems. Typical process mining starts with discovering a process model and proceed with further analyses, for instance to eliminate problems with the process's implementation [ARW07].

Process mining is a discipline concerned mainly with techniques for retrospective data analysis [Aa11]. Hence, its techniques are applied with the same goals traditional business intelligence (BI) techniques are used—enabling knowledge workers to make good decisions fast [CDN11]—but the techniques are tailored to the BPM domain. However, BI is also increasingly used for operational support, i.e. in real-time settings [CDN11]. In BPM, complex event processing (CEP) software can be used to monitor events in real-time and trigger suitable actions when predefined patters are detected [EN10].

With analysis of the past and the present being important BPM topics, it is not surprising that analysis of the future is gaining momentum. Process mining scholars emphasize the opportunities of using event data to predict future events of running process instances, for instance to recommend appropriate actions [APS10]. Architectures to integrate event-driven process analytics with technologies such as CEP have been proposed [SMJ13]. Approaches to predict the completion time of process instances based on their current states have been developed recently [ASS11].

Motivated by the attention paid to predictive modelling in BPM, we present a framework for event-based predictive modelling of business processes. The framework requires the existence of historical event data and uses it to fit a probabilistic model. By feeding real-time data of running process instances into the framework, it allows evaluating probabilistically how the process will behave in the future. To implement this functionality, we draw on machine learning research [Mu12] and, in particular, grammatical inference, which is a discipline concerned with learning formal languages from data [Hi10]. The ultimate goal is to provide a technical basis for implementing process intelligence approaches capable of anticipating opportunities and threads before they occur, which gives managers a window of opportunity to take suitable actions. For instance, such functionality is a core part of modern intelligent BPM suites [JSC14]. These tools however require processes to be modelled explicitly, whereas our approach follows a process mining approach in which an appropriate model is learned from the data.

In predictive modelling projects, it is often necessary to convince domain experts with non-technical background that a predictive model will do more good than bad [PF13]. Hence, predictive modelling techniques should ideally produce comprehensible, interpretable models. Many techniques though do not lend themselves to interpretation. Thus, comprehensible models are sometimes used even if their predictive performance is worse than that of a black-box model [SK11]. We carefully design our framework such that the probabilistic models can be transformed into conceptual models in notations used frequently in BPM. This ensures that predictive models can be visualized in a form domain experts are familiar with. As a consequence, they can view and analyse these models to compare them with their domain knowledge.

The framework provides means to visualize predictive models at varying levels of detail. While this abstraction mechanism can provide a small, easy to read visualization at a high level of abstraction, a user may want analyse a visualization in more detail. These models can easily get too complicated to be understood. Fortunately, BPM scholars have long been concerned with managing and analysing large quantities of complex process models [Aa13]. For this reason, we integrate model analysis functionality into the framework. It allows domain experts to work even with complex models.

The remainder of this paper is structured as follows. Section 2 describes and justifies the design decisions we made when conceptualizing our framework. In section 3, we present results of an evaluation with real-world event data and illustrate the use of the visualization techniques with an example. Related work is discussed in section 4, while section 5 concludes and provides an outlook to future research.

2 Event-based Prediction Framework

2.1 Overview of the Framework

Before introducing the components of the framework in detail, we present an overview of its architecture (cf. figure 1). There are two data sources. The first is historical event data from finished process instances. The other source is real-time data from running process instances. Once a running process instance is completed, its event sequence should be added to the historical data. We use a simple data format that corresponds to event logs as they are used in process mining [Aa11]. Conceptually, an event log $X = \{x^{(1)}, \dots, x^{(C)}\}$ consists of C sequences of events. Each event sequence $x^{(c)}$ is a tuple $\left(x_1^{(c)}, \dots, x_{t_c}^{(c)}\right)$ of t_c events. Thus, t_c is the length of event sequence $x^{(c)}$, which may be different for each sequence. Each event uniquely belongs to an event type $x_t \in \{1, \dots, E\}$. There are E different event types.

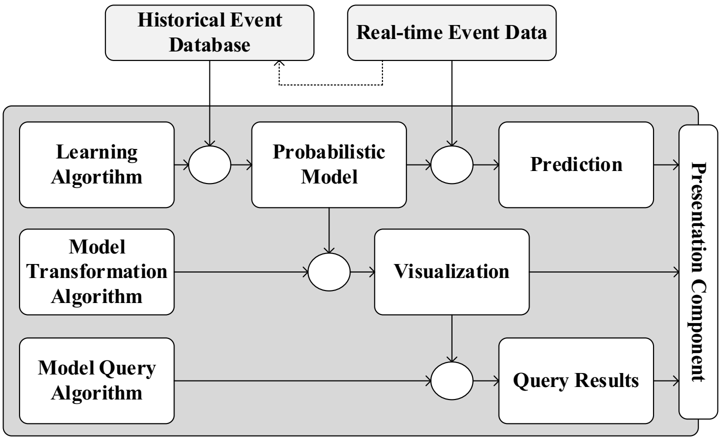

Figure 1: Overview of the Event-based Prediction Framework

Using the framework starts with selecting an appropriate subset of the event data. This is a creative, application-specific task and its support is not within the scope of our framework. Once a dataset is created, a learning algorithm is used to fit a probabilistic model that (hopefully) represents the event data well. The result delivered by the algorithm is a representation that can be used to make predictions. To this end, event data from running process instances must be fed into the framework. Given a sequence of events observed so far, the probabilistic model allows evaluating the probabilities of the process continuing with certain other event sequences. In particular, this functionality can be used to predict the most likely type of event with which a process will continue.

A user of the framework may want to make sure the predictive model does not contradict common sense. This analysis starts with deriving a conceptual model from the probabilistic model using a transformation algorithm. This conceptual model can be visualized and interpreted. It allows a user to compare the model with his expectations. As a result, he may either conclude that the probabilistic model is adequate or he may identify behaviour that contradicts his expectations. In the latter case, problem analysis could follow to find out if the probabilistic model is inadequate or if the expectations were wrong.

When the conceptual model generated by the model transformation algorithm is too complex to be read easily by humans, the framework provides algorithmic support to its users. In particular, model query algorithms can be used to determine whether certain patterns are found in the model. The patterns can represent model structure that a user either expects to find or expects not to find. By comparing this expectation with the query results, a model can be evaluated even if it is highly complex.

All components implemented are summarized in table 1 and described in detail in the following subsection. The framework is implemented entire in Java and is publicly available at https://github.com/DominicBreuker/RegPFA. It is meant to be used as a library and provides only rudimentary support for visualization. This paper describes the current state of the implementation and the rationale behind the framework's design, which should be understood as work in progress rather than a final version.

Table 1: Summary of the current components

Area	**Component**	**Description**
Learning algorithms	*EmMapLerner*	Learns a regularized PFA with EM-based MAP parameter estimation combined with grid search.
Model transformation algorithms	*Threshold pruning*	Visualizes a PFA at varying levels of detail.
	Automaton minimization	Simplifies the structure of an automaton obtained with *threshold pruning*.
	Petri net synthesis	Creates a petri net from an automaton and ensures that both models describe the same behavior.
Model query algorithms	*Subgraph pattern search*	Decides whether a subgraph pattern is part of a model created by a model transformation algorithm.

2.2 Learning Algorithms and Probabilistic Models

Given event data in the form described in section 2.1, modelling the event data probabilistically means finding a representation that specifies a probability distribution over the event sequences in the event log. Hence, the first design decision in our framework is to define what kind of probabilistic model should be used. Approximating distributions over sequences of symbols (~events) from a discrete alphabet (~set of event types) is a problem extensively researched in the field of grammatical inference [Hi10]. For this reason, grammatical inference constitutes the starting point of our discussion.

While numerous different probabilistic models are considered in the literature, the two best known are the Hidden Markov Model (HMM) and the Probabilistic Finite Automaton (PFA) [VEH13]. The commonality of both models is that the process generating the events is explained with a process of traversing an invisible state space. At each discrete step in time, the process is in exactly one state, and the event observed depends on that state. In the HMM [Ra89], the state space traversal is independent of the events, i.e., given a current HMM state, the next state depends only on the current state. A PFA [VTH05] is different. The next state depends on the current state and the current event.

For business processes, the PFA is more adequate than the HMM. To illustrate why, consider the simple example business process in figure 2 (a), represented as a business process model and notation (BPMN) diagram. Assuming that the performance of each BPMN activity corresponds to observing a corresponding event, the possible event sequences are *ABE*, *ACDE*, and *ADCE*. Their probabilities are 0.3, 0.35 and 0.35. A corresponding PFA is illustrated in figure 2 (b). By comparing the two models, it is obvious how the PFA's states can be interpreted as states of the BPMN process. For instance, the PFA can express that observing event *B* in state *s2* leads to state *s5* while observing *D* leads to *s3*. In a HMM, the states would not have this direct correspondence. For this reason, we decided to use the PFA as a probabilistic model.

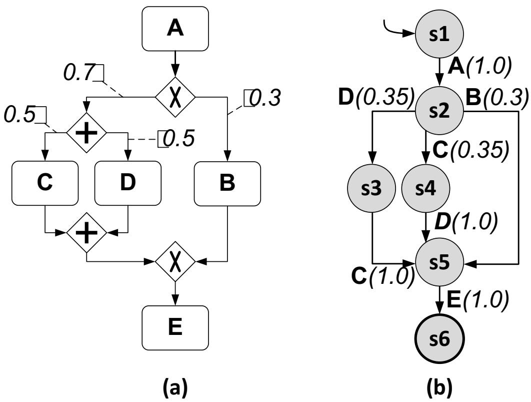

Figure 2: (a) shows a business process in BPMN notation with annotated probabilities. Annotations to the XOR-split represent probabilities of following the corresponding paths, while annotations to the AND-split represent probabilities of performing either *C* or *D* first. (b) shows a corresponding probabilistic automaton.

Numerous techniques for PFA learning have been developed in the literature. The authors of [VEH13] distinguish three types: state merging, Bayesian inference and parameter estimation. State merging techniques build a large PFA and iteratively merge states that are similar. Bayesian inference methods use sampling techniques to average over a large number of models. Parameter estimation uses maximum likelihood (ML) techniques to fit a standard PFA, usually with a fully connected state space.

In [VEH13], the authors also report results of a competition in which several methods have been benchmarked. The winning technique was based on Bayesian inference, directly followed by an ML parameter estimation approach. On first sight, these results suggest using Bayesian inference. However, prohibits model visualization as a huge number of models is drawn to average predictions over all of them. Since visualization is crucial to ensure comprehensibility, which is a key requirements of our approach, we resort to ML parameter estimation and defer Bayesian inference to future research.

ML estimation is prone to overfitting if the sample size is small. Small samples are considered a major challenge in working with event-based process data [AAM11]. Thus, precautions are necessary. We apply Bayesian regularization, which assumes that the

PFA's parameters are themselves random variables with a distribution [SJ02]. The standard choice for a PFA is the (symmetric) Dirichlet distribution, which is conjugate to the discrete distributions used in the remaining parts of the PFA [SJ02]. A user can modify the Dirichlet parameters to increase or decrease the tendency of the learning algorithm to output more evenly distributed parameters. Parameter estimation for models regularized in this way is called maximum a posteriori (MAP) estimation.

For PFAs, MAP estimation ca be done with the expectation maximization (EM) algorithm [DLR77], which was also used in [VEH13]. We have implemented a version of EM modified to work with a regularized PFA. We call this learning algorithm the *EmMapLerner*. A formal specification of the model can be found in equations (1) to (6). Equation (1) defines the distribution over the state in which the PFA is at the beginning of a process. Equation (2) defines the state-dependent distributions over the types of events observed at any point in time. Equation (3) defines the state- and event-dependent distributions over the states to which the PFA moves. Equation (4) to (6) define the corresponding Dirichlet distributions. K is the symbol denoting the number of states.

$$P(Z_0) \sim Discrete(\pi_0, \dots, \pi_K) \quad (1)$$

$$P(X_t | Z_t = k) \sim Discrete(b_{k0}, \dots, b_{kE}) \quad (2)$$

$$P(Z_t | Z_{t-1} = k, X_{t-1} = e) \sim Discrete(a_{ke0}, \dots, a_{keK}) \quad (3)$$

$$P(\pi_1, \dots, \pi_K) \sim Dirichlet(\rho_1, \dots, \rho_K) \quad (4)$$

$$P(b_{k1}, \dots, b_{kE}) \sim Dirichlet(s_{k1}, \dots, s_{kE}), \qquad \forall k \in \{1, \dots, K\} \quad (5)$$

$$P(a_{ke1}, \dots, a_{keK}) \sim Dirichlet(r_{ke1}, \dots, r_{keK}), \qquad \forall k \in \{1, \dots, K\}, e \in \{1, \dots, E\} \quad (6)$$

As K defines the dimension of the PFA, it is an input for EM. Determining appropriate values is known as the model selection problem [Mu12]. A standard approach is grid search, i.e., trying a range of values, fitting a model for each, and selecting the best of these candidate models [Mu12]. Applying grid search usually requires splitting the data into two separate parts, one to which the learning algorithm is applied, another to compare the candidate models. We have implemented grid search in the framework. It can also be used to systematically compare different degrees of regularization.

2.3 Model Transformation Algorithms and Visualizations

As the probabilistic model was chosen such that it can be interpreted as an automaton with a finite number of states and probabilities attached to the transitions, it is possible to visualize it graphically. For a given state k, b_{ke} is the (estimated) probability of observing an event of type e, and a_{kej} is the (estimated) probability of the PFA moving to state j if an event of type e is ovserved. Hence, $b_{ke}a_{kej}$ is the probability of the transition between the states k and j which is labeled with event type e.

Unfortunately, our EM learning algorithm estimates a transition probability for all the K^2E possible transitions of a fully connected PFA. A visualization will therefore be unreadable unless the number of event types and states is very small. Thus, abstraction is necessary to generate a useful visualization. A simple but effective approach is to discard all transitions with a probability below a given threshold. This way, only the most important parts are visualized. This is the basic model transformation approach implemented in our framework and we call it *threshold pruning*. It delivers a visualization such as the one shown in figure 2 (b). Users can generate visualizations with varying thresholds to view the probabilistic model at varying levels of detail.

Even when using thresholds, the visualization can be unnecessarily complex. Different automata can be equivalent in terms of the behavior they describe, yet their structure can be different. Most importantly, the number of states can vary, which is why automata minimization is a problem with a long tradition in computer science [HMU01]. Different standard algorithms exist and have been evaluated in the BPM literature [BDD14]. They are all equally effective as they transform any automaton into an equivalent one with the minimum number of states. The Hopcroft algorithm however turned out to be most efficient in terms of runtime complexity [BDD14]. Hence, we integrated *automaton minimization* based on Hopcroft's algorithm into our framework.

Additionally, we provide a third transformation approach extending the other two. In BPM, higher-level notations such as BPMN, Event-driven Process Chains (EPC), and Petri nets are most popular [Aa13]. Thus, domain experts may refuse to use low-level automata-based visualizations. Fortunately, techniques exist that can transform automata to higher-level representations. We included Petrify [CKK97] into our framework, which is a tool used as a component of a process discovery algorithm [ARV10]. It transforms an automaton-based visualization into a Petri net. We call this model transformation approach *Petri net synthesis*. It is particularly useful for highly concurrent processes since petri nets can represent concurrency more compactly than automata.

2.4 Model Query Algorithms

The decision whether to use predictive models operationally is often not made by technical staff alone. Typically, at least one manager is involved. Managers often demand to understand the models at least at a basic level [PF13]. With our model transformation approaches, different visualizations can be produced that visualize the inner workings of a PFA. However, depending on the complexity of the event data and the visualization's level of detail, a visualization may just be too complex to be understood easily. Particularly because threshold pruning comes along with information loss, users may demand analysis on a low level of detail. A visual analysis could be tedious in these cases.

The problem of working with large sets of complex business process models is relevant in the BPM domain, which motivated scholars to develop numerous model-based analysis techniques [Aa13]. In particular, model query techniques allow specifying a pattern and then decide whether a process model contains this pattern. For instance, a pattern can be a partial specification of behaviour such as "event B can follow on event A" [Aa13]. Queries of these kinds applied to our visualizations can be used to verify that a probabilistic model actually behaves in a way that external domain knowledge suggests. It allows specifying and testing assumptions about the process.

Many model query algorithms can be found in the literature, either based on established theory such as temporal logic and graph theory, or based on custom-made algorithms (for surveys, see [Aa13], [BDD14], [DRR12], [WJW14]). As our primary objective is to include a generic query approach that could be applied to any type of model created by a model transformation approach, we decided to implement a graph-based approach first. A wide range of graphical visualizations can be interpreted mathematically as graphs, which consist of nodes and edges between the nodes. Both nodes and edges can be labelled to encode any information required for the matching. Our model query algorithm is based on an established algorithm for subgraph matching [CFS04], which was evaluated in the BPM domain and recommended as the most efficient algorithm [BDD14]. We call this model query algorithm the *subgraph pattern search*.

3 Demonstration and Evaluation

3.1 Data

To illustrate how our framework can be used and to evaluate how effective it is for predictive modelling of business processes, we apply it to real-world event data collected at Volvo IT Belgium [Do13]. The event data describes the incident and problem management processes. As process execution is supported with a dedicated information system, different status changes for registered incidents and problems are available.

The incident management event log consists of 3777 process instances with a total of 36730 events. Each event belongs to one of 11 types. The types describe the incident's status (a*ccepted*, *queued*, or *completed*) and substatus (*awaiting assignment*, *assigned*, *in progress*, *in call*, *resolved*, *closed*, *wait*, *wait-user*, *wait-implementation*, *wait-vendor*, or *wait-customer*). Similarly, the problem management event log consists of 744 process instances with a total of 4045 events. Each event belongs to one of 5 types which describe the status (a*ccepted*, *queued*, or *completed*) and substatus (*awaiting assignment*, *assigned*, *in progress*, *wait*, or *closed*) of the problem. This event log only contains data about problems that are already closed. To each process instance in both event logs, we added one event of a special event type called *END* to make process completion explicit. After this modification, the event logs had 40507 events of 12 different types and 4789 events of 6 different types respectively.

3.2 Analysis

To evaluate how effective the *EmMapLerner* is in building predictive models, we used it to implement predictors for different prediction problems. The first is predicting the type of the next event in a running process instance (or predicting that no more events will follow, i.e., that the process is finished). The second problem is focused on specific types of events. For each type of event and given a running process instance, the task is to decide if the next event will be of this type.

To build a predictor, we used the framework to fit a probabilistic model to both the incident and problem management event logs. We applied the *EmMapLearner* with the following parameters. As discussed in section 3.2, the number of states and the degree of regularization must be determined with grid search. For the number of states, we considered $[2,4,6,8,10,12,14,16,18,20,30,40]$ as a set of candidate values. To express the degree of regularization, we make use of the interpretation of the (symmetric) Dirichlet distributions' parameters as pseudo-observations. This means that the degree can be specified relative to the number of events in the event logs [SJ02]. For instance, a degree of 0.1 can be interpreted as follows. For each of the event log's events providing evidence that any of the event type or state transition probabilities defined in equations (1) to (3) should be biased towards specific event types or transitions, 0.1 artificial events are added which provide evidence that all probabilities are equal. $[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]$ is the set of candidate degrees we used. For each combination, we ran the EM algorithm. It is an iterative algorithm converging to a local optimum. We stopped iterating once the increase of data's log-likelihood was less than 0.001. The local optimum found with EM is not necessarily a global optimum, which is why EM is usually run many times with different initial values [Mu12]. We repeated EM 5 times per combination of state number and regularization degree with randomly generated initial values to reduce the risk of hitting bad local optima.

It is good practice to evaluate predictors on held-out test data not touched when fitting predictors [Mu12]. Moreover, grid search requires to split the event logs into one part used for EM and another part used for model selection. Hence, we spitted both event logs into three parts. 50% were used for EM, 25% to select the best model, and the remaining 25% were held-out test data to evaluate predictive performance. To measure performance, we generated all possible prefixes of all process instances in the test data. We recorded the accuracy with which the type of the next event was predicted (#correct predictions / # of predictions) as well as the sensitivities (true positives) and specitivities (true negatives) of the predictors deciding whether a given event type will be next. The results can be found in table 2. Sensitivities and specitivities have been averaged over all event types to present compact results.

To establish a benchmark for the performance of our approach, we implemented a simple prediction technique based on n-grams. It was also used as a benchmark in the grammatical inference competition [VEH13]. For each consecutive sequence $x_1, \ldots, x_n$ of n evens, the next event's type is recorded. From the resulting frequency tables, probability distributions $P(x_{n+1}|x_1, \ldots, x_n)$ are built which can be used for prediction. n must be chosen arbitrarily. We considered values from one to five. The results of these benchmark predictors can also be found in table 2.

The results demonstrate that no predictor can accurately predict what will happen next. However, this is not surprising since given a running instance, there is likely some degree of uncertainty regarding the type of the next event. However, the predictors are all much better than a predictor returning random guesses would be. For instance, with 12 (6) different types of events, the expected accuracy of random guessing is 0.083 (0.167). All accuracies of the predictors in table 2 are much higher.

Table 2: Results of the evaluation of the predictive power. Results of the *EmMapLearner* and the best of the predictors based on the last n events are boldface.

Process	Predictor	Accuracy	Ø Sensitivity	Ø Specitivity
Incident management	**EmMapLearner**	**0.714**	**0.383**	**0.974**
	Last1	0.621	0.346	0.965
	Last2	0.633	0.368	0.966
	Last3	**0.635**	**0.377**	**0.967**
	Last4	0.631	0.378	0.966
	Last5	0.624	0.375	0.966
Problem management	**EmMapLearner**	**0.686**	**0.519**	**0.944**
	Last1	0.690	0.521	0.945
	Last2	**0.699**	**0.564**	**0.948**
	Last3	0.686	0.553	0.946
	Last4	0.680	0.543	0.944
	Last5	0.665	0.530	0.942

The *EmMapLearner* performed better than the best of the benchmark predictors on the larger and more complex incident management event log. This is true for all three performance measures. However, the situation changes when considering the smaller problem management event log. While the predictor delivered by the *EmMapLearner* outperforms some of the benchmark predictors, it also is outperformed by others. These results suggest that building predictors with simple techniques such as using the last n events can be more effective for some datasets than using a more complex technique fitting a PFA. However, the PFA has its merits when processes are complex and context information not contained in the last n events is important.

To demonstrate how model transformation and model query analysis can be applied to gain insights about a PFA fitted with the *EmMapLearner*, consider the example in figure 3. On the left, the result of applying *threshold pruning* to a PFA fitted on the problem management event log is illustrated. The threshold was set such that only transitions with more than 15% probability are shown. Transitions are labelled with the corresponding event types and probabilities.

The visualization allows characterizing the predictive model's structure. The process most likely starts with an event of type *Accepted(in progress)*, which can lead to two different states. In the first, the process either proceeds directly with *Completed(closed)* or there is an event of type *Accepted(wait)* observed before completion. In the second state, the process proceeds with *Queued(awaiting assignment)* and then either returns back into the initial state with the same two event types observed so far or proceeds with *Accepted(in progress)*, *Accepted(assigned)*, *Accepted(in progress)*, and *Competed(closed)*. This description of the process can now be given to the users of the problem management system so that they can compare it to their experience with the system. It allows them to judge the appropriateness of the probabilistic model. It is important to note though that this visualization represents only the PFA's most likely transitions. Clearly, the process may behave differently in rare cases.

With a model query approach, judging the appropriateness of a probabilistic model could be done in a different way. It could start with defining patterns such as the two illustrated on the right of figure 3. The upper pattern shall specify a situation in which there are two transitions, one labeled *Accepted(in progress)* and the other labeled *Accepted(assigned)*. The probabilities do not matter, which is indicated with the symbol *. The dashed line from the second to the third state shall indicate that any path through the automaton should be a match in this query. From a system user, it could be known that whenever *Accepted(in progress)* is observed, it often happens that *Accepted(assigned)* will be observed later. To find out if the probabilistic model reflects this, a visualization could be queried with this pattern.

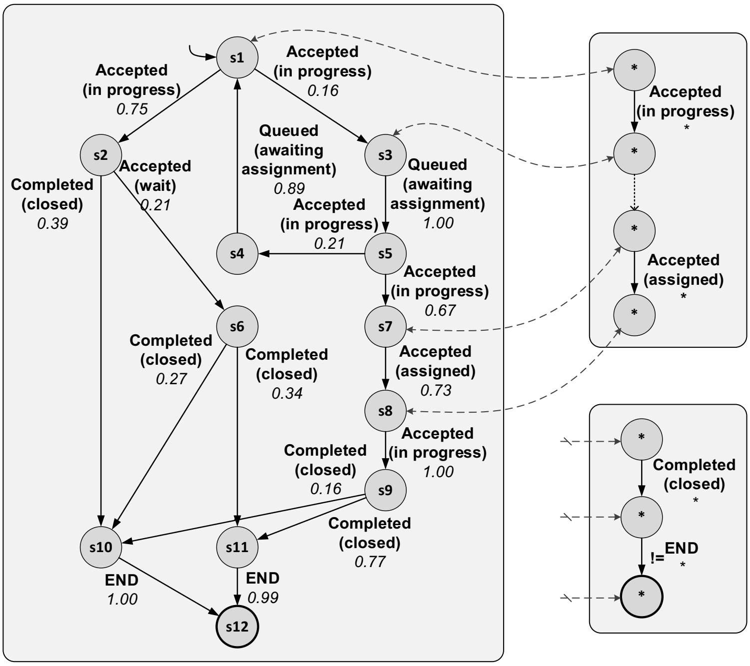

Figure 3: Visualization of a predictive model for the problem management process. Possible patterns for model queries can be found on the right.

Another possible pattern is illustrated on the lower right of figure 3. It matches if there is a transition labeled *Completed(closed)*, immediately followed by a transition labeled anything else than *END* (denoted with the *!=* modifier). It expresses a situation in which the process does not end after seeing the *Completed(closed)* event. Again, a model query algorithm can be used to ensure that there is actually no match for this pattern. It could be used even with very complex visualizations.

These two example patterns illustrate the limits of the *subgraph pattern search* algorithm, which is currently the only model query algorithm implemented in our framework. While the second pattern can be applied, the dashed line of the first pattern is a problem. Subgraph search does not support paths of arbitrary length. Hence, future versions of our framework require more extensive model query support.

4 Related Work

Predicting the future behaviour of running process instances is a topic in BPM research that has emerged only recently [APS10]. Thus, only few such approaches can be found in the literature. A notable example is presented by [ASS11], who use automata constructed with a process discovery technique to predict the time at which running process instances will be completed. Similar approaches for time prediction have been proposed as well (e.g., [RW13], [PNC11]). Our approach is different since its goal is to predict the events that are about to happen, not quantities such as time until completion.

There is also work on providing employees with recommendations. For instance, [SWD08] use an approach that compares running process instances with a set of (already finished) similar instances. Given a target to optimize, e.g., the total working time required for a process instance, the approach identifies the best action by predicting the outcomes of all actions available. Our approach is different since the goal is to predict what employees are likely to do, not what they should do to optimize a target function.

Similar to our approach is that of [LSD13]. Their goal is also to predict the event that will happen next, yet they consider different data. Their approach is tailored to case-oriented semi-structured business processes for which a large amount of context data is available. They characterize the current state of the process as all the data available and apply decision trees to predict the decisions employees will make. Another similar approach applying decision trees is presented in [Ro06]. Our approach is different since it assumes the existence of sequential event data.

In [RWA09], a simulation approach is presented which is designed to support operational decision making. By combining available process models from workflow management systems with operational event data, a simulation model is built and used to predict how running instances will behave. Our approach is different since it does not assume the existence of a process model.

Also related to our approach is the work of [Da98]. The author’s goal is not to predict future events but to discover process models. However, the probabilistic technique they use is based on the same principles we applied in section 3.2 to construct the benchmark predictors. Effectively, the model discovery technique developed in [Da98] could be used to visualize the benchmark predictors in form of automata.

5 Current Status and Future Research

In this paper, we presented a framework designed to support predictive modelling of business process event data. We integrated techniques from different fields of research, most importantly grammatical inference, process mining, and conceptual model analysis. The learning algorithm has been evaluated on real-world event data and was compared against a suitable benchmark technique identified in the literature. Results indicate that the approach is effective. How to use the other parts of the framework has been illustrated with an example based on the data used in the evaluation. It uncovered a shortcoming of the current model query component. Our framework is free to use and can be downloaded from https://github.com/DominicBreuker/RegPFA. We hope to encourage practitioners to use the approach and scholars to benchmark other approaches against it.

In the future, we plan to extend the framework in several areas. Since the evaluation of the *EmMapLearner* revealed that the benchmark predictors perform well and can even outperform our approach, including the benchmark predictors as learning algorithms is the next step. In [Da98], it is described how to construct a visualization using this approach, which is why it does not conflict with our requirement of easy visualization.

Furthermore, by applying the model transformation and model query approaches to the probabilistic model constructed in the experiments, we identified simple patterns that cannot be used with the model query algorithm that is currently implemented. Hence, a deeper analysis of the requirements will be conducted in order to prepare for a structured comparison of the more advanced model query approaches available in the literature.

References

[Aal11] van der Aalst, W. M. P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin / Heidelberg: Springer, 2011.

[Aal13] van der Aalst, W. M. P.; Business Process Management: A Comprehensive Survey. ISRN Softw. Eng., vol. 2013, pp. 1–37, 2013.

[AAM11] van der Aalst, W. M. P.; Adriansyah, A.; de Medeiros, A. K. A.; Arcieri, F.; et al.: Process mining manifesto. Lect. notes Bus. Inf. Process., vol. 99, pp. 169–194, 2011.

[APS10] van der Aalst, W. M. P.; Pesic, M.; Song, M.: Beyond Process Mining: From the Past to Present and Future. 22nd International Conference, CAiSE 2010, 2010, pp. 38–52.

[ARV10] van der Aalst, W. M. P.; Rubin, V.; Verbeek, H. M. W.; van Dongen, B. F.; Kindler, E.; Günther, C. W.: Process Mining: A Two-Step Approach to Balance Between Underfitting and Overfitting. Softw. Syst. Model., vol. 9, no. 1, pp. 87–111, 2010.

[ARW07] van der Aalst, W. M. P.; Reijers, H. A.; Weijters, A.; van Dongen, B.; de Medeiros, A. K.; Song, M.; Verbeek, H. M. W.: Business process mining: An industrial application. Inf. Syst., vol. 32, no. 5, pp. 713–732, 2007.

[ASS11] van der Aalst, W. M. P.; Schonenberg, M. H.; Song, M.: Time prediction based on process mining. Inf. Syst. J., vol. 36, no. 2, pp. 450–475, 2011.

[BDD14] Breuker, D.; Delfmann, P.; Dietrich, H.; Steinhorst, M.: Graph theory and model collection management: conceptual framework and runtime analysis of selected graph algorithms. Inf. Syst. E-bus. Manag., 2014.

[CDN11] Chaudhuri, S.; Dayal, U.; Narasayya, V.: An overview of business intelligence technology. Commun. ACM, vol. 54, no. 8, pp. 88–98, 2011.

[CFS04] Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 10, pp. 1367–72, 2004.

[CKK97] Cortadella, J.; Kishinevsky, M.; Kondratyev, A.; Lavagno, L.; Yakovlev, A.: Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. IEICE Trans. Inf. Syst., vol. E80-D, no. 3, pp. 315–325, 1997.

[CS12] Chen, H.; Storey, V. C.: Business Intelligence and Analytics: From Big Data to Big Impact. MISQ, vol. 36, no. 4, pp. 1165–1188, 2012.

[Da98] Datta, A.: Automating the discovery of AS-IS business process models: Probabilistic and algorithmic approaches. Inf. Syst. Res., vol. 9, no. 3, pp. 275–301, 1998.

[DLR77] Dempster, A.; Laird, N.; Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. R. Stat. Soc. Ser. B, vol. 39, no. 1, pp. 1–22, 1977.

[Do13] van Dongen, B.: BPI Challenge 2013. 2013. Retrieved from http://www.win.tue.nl/bpi/2013/challenge.

[DRR12] Dijkman, R.; la Rosa, M.; Reijers, H. A.: Managing Large Collections of Business Process Models - Current Techniques and Challenges. Comput. Ind., vol. 63, no. 2, pp. 91–97, 2012.

[EN10] Etzion, O. ; Niblett, P.: Event Processing in Action. Manning Publications Co., 2010.

[Hi10] de la Higuera, C.: Grammatical Inference. Cambride University Press, 2010.

[HMU01] Hopcroft, J. E.; Motwani, R.; Ullman, J. D.: Introduction to automata theory, languages, and computation. 2nd ed., Addison-Wesley Longman, 2001.

[JSC14] Jones, T.; Schulte, W.R.; Cantara, M.: Magic Quadrant for Intelligent Business Process Management Suites. 2014.

[LSD13] Lakshmanan, G. T.; Shamsi, D.; Doganata, Y. N.; Unuvar, M.; Khalaf, R.: A markov prediction model for data-driven semi-structured business processes. Knowl. Inf. Syst., 2013.

[Mu12] Murphy, K.: Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.

[PF13] Provost, F.; Fawcett, T.: Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media Inc., 2013.

[PNC11] Pandey, S.; Nepal, S.; Chen, S. A Test-bed for the Evaluation of Business Process Prediction Techniques. 7th International Conference, CollaborateCom, pp. 382–391, 2011.

[Ra89] Rabiner, L. R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[Ro06] Rozinat, A.: Decision Mining in ProM. 4th International Conference, BPM 2006, 2006, pp. 420–425.

[RW13] Rogge-Solti, A. ; Weske, M.: Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays. 11th International Conference, ICSOC 2013, 2013, pp. 389–403.

[RWA09] Rozinat, A.; Wynn, M. T.; van der Aalst, W. M. P.; ter Hofstede, A. H. M.; Fidge, C. J.: Workflow simulation for operational decision support. Data Knowl. Eng., vol. 68, no. 9, pp. 834–850, 2009.

[SJ02] Steck, H.; Jaakkola, T.: On the Dirichlet Prior and Bayesian Regularization. Neural Information Processing Systems, 2002, vol. 15, pp. 1441–1448.

[SK11] Shmuelli, G. ; Koppius, O. R.: Predictive Analytics in Information Systems Research. MISQ, vol. 35, no. 3, pp. 553–572, 2011.

[SMJ13] Schwegmann, B.; Matzner, M.; Janiesch, C.: A Method and Tool for Predictive Event-Driven Process Analytics. 11. Internationale Tagung Wirtschaftsinformatik (WI), pp. 721–735, 2013.

[SWD08] Schonenberg, H.; Weber, B.; van Dongen, B.: Supporting Flexible Processes through Recommendations Based on History. 6th International Conference, BPM 2008, 2008, pp. 51–66.

[VEH13] Verwer, S.; Eyraud, R.; de la Higuera, C.: PAutomaC: a PFA/HMM Learning Competition. Mach. Learn. J., 2013.

[VTH05] Vidal, E.; Thollard, F.; de la Higuera, C.; Casacuberta, F.; Carrasco, R. C.: Probabilistic finite-state machines - part I. IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 7, pp. 1013–1025, 2005.

[WJW14] Wang, J.; Jin, T.; Wong, R. K.; Wen, L.: Querying business process model repositories. World Wide Web, vol. 17, no. 3, pp. 427–454, 2014.

BPMN Extension for Business Process Monitoring*

Anne Baumgrass, Nico Herzberg, Andreas Meyer, Mathias Weske
Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2–3, D-14482 Potsdam, Germany
{Anne.Baumgrass,Nico.Herzberg,Andreas.Meyer,
Mathias.Weske}@hpi.de

Abstract: The execution of business processes generates a lot of data representing happenings (also called events) that may be utilized for process monitoring and analysis. This, however, is not supported by typical Business Process Management Systems (BPMSs). Especially, in manual executing business process environments, i.e., not driven by a BPMS, the correlation of events to processes for monitoring and analysis is not trivial. At design-time, Process Event Monitoring Points are used in process models to specify the locations, where particular events are expected. Therewith, occurring events can be assigned to a process during run-time. In this paper, we introduce an extension to BPMN, which implements this connection between process models and events. We show applicability of this extension by applying it to a logistics scenario taken from an EU project.

1 Introduction

Nowadays, the availability and creation of process-relevant information in terms of events[1] increases substantially, e.g. through the Internet of Things, Big Data as well as new and faster in-memory and data streaming technologies. Furthermore, sensors such as GPS receivers, RFID chips, transponders, detectors, cameras, or satellites enable the depiction of the current status of processes. Although the amount of events would lead to a fine-grained monitoring, mining, and decision support for business processes, a large number of business processes controlled by a BPMS operate without them. Especially, monitoring events of business processes not controlled by a single system and across enterprise boundaries is valuable for gaining insights about business process execution [Luc02, Luc11], e.g., to ensure a business process is executed as expected. However, enabling business process monitoring through events is not trivial [RLM+12, HMW13, DGD12].

Central to Business Process Management (BPM) [Wes12] are process models as they specify business processes and, therefore, capture the operations that need to be carried out to achieve a business goal. Therewith, process models are used, among others, for documentation, verification, enactment, process monitoring, and process analysis. Thus,

*The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement 318275 (GET Service).

[1] In this paper, we refer to events as real-world happenings (and not as the modeling elements in BPMN).

the event information of different sensors and produced by novel technologies need to be aggregated over these process executions to see the overall performance of a defined business process, e.g., the consumed resources, the time spend for the operations, and the arisen cost of the business process execution. Especially for process analysis but also for monitoring, it is valuable to have the process context, i.e., the process model, connected to the business process execution data – the events.

In [BBH+13a], we introduced an approach to allow model-driven business process monitoring by automatically creating Complex Event Processing (CEP) queries from appropriately annotated process models. In this paper, we present the corresponding extension to the Business Process Model and Notation (BPMN) metamodel, which formally describes the association of events to process models for business process monitoring and analysis. Associating event information to business processes is supported by the concept of Process Event Monitoring Points (PEMPs) [HMW13] that specify where and when which event is expected during business process execution. We extend BPMN to integrate PEMPs into process models verified by a prototypical implementation of the connection of PEMPs and process models.

The remainder of this paper is structured as follows. Section 2 motivates and demonstrates our BPMN extensions with a scenario from the logistics domain. We elaborate on event modeling in business process models in general in Section 3, before we present the BPMN extension with process event monitoring points to establish process monitoring and analysis in Section 4. Afterwards, an application and implementation of the BPMN extension is shown in Section 5. In Section 6, major related literature is discussed before we conclude this paper in Section 7.

2 Motivating Scenario

Especially for logistics, monitoring (resp. tracking and tracing) the complete logistics process from client to customer is essential to locate the shipment, reveal the progress in delivery, and to inform other stakeholders as well as the customer about arrivals and departures. Furthermore, monitoring a logistics process allows to replan and avoid penalties in case of deviations. However, the ongoing globalization including an internationalization of production and distribution [SP09] leads to more and more complex logistics processes that get harder to monitor because of the increasing amount of transportations and transshipments between countries, as well as the increasing amount of often distributed and loosely coupled systems to support these. Below, the example of a container pick-up handling is used to describe our approach as well as show the necessity of a BPMN extension enabling the connection of processes and event information.

Figure 1 shows a business process model for the container pick-up handling at a terminal modeled in BPMN. After receiving the container location, a driver needs to drive the corresponding truck to the pick-up location (activity `Drive to container location`) Afterwards, a container is assigned to the driver, which then needs to be checked for appropriateness (sufficient capacity, special capabilities as cooling support if required) and

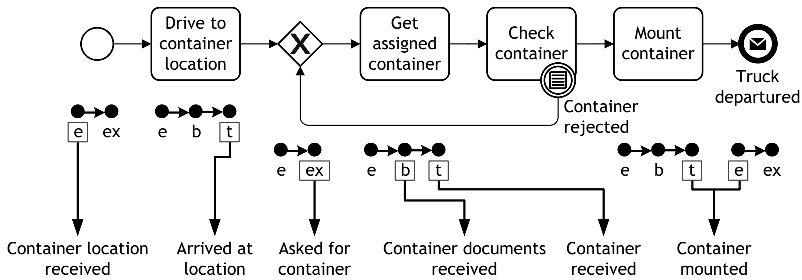

Figure 1: Business process model of a container pick-up process modeled in BPMN with a connection to event information via PEMPs associated with state transitions ((e)nable, (b)egin, (t)erminate, and (ex)ecute) occurring during node execution (see Section 3 for terminology).

mint condition (including tidiness and integrity). While the appropriateness checks are rather guaranteed due to the booking in advance of containers, the mint condition checks are critical and fail sometimes. If any of the checks fails, the activity `Check container` gets canceled and the corresponding intermediate event *Container rejected* is raised. This triggers another container assignment to the driver followed by another check. If all checks succeed, the driver mounts the container to the truck and departs from the pick-up location

An unobstructed process of the container handling is very important for the logistics company, as this is one of their core business processes. Any delays while picking up the container will affect schedules, contracts with customers, and partners as well as resources, i.e., drivers and truck availability. Therefore, it is necessary to monitor and analyze the process as detailed as possible. This allows for the measurement of activity execution times, delays, and resource consumption for instance. Establishing basic monitoring capabilities is possible in BPMN, however, the definition of process monitoring and analysis in a fine-grained and by the industry required manner is not possible. Establishing valuable process monitoring and analysis requires the capturing of events at certain locations within a process and the correlation of them to the process context specified by process models.

In this paper, we present a BPMN extension that implements the framework proposed by Herzberg et al. [HMW13] connecting event and process information via PEMPs associated to transitions describing state changes of nodes occurring during execution of a node For the example, we stick to simplified node life cycles. An activity may be in states enabled, running, or terminated with the corresponding transitions *(e)nable*, *(b)egin*, and *(t)erminate* while for gateways and BPMN events only transitions *(e)nable* and *(ex)ecute* may be triggered; for details about terminology, see Section 3. Potentially, each of these transitions may be connected to a PEMP [HKRS12] used for process observation. However, this does not mean that each transition is observable and therefore monitored during process execution.

In practice, the start of the process may be observed, i.e., the receipt of the location details can be tracked. Therefore, an event called 'Container location received' is associated to the transition *(e)nable* of the start event in the process model in Figure 1. For activity `Drive to container location`, the model associates Global Positioning System (GPS) coordinates aggregated to identify when the truck arrived at its destination, i.e., modelling when an activity termination will take place. In contrast, we model activity `Check container` as not observable, because it is a manual task without any IT system interaction. Likewise, the intermediate conditional event cannot be captured, because it represents a manual interaction between the driver and the pick-up location worker. However, indirectly, the happening of both may be derived whenever another container is requested. Altogether, at design-time, we model six different events as observable and correlate them to the correct locations in the process model and utilize this information for subsequent process analysis during process execution. If the PEMPs would have not been specified, events still occur, but correlation to the process (model) becomes a time-consuming and probably manual task depending on the information existent within the events.

3 Modeling Events in Business Processes Models

In this section, we formally introduce the concepts for Complex Event Processing (CEP) in the context of BPM and correlate both worlds. Figure 2 provides an overview about all concepts and their relations. The Process Event Monitoring Point (PEMP) is the connection between process models and events by relating events to transitions of the node life cycle defined for each node of a process model. The structure of a process model aligns with Definition 4 discussed below.

An event is a real-world happening occurring in a particular point in time at a certain place in a certain context [Luc02]. Processing an event requires to first store it in an information system. Thereby, the event gets transformed into an *event object* which structure is defined by its corresponding *event object type* . In accordance with [BBH+13a] and [HMW13], we define both concepts in the context of BPM as follows.

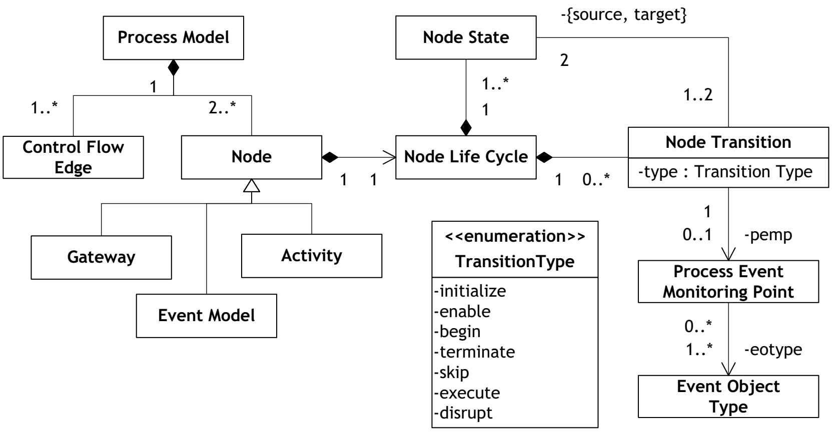

Figure 2: Model of relations for process models and events.

Definition 1 (Event object type) An *event object type* $\mathcal{ET} = (name, cd)$ refers to a unique *name* indicating the object type identifier and has a *content description* cd that specifies the structure of a particular event being of this event type. ◇

Definition 2 (Event object) An *event object* $\mathcal{E} = (\mathcal{ET}, id, P, timestamp, \mathcal{C})$ refers to an event object type $\mathcal{ET}$, has a unique identifier *id*, refers to a set P of *process instances* being affected by the event object, has a *timestamp* indicating the occurrence time, and contains an additional *event content* $\mathcal{C}$. ◇

Definition 3 (Event content and event content description) A *content description* $cd = \{(attributename, datatype)\}$ of an event object type $\mathcal{ET}$ is a collection of key-value-pairs specifying the available attributes by name *attributename* and by the data type *datatype* in which the corresponding attribute may be given. The datatype may be a complex, i.e., allowing hierarchical structures of attributes. For example, the coordinate which is identified by its latitude and longitude. Consequently, the *event content* $\mathcal{C} = \{(attributename, value)\}$ of an event object $\mathcal{E}$ is also a collection of key-value-pairs containing the specific *values* for the *attributename* that references the attribute specified in the corresponding event object type. ◇

Event objects exist on different levels depending on the information provided as described in [HMW13]. We refer to an event object containing at least one reference to a process instance as *process event*; the corresponding structure definition is named *process event type*. The execution of business processes is described by *process instances* which follow instructions specified in *process models*. As indicated above, events affect the execution of business processes in multiple ways. They may, for instance, trigger the execution of specific activities or indicate a specific state of the business process meaning that some activity has been successfully or unsuccessfully executed.

Definition 4 (Process model) A *process model* $M = (N, F, \psi)$ consists of a finite non-empty set $N \subseteq A \cup E \cup G$ of nodes being *activities* A, *event models* E, and *gateways* G $F \subseteq N \times N$ represents the *control flow relation* over nodes. Function $\psi : G \rightarrow \{xor, and\}$ assigns a type to each gateway. ◇

During process execution, each node follows its own *node life cycle* that specifies the states the node may be in and the transitions between pairs of states. The control flow relation and these node states influence each other such that the control flow triggers state transitions and some state transitions trigger the control flow. In the context of this paper, the life cycle L_A for activities consisting of states *initialized*, *ready*, *running*, *terminated*, *disrupted*, and *skipped* as well as the initial state visualized by a blank circle. These node states are connected by transitions *initialize*, *enable*, *begin*, *terminate*, *disrupt*, *skip*, and *(ex)ecute* For events and gateways, we utilize a subset of these states removing states *running* and *disrupted* and the transitions leading to them. Upon process instantiation, i.e., start of the process instance, all nodes transition into state *initialized*. If the control flow reaches a node, it transitions into state *ready*. Completing the execution of a node (reaching the state *disrupted*, *terminated*, or *skipped*) triggers the control flow. Formally, we define the underlying state transition net as follows.

Definition 5 (Node life cycle) A *node life cycle* $L = (S, T, \varphi)$ consists of a finite non-empty set S of node states and a finite set $T \subseteq S \times S$ of node state transitions. Let $\mathcal{L}$ be the set of all node life cycles defined for the nodes N of process model M. Then, there exists a function $\varphi : N \to \mathcal{L}$ assigning a node life cycle to each node $n \in N$ of M. $\diamond$

Process monitoring deals with capturing events based on node execution. We define process event monitoring points (PEMPs) [HKRS12] to specify on model level where we expect events to occur. Assigning PEMPs on node level is possible but results in issues identifying whether the start, end, or some other happening during node execution shall be captured Assume, there exist waiting times between the enablement and the start of an activity Then, the enablement of the activity does not indicate the immediate start (state *running*) Capturing the time passed between activity enablement and activity begin is only possible, if both state transitions are captured. Therefore, we assign each PEMP to one specific state transition in the life cycle of the corresponding node. We distinguish node state transitions into the ones observable by occurring events and the ones requiring the context of the process instance to deduce their triggering. For a given node of a process model, each state transition belongs to either group. As utilizing PEMPs for process monitoring is independent from the process instance execution, a PEMP can only be attached to directly observable node state transitions. A PEMP is defined as follows.

Definition 6 (Process event monitoring point) Let M be a process model, L a node life cycle, and $O_L \subseteq T_L$ the set of state transitions not requiring process instance information Then, a *process event monitoring point* is a tuple $PEMP = (M, n, t, et)$, where M is the process model it is contained in, $n \in N$ is the node of the process model it is created for, $t \in O_L$ is a state transition within the node life cycle L it is created for, and $et \in \mathcal{ET}$ is an event object type specifying the event object to be recognized. $\diamond$

Recapitulating the concepts introduced in this section, there exist two direct connections between CEP and BPM. At design-time, PEMPs specify where in the process model which event is supposed to happen by linking a node state transition and a process event type. At run-time, each event object is or can get related to a set of process instances by utilizing the framework described in [HMW13].

4 Extending BPMN with Process Event Monitoring Points

Typically, business process execution is monitored by BPMSs on the basis of a process model. In distributed environments, however, parts of this execution can only be monitored via CEP techniques. To combine BPM and CEP, process models must include the node life cycles and the definition of PEMPs (see Section 3).

A widely accepted standard for process modeling is BPMN. Although it does not provide native language elements to model node life cycles and PEMPs (see Definitions 5 and 6), it benefits from its organizational maintenance through the Object Management

Group (OMG), builds upon the standardized metamodel Meta Object Facility (MOF)[2], and provides its own extension mechanism [OMG11]. Thus, using metamodel extensions, we are able to transform process models to platform-specific models and code.

In this paper, we provide a BPMN extension definition for modeling the life cycle and PEMPs as defined in Section 3 for each node in a BPMN model. This extension can, for example, be used to transform a process model to CEP-specific code for event detection and, thus, enable the monitoring of events in distributed systems that are associated with a single business process (cf. [BBH$^+$13a]).

In particular, we extend the BPMN metamodel based on MOF to define new domain-specific language concepts used to serialize BPMN models in the XML Metadata Interchange (XMI) specification without contradicting the semantics of any BPMN element. This BPMN metamodel extension implements node life cycles and PEMPs in BPMN models. Figure 3 illustrates our extension (shown in white) to the existing BPMN metamodel elements (shown in gray). We associate a life cycle with a *FlowNode* of the BPMN specification This life cycle includes states and transitions that can be associated with a PEMP that is defined by an event object type (see Section 3).

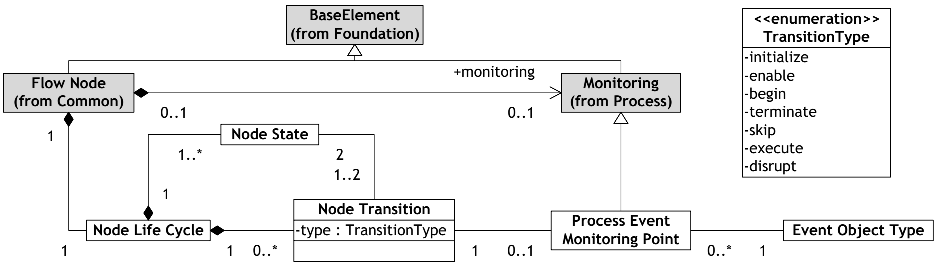

Figure 3: BPMN metamodel extension for including PEMPs and life cycles in BPMN models.

Complementary, in a BPMN specification, new attributes and elements have to be specified in an *ExtensionDefinition* that can be bound to a model definition by an *Extension* to any *BaseElement* (see [OMG11]). As this paper concentrates on the general concept of automating the derivation of CEP queries from business models, we use BPMN and its own eXtensible Markup Language (XML) format for demonstration purposes. This is also because the transformation from BPMN's own extension mechanism to a direct extension of the BPMN metamodel and MOF's own interchange format called XMI can be found in [SCV11b].

For the representation of BPMN models, BPMN defines a set of XML Schema documents specifying the interchange format for BPMN models. Thus, to model and exchange BPMN models that include nodes' state transitions and PEMPs, we derive an own XML Schema for our extension (see Listing 1). In particular, we developed a BPMN+X model [SCV11b] to specify our extension as shown in Figure 4. It allows the attachment of a transition element to a *FlowNode*. This transition element references a PEMP and defines its type by enumeration (see Figure 5). In this way, any state transition of an activity, gateway, or event

[2] http://www.omg.org/mof/

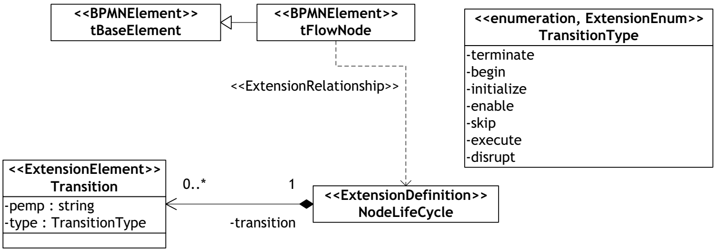

Figure 4: BPMN+X model for including PEMPs and life cycles in BPMN models.

models including a PEMP that references an event type can be represented in a BPMN model.

Applying the approach described in [SCV11b] to our model shown in Figure 4 generates the XML Schema that is conform with BPMN's extension definition shown in Listing 1. The node life cycle is an *ExtensionDefinition* introduced as <xsd:group> element that includes a transition (*ExtensionAttributeDefinition*) as <xsd:element> (Listing 1 Lines 6-10). The type of a transition is externally defined by an <xsd:complexType> element (Listing 1 Lines 12-15). Finally, we added the literals for node's transitions as enumeration (Listing 1 Lines 16-21) that can be made available for the type attribute definition in the transition itself. Although possible, we did not explicitly specify the event types that can be bound to a transition as we assume those are defined differently in each CEP environment However, we allow referencing the event type via its identifier (e.g., its name) for a binding of event types in CEP implementation (see Section 5).

Listing 1: XML Schema for the definition of PEMPs and life cycles in BPMN models.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsd:schema
3   xmlns:cep="http://www.extensions.bpmn/cep/deriviation"
4   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5   targetNamespace="http://www.extensions.bpmn/cep/deriviation">
6   <xsd:group name="NodeLifeCycle">
7    <xsd:sequence>
8     <xsd:element ref="transition" maxOccurs="unbounded" minOccurs="0"/>
9    </xsd:sequence>
10  </xsd:group>
11  <xsd:element name="transition" type="tTransition"/>
12  <xsd:complexType name="tTransition">
13    <xsd:attribute name="pemp" type="xsd:string" use="required"/>
14    <xsd:attribute name="type" type="TransitionType" use="required"/>
15  </xsd:complexType>
16  <xsd:simpleType name="TransitionType">
17   <xsd:restriction base="xsd:string">
18    <xsd:enumeration value="initialize"/>
19    ...
20   </xsd:restriction>
21  </xsd:simpleType>
22 </xsd:schema>
```

Finally, using our XML schema definition we can define BPMN-conform models in XML format. The transitions, we defined above via our schema, are added as elements in a

BPMN-element through its `<extensionElements>`. For instance, Listing 2 shows an example of a BPMN task and its two transitions *begin* and *terminate* for which the PEMP requires events from the event types `ContainerDocsReceived` and `ContainerReceived` respectively to monitor the state transitions of this task. These event types must be available in the CEP system of usage. An example derivation of a query is shown in the next section

Listing 2: Definition of a PEMP for a BPMN activity with XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
  xmlns:cep="http://www.extensions.bpmn/cep/deriviation"
  xsi:schemaLocation="http://www.extensions.bpmn/cep/deriviation CEPDerivation.
      xsd"
  targetNamespace="">
  <import importType="http://www.w3.org/2001/XMLSchema" location="CEPDerivation.
      xsd"
    namespace="http://www.extensions.bpmn/cep/deriviation" />
  <extension mustUnderstand="true" definition="cep:NodeLifeCycle" />
  <process id="Process_1">
    <task id="Task_2" name="Get assigned container">
      <extensionElements>
        <cep:transition pemp="ContainerDocsReceived" type="begin" />
        <cep:transition pemp="ContainerReceived" type="terminate" />
      </extensionElements>
      ...
    </task>
    ...
</definitions>
```

5 Experimental Evaluation

While the scenario in Section 2 introduced the benefits of the modeling extension at design time, this section discusses how we use the extension during process execution of the scenario for monitoring the current progress of its process instances. In particular, we implemented a service that is able to import the BPMN models specified in Section 4. The service decomposes the BPMN models and generates interleaving CEP queries with which we are able to monitor business process executions. Furthermore, this service may be used to detect execution violations related to a process model specification. This service is directly implemented in the Event Processing Platform (EPP) [HMW13, BBH$^+$13b][3]. In general, the EPP provides services to capture real-world events from different sources, to process these events, e.g., by aggregation or transformation, to correlate the events to its corresponding business process, and to provide and to manage these events for event consumers, e.g., business process monitoring applications. Especially for the event correlation to the right business process and the distribution of the events to the responsible

[3]The whole project is currently under development. Downloads, tutorials, and further information of the stable version can be found at: `http://bpt.hpi.uni-potsdam.de/Public/EPP`. To see the preliminary application of this paper, we refer the interested reader to our screencast available at `https://www.youtube.com/watch?v=doAFKwIEp6w`, starting from minute 7:37.

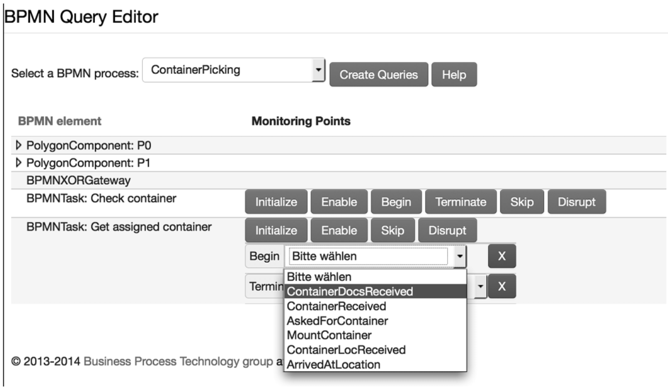

Figure 5: EPP implementation showing the manipulation opportunities of PEMPs in a BPMN process

party of this business process, the platform not only stores event object types but also business process models and event queries.

A series of actions is necessary to use our extension for event processing and business process monitoring in the EPP. First, the user has to define the available event object types by name and its content description (see Definition 1). These event object types are determined by the real-world event sources that the EPP is connected to. The EPP is able to upload event object type descriptions defined as XML Schema. Second, event types must be correlated to group event objects to process instances. This is usually done by using one or more attributes defined in the content description of an event object type. In the introduced example in Section 2, this grouping to process instances could be based on the driver's identification and the truck she drives. It is also possible to define such a group based on the booking number of the transportation order that each event object references. Subsequently, the BPMN model must be created. In this BPMN model, each BPMN element that we want to monitor should include the definition of its state transitions including the event object types referenced in its PEMP as specified in Section 4.

The EPP enables the import of business process models specified in the BPMN-conform XML format including PEMPs. Furthermore, the user gets the possibility to change the association of event types to state transitions of each imported BPMN model directly in the user interface of the EPP as shown in Figure 5. Thus, we are able to upload arbitrary BPMN-specific models and even associate event objects types that are stored in the EPP to PEMPs in a specific BPMN model. Next, the uploaded models can be used to derive CEP queries. Queries in the EPP are implemented using the Event Pattern Language (EPL) provided by Esper [BV07]. The EPP registers each CEP query in Esper via listeners. These listeners get informed if the query matches observed events with the specified conditions defined by the query. For instance, Listing 3 shows an example query in Esper query language for our scenario process model in Section 2; in total, four queries could be derived

As not every activity is observable, the derived queries are restricted to those events that are observable. Listing 3 shows the *St1* query that monitors the sequence of two monitoring

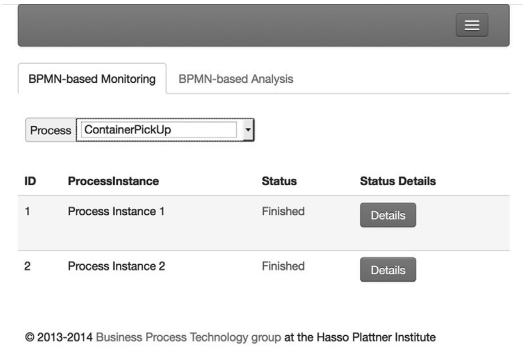

Figure 6: EPP implementation showing the monitoring of process instances.

points with the event types of the `Get assigned container` activity.

While the definition of the sequential ordering of the events for this query is enclosed in `PATTERN[...]` in the `FROM`-clause, the `WHERE` clause checks whether the events from both event types have occurred in the same process instance. In particular, this query is activated for each *ContainerDocsReceived* event object imported in the EPP that is followed by a *ContainerReceived* event object belonging to the same process instance. Similar queries are generated for all other PEMPs of the example (cf. [BBH+13a]). Altogether, we are now able to monitor all instances of the process and their status as shown in Figure 6.

Listing 3: Example of a monitoring query using Esper.

```
St1:
SELECT *
FROM PATTERN [((EVERY S0=ContainerDocsReceived -> EVERY S1=ContainerReceived)]
WHERE SetUtils.isIntersectionNotEmpty({S0.ProcessInstances , S1.ProcessInstances})
```

6 Related Work

With the proposed BPMN extension, the basis for high quality process monitoring and analysis is established. Besides process monitoring and analysis, prediction, control, and optimization are further major disciplines of Business Process Intelligence (BPI) [MR06, GCC+04]. [MR06] describes a reference architecture for BPI containing an integration, a functional, and a visualization layer. However, business process models are not integrated into that architecture. We argue that correlating events and the process monitoring and analysis results need to be done in the context of the process, which is described with process models. Process mining [vdA12] discovers this in the same direction by profiling process models from event logs using process discovery techniques [vdAea12] and enriching that models by execution information. Process Mining is utilizing information that is available in a well-defined event log, while our BPMN extension allows the definition of PEMPs that describe which event respectively information is required at which point in the process for process monitoring and analysis purposes.

Another area of BPI focuses on Business Activity Monitoring (BAM). [DWC11] presents an overview of BAM and gives a four class-categorization of BAM systems: pure BAM, discovery-oriented BAM, simulation-oriented BAM, and reporting-oriented BAM. All of these classes benefit from the BPMN extension as the resulting process events will have high information content and BAM techniques and methods can be applied to these process events to provide valuable monitoring results.

Del-Río-Ortega et al. [dRORRC10] present the concept of Process Performance Indicators (PPI), the process related form of key performance indicators, to enable process evaluation They propose a metamodel called PPINOT that supports the graphical notation of PPIs with process models, e.g., modeled with BPMN [dRORCRC13].

Furthermore, we focused on the automation of BPI with the help of CEP. [BPG12] introduces techniques to automatically generate Esper queries by taking a choreography model as a formalization of the process. In [WZM+11], BPMN models are taken as a basis to create EPL statements to monitor process violations but not to monitor the process execution in general as it is our goal. Barros et al. [BDG07] present a set of patterns describing relations and dependencies of events in business processes that have to be captured in process models to observe the overall process context. Their assessment of the modeling languages BPMN and Business Process Execution Language (BPEL) resulted in their language proposal called Business Event Modeling Notation (BEMN) [DGB07], a graphical language for modeling composite events in business processes. BEMN allows to define event rules, e.g., specific combinations of events, that are to be used in stand-alone diagrams or as integration into BPMN. In the CEP context, Rozsnyai et al. [RLM+12] introduce an approach to monitor the execution of semi-structured processes. Our extension of the metamodel complements these approaches by providing an interchange format that may be used by various systems to combine BPM and CEP.

Several approaches have extended BPMN to represent their domain specific requirements [RFMP07, ZSL11, SCV11a]. For example, Rodríguez et al. [RFMP07] present a metamodel extension of the diagram specification of BPMN 1.0 in order to model security requirements, while Stroppi et al. [SCV11a] specify resources in business processes using BPMN's own extension mechanisms. In addition, Stroppi et al. [SCV11b] presented a generic approach to transform metamodel extensions into BPMN's own interchange format Furthermore, Zor, Schumm, and Leymann [ZSL11] graphically extend BPMN models for the manufacturing domain. Complementary, our approach used the tools presented in [SCV11b] and extended the BPMN metamodel to consider CEP concepts.

7 Conclusion

In this paper, we introduced a BPMN extension to facilitate the concept of Process Event Monitoring Points (PEMPs) within process models. PEMPs are utilized to specify at which place in the process model a specific event is expected. Applying this concept, process monitoring and analysis based on events occurring during business process execution is enabled. This is important, especially in manual executing business process environments, as the event information is usually not directly correlated to the business process execution In essence, we used PEMPs to relate the domains of BPM and CEP. Based on the proposed BPMN extension, we modeled and implemented a scenario taken from the logistics domain and have shown the applicability and value of the approach.

In future work, we will investigate how PEMPs can be graphically represented in BPMN models and how this can be supported with appropriate implementations. Furthermore, we will apply the BPMN extension to process monitoring and analysis tasks in different domains, e.g., to measure run-time or process cost analysis. Further, we will adapt the approach to analyse process event occurrences.

References

[BBH+13a] M. Backmann, A. Baumgrass, N. Herzberg, A. Meyer, and M. Weske. Model-driven Event Query Generation for Business Process Monitoring. In *International Conference on Service Oriented Computing Workshops*. Springer, 2013.

[BBH+13b] S. Bülow, M. Backmann, N. Herzberg, T. Hille, A. Meyer, B. Ulm, T. Y. Wong, and M. Weske. Monitoring of Business Processes with Complex Event Processing. In *Business Process Management Workshops*. Springer, 2013.

[BDG07] A. Barros, G. Decker, and A. Grosskopf. Complex Events in Business Processes. In *Business Information Systems*. Springer, 2007.

[BPG12] A. Baouab, O. Perrin, and C. Godart. An Optimized Derivation of Event Queries to Monitor Choreography Violations. In *International Conference on Service Oriented Computing (ICSOC)*. Springer, 2012.

[BV07] T. Bernhardt and A. Vasseur. Esper: Event Stream Processing and Correlation. Published at `http://onjava.com/` by O'Reilly Media, 2007.

[DGB07] G. Decker, A. Grosskopf, and A. Barros. A Graphical Notation for Modeling Complex Events in Business Processes. In *Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2007.

[DGD12] M. Daum, M. Götz, and J. Domaschka. Integrating CEP and BPM: How CEP realizes Functional Requirements of BPM Applications (Industry Article). In *Distributed Event-Based Systems (DEBS)*, 2012.

[dRORCRC13] A. del Río-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortés. On the Definition and Design-time Analysis of Process Performance Indicators. *Information Systems*, 38(4), 2013.

[dRORRC10] A. del Río-Ortega, M. Resinas, and A. Ruiz-Cortés. Defining Process Performance Indicators: An Ontological Approach. In *On the Move (OTM)*. Springer, 2010.

[DWC11] A. Dahanayake, R.J. Welke, and G. Cavalheiro. Improving the Understanding of BAM Technology for Real-time Decision Support. *International Journal of Business Information Systems (IJBIS)*, 7(1), December 2011.

[GCC+04] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M. Shan. Business Process Intelligence. *Computers in Industry*, 53(3), April 2004.

[HKRS12] N. Herzberg, M. Kunze, and A. Rogge-Solti. Towards Process Evaluation in Non-automated Process Execution Environments. In *Services and their Composition (ZEUS)*, 2012.

[HMW13] N. Herzberg, A. Meyer, and M. Weske. An Event Processing Platform for Business Process Management. In *Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2013.

[Luc02] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2002.

[Luc11] D. C. Luckham. *Event Processing for Business: Organizing the Real-time Enterprise* John Wiley & Sons, 2011.

[MR06] B. Mutschler and M. Reichert. Aktuelles Schlagwort: Business Process Intelligence *EMISA Forum*, 26(1), January 2006.

[OMG11] OMG. Business Process Model and Notation (BPMN), Version 2.0, 2011.

[RFMP07] A. Rodríguez, E. Fernández-Medina, and M. Piattini. A BPMN Extension for the Modeling of Security Requirements in Business Processes. *IEICE transactions on information and systems*, 90(4), 2007.

[RLM+12] S. Rozsnyai, G. T. Lakshmanan, V. Muthusamy, R. Khalaf, and M. J. Duftler Business Process Insight: An Approach and Platform for the Discovery and Analysis of End-to-End Business Processes. In *SRII Global Conference (SRII)*. IEEE, 2012.

[SCV11a] L. J. R. Stroppi, O. Chiotti, and P. D. Villarreal. A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models. In *XIV Iberoamerican Conference on Software Engineering*, 2011.

[SCV11b] L. J. R. Stroppi, O. Chiotti, and P. D. Villarreal. Extending BPMN 2.0: Method and Tool Support. In *Business Process Model and Notation*, volume 95. Springer, 2011

[SP09] Frank Straube and Hans-Christian Pfohl. *Trends and Strategies in Logistics - Global Networks in an Era of Change: Environment, Security, Internationalisation, People* DVV Media Group, Dt. Verkehrs-Verlag, 2009.

[vdA12] W. M. P. van der Aalst. Process Mining: Overview and Opportunities. *ACM Transactions on Management Information Systems (TMIS)*, 3(2), 2012.

[vdAea12] W. M. P. van der Aalst and et al. Process Mining Manifesto. In *Business Process Management Workshops*. Springer, 2012.

[Wes12] M. Weske. *Business Process Management: Concepts, Languages, Architectures. Second Edition*. Springer, 2012.

[WZM+11] M. Weidlich, H. Ziekow, J. Mendling, O. Günther, M. Weske, and N. Desai. Event-Based Monitoring of Process Execution Violations. In *Business Process Management*. Springer, 2011.

[ZSL11] S. Zor, D. Schumm, and F. Leymann. A Proposal of BPMN Extensions for the Manufacturing Domain. In *Proceedings of the 44 th CIRP International Conference on Manufacturing Systems*, 2011.

Process Management Technology

Towards Schema Evolution in Object-aware Process Management Systems

Carolina Ming Chiao, Vera Künzle, Manfred Reichert

Institute of Databases and Information Systems
University of Ulm, Germany
{carolina.chiao, vera.kuenzle, manfred.reichert}@uni-ulm.de

Abstract: Enterprises want to improve the lifecycle support for their businesses processes by modeling, enacting and monitoring them based on process management systems (PrMS). Since business processes tend to change over time, process evolution support is needed. While process evolution is well understood in traditional activity-centric PrMS, it has been neglected in *object-aware* PrMS so far. Due to the tight integration of processes and data, in particular, changes of the data and process schemes must be handled in an integrated way; i.e., the evolution of the data schema might affect the process schema and vice versa. This paper presents our overall vision on the controlled evolution of object-aware processes. Further, it discusses fundamental requirements for enabling the evolution of object-aware process schemas in PHILharmonicFlows, a framework targeting at comprehensive support of object-aware processes.

1 Introduction

Aiming at improved process lifecycle support, a decade ago, many researchers started working on process schema evolution. In general, business processes may evolve for several reasons; e.g, due to changes in the business, technological environment, or legal context [RW12]. Consequently, business process changes need to be rapidly mapped to the process-aware information system (PAIS) implementing these processes.

Activity-centric process management systems (PrMS) like YAWL [vdAtH05] and ADEPT [RD98, RRD04, RW12] already provide comprehensive process lifecycle support, including the controlled evolution of business processes. Regarding object-aware processes [Kün13], however, this does not apply yet. Due to the tighter integration of process and data, changes of the data and process schemes must be handled in an integrated way. In other words, changing the data schema may affect the schema of an object-aware process and vice versa. Note that respective dependencies might become complex when taking different levels of process granularity as well as authorization constraints into account as well.

The example below is based on a real educational scenario. It comprises a process for managing extension course projects. Extension courses target at professionals that want to refresh and update their knowledge in a certain area. In order to propose a new extension course, the course coordinator must create a corresponding project description. The latter must then be approved by the faculty coordinator and the extension course committee.

Example 1 (Object-aware Process: Extension course proposal). The course coordinator creates an extension course project using a form. In this context, he must provide details about the course, like name, start date and description. Following this, professors may start creating the lectures of the extension course. In turn, each lecture comprises study plan items, which describe the topics to be covered by the lecture. After creating the lectures, the coordinator may request an approval of the extension course project. First, an approval must be provided by the faculty director. If he wants to reject the proposal, the extension course must not take place. Otherwise, the project is sent to the extension course committee, which will evaluate it. If there are more rejections than approvals, the extension course project is rejected. Otherwise, it is approved and hence may take place in future.

The process from Example 1 can be characterized by its need for *object-awareness*; i.e., business processes and business objects must not be treated independently from each other. In general, *object-aware processes* show three major characteristics. First, they are based on *two levels of granularity*. On the one hand, the *behavior* of individual object instances needs to be considered during process execution; on the other, the *interactions* among different object instances must be taken into account. Second, process execution is *data-driven*; i.e., the progress of a process depends on available object instances as well as the values of their attributes. Third, *flexible activity execution* is crucial. In particular, activities need not always coincide with process steps.

The PHILharmonicFlows framework we are developing targets at a comprehensive support of object-aware processes [KR09b, KR09a, KR11, Kün13]. It comprises modules for the modeling, execution and monitoring of object-aware processes. In this framework, *object behavior* is captured through *micro processes*. In turn, *object interactions* are captured by a *macro process*. Furthermore, data is modeled separately from micro and macro processes. Note that each of these models comprises different *components*. For example, a data model comprises object types as well as their attributes and relations to other object types.

Schema evolution has neither been considered by PHILharmonicFlows nor other frameworks for artifact-based or object-aware processes yet. As a major challenge, one must cope with the complex interdependencies that exist between the models and components (e.g., data model, object types, attributes, or micro process types) of the framework; i.e., changing one component (e.g., deleting an object attribute) may require concomitant changes of other components (e.g., changing the behavior of the object type). Moreover, changes must be handled at both the static and dynamic (i.e., instance) level. Changing an object-aware process without any user assistance will be error-prone and time-consuming. Therefore, user interactions should be properly supported in order to guide the modeler when changing an object-aware process. In particular, any guidance must hide complexity from users, taking correctness constraints and component dependencies into account.

This paper presents requirements necessary to enable schema evolution for object-aware processes. To illustrate how these requirements were derived, we sketch our vision on how the user should interact with the PHILharmonicFlows tool when changing an object-aware process. Sect. 2 provides an overview of the PHILharmonicFlows framework. Sect. 3 presents research questions to emphasize the scope of our work. In Sect. 4, we introduce our vision on how the user (i.e., modeler) should be supported when evolving object-aware processes. Sect. 5 presents major requirements emerging in this context. Sect. 6 discusses the related work and Sect. 7 gives a summary and outlook.

2 The PHILharmonicFlows Framework

The PHILharmonicFlows framework enforces a modeling methodology governing the object-centric specification of business processes based on a well-defined formal semantics [KR11, Kün13]. In general, an *object-aware process schema* comprises the following sub-schemas: data model (cf. Fig. 1a), micro process types (cf. Fig. 1b), macro process types (cf. Fig. 1d), and authorization settings (cf. Fig. 1c). In turn, each sub-schema comprises a set of components (e.g., a data model comprises object types, object type attributes and relations to other object types), which may be related to components of other sub-schemas (e.g., a micro step type depends on an attribute of an object type). When changing components, hence, concomitant changes of dependent components become necessary as well.

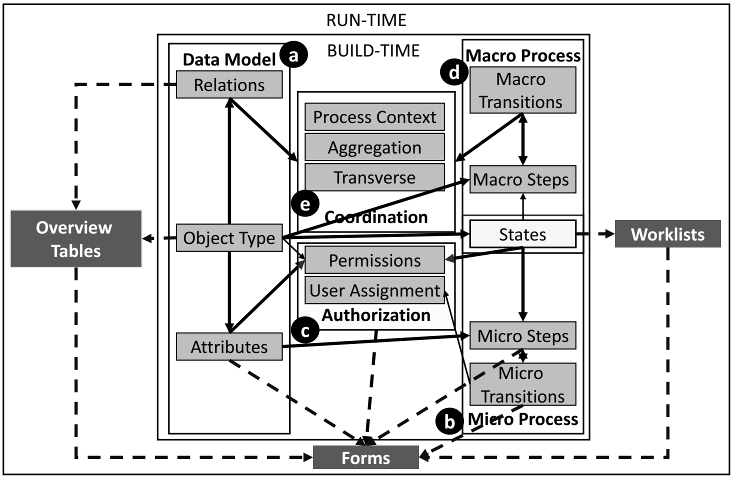

Figure 1: The PHILharmonicFlows framework

As a fundamental prerequisite, object types and their relations need to be captured in a *data model* (cf. Fig. 1a). Furthermore, for each *object type*, a corresponding *micro process type* needs to be specified (cf. Fig. 1b). The latter defines the behavior of related object instances, and consists of a set of *micro steps* as well as the *transitions* between them. In turn, each micro step is associated with an *object type attribute*. Further, micro steps are grouped in object *states*. At run-time, for each *object instance*, a corresponding *micro process instance* is created. A micro process instance being in a particular state may only proceed if specific values are assigned to the object instance attributes associated with this state; i.e., *data-driven process execution* is enabled. In addition, *optional data access* is accomplished asynchronously to micro process execution based on the permissions granted for reading or writing object attributes. In this context, access rights for an object instance depend on the progress of the corresponding micro process instance as well. Altogether, the framework maintains an *authorization table* assigning data permissions to user roles which may also depend on the respective state of the micro process type (cf. Fig. 1c).

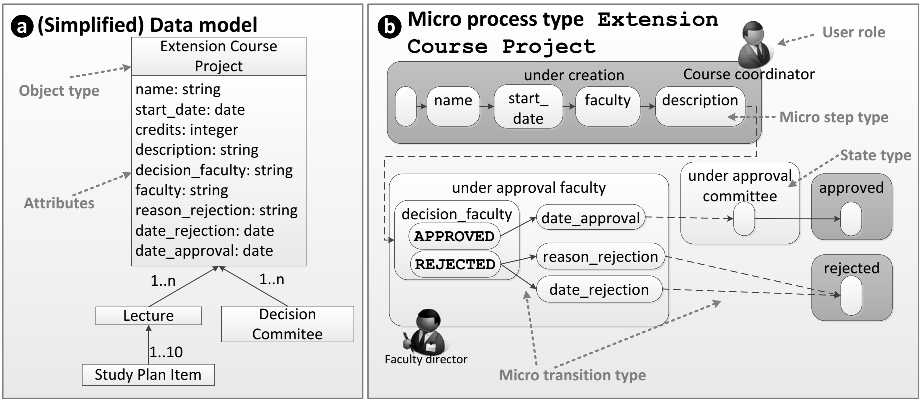

Figure 2: Data model and micro process type modeled with PHILharmonicFlows

Taking the relations between the object instances of the overall *data structure* (i.e., the instance of a data model) into account, the corresponding micro process instances form a complex *process structure*; i.e., their execution needs to be coordinated according to the given data structure. In PHILharmonicFlows, this is accomplished by means of macro processes. A *macro process type* consists of *macro steps* linked by *macro transitions* (cf. Fig. 1d). Opposed to micro steps, which refer to single attributes of a particular object type, a macro step refers to a particular state of an object type. In addition, for each macro transition, a coordination component must be specified (cf. Fig. 1e). The latter hides the complexity of large process structures from modelers as well as end-users. More precisely, such a coordination component coordinates the interactions among the object instances of the same type as well as different types. Opposed to existing approaches, the semantic relations between the object instances and their cardinalities are also taken into account.

Figs. 2 and 3 show how Example 1 can be modeled based on PHILharmonicFlows. Micro process type `Extension course project` is derived from the object type having same name (cf. Figs. 2a+b). In this micro process type, the `course coordinator` must write attributes `name`, `start_date`, `faculty`, and `description`. Note that this configuration needs to be reflected in the authorization settings (cf. Fig. 3a). Furthermore, attribute `credits` may be *optionally* written. In turn, a macro process type (cf. Fig. 3b) is composed of macro step types. The latter reference state types from the micro process types. For example, macro step type `Extension Course Project - under creation` refers to state type `under creation` of micro process type `Extension Course Project` (cf. Fig. 2b).

3 Research Questions

To emphasize the scope of our problem, we consider the following research questions:

Research Question 1 (RQ1): How to change an object-aware process schema without violating correctness neither of the modified component itself nor any dependent compo-

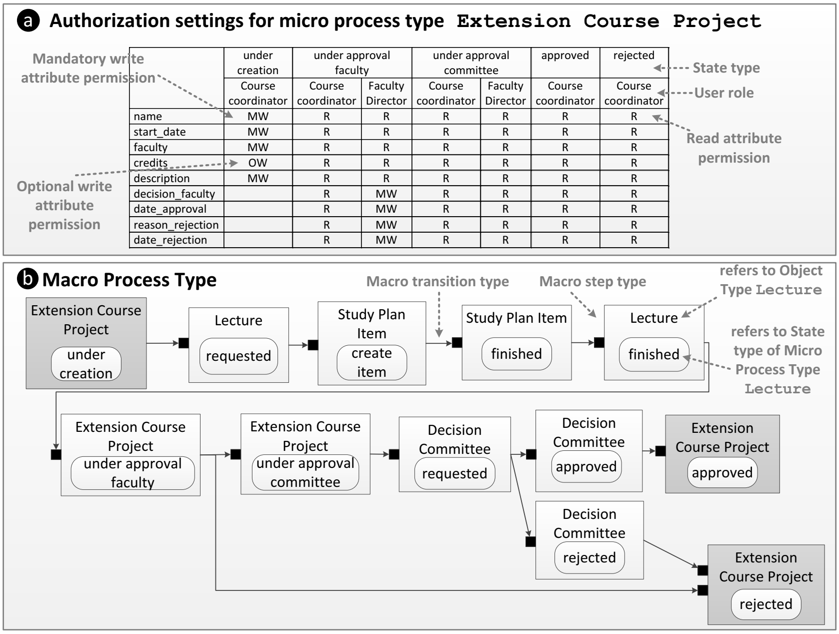

Figure 3: Authorization settings and macro process type modeled with PHILharmonicFlows

nents?

Research Question 2 (RQ2): How to handle active instances (i.e., object and micro process instances) when evolving the object-aware process schema?

Research Question 3 (RQ3): How to assist users in evolving an object-aware process schema?

RQ1 refers to changes at the static level. It deals with structural changes of an object-aware process schema; i.e., its sub-schemas and their components. In turn, RQ2 addresses issues related to dynamic changes; i.e., managing different schema versions and adopting the best policy to migrate active instances to the new schema version. Finally, RQ3 deals with user issues, such as providing user guidance while hiding the complexity of schema changes from them. These research questions guide our vision discussed in Sect. 4. Further, they serve as starting point for eliciting requirements related to the evolution of object-aware processes.

4 Overall Vision

To illustrate the scope of our research, we define a number of scenarios (i.e., user stories) dealing with schema changes of our sample process. While some scenarios are rather simple, not requiring any concomitant change, others are more complex involving several schemas of the object-aware process. Thereby, a major challenge concerns user interaction as changing an object-aware process schema constitutes an error-prone and complex task. Hence, user guidance is required to assist users when changing the schema of an object-aware process, e.g., by indicating the components affected by an intended change. For

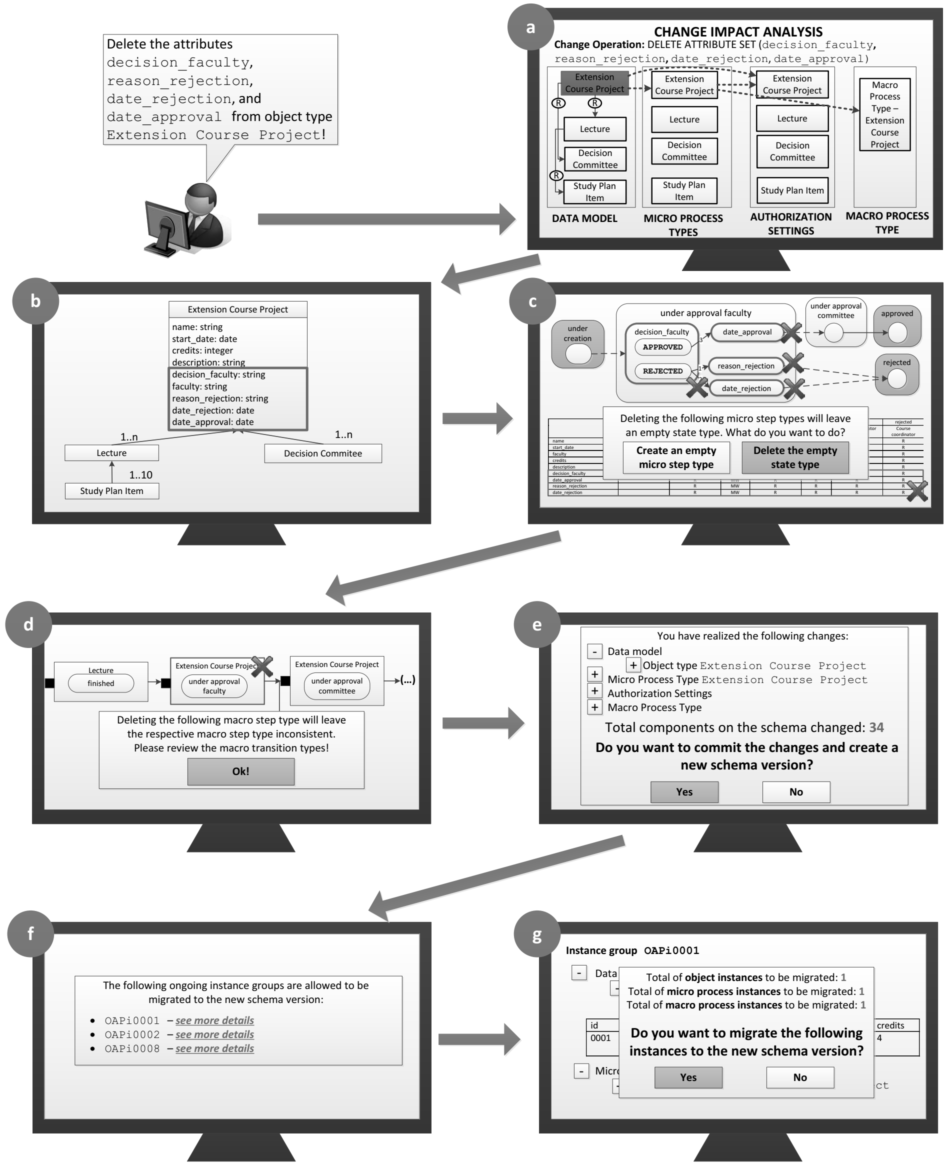

Figure 4: Sketch of end-user guidance for object-aware process schema changing

illustration purpose, we provide a mockup showing how to guide a user in the context of a concrete change scenario.

Change scenario (SC): The `extension course project` needs not be approved by the `faculty director` anymore, and the overall schema shall be adapted accordingly.

Since for an extension course project the approval of the faculty director is no longer needed, the user wants to delete the attributes referring to it (e.g., `decision_faculty`, `reason_rejection`, `date_rejection`, and `date_approval`); i.e., he wants to change the data model. In turn, this requires concomitant schema adaptations. First of all, the user should be notified about the effects of the change; i.e., a *change impact analysis* is required. Based on such an analysis, it can be visualized which sub-schemas and components are affected by the change and how they are related. Regarding our mockup (cf. Fig. 4a), the object-aware process schema is divided into four levels corresponding to the sub-schemas data model, micro process types, authorization settings, and macro process type. The dotted arrows represent potential effects caused by the change. In the given scenario, changing object type `Extension Course Project` affects the corresponding micro process type as well as the authorization settings. Changing the micro process type `Extension Course Project`, in turn, may further affect the macro process type.

Being aware of the possible effects of the intended change, the user may then be guided in performing required concomitant changes (cf. Fig. 4b). According to PHILharmonicFlows (cf. Sect. 2), micro steps of a particular micro process are directly related to the attributes of the corresponding object type. Therefore, when an attribute is deleted, the corresponding micro step type needs to be deleted as well. In Fig. 4c, in turn, the deletion of the respective micro step types will result in an "empty" state type. According to the correctness constraints of the framework, this is not possible, and would leave the micro process type in an inconsistent state. Therefore, the user should be notified about this problem and guided in resolving it. In Fig. 4d, the user decides to delete the entire state type, which, in turn, causes another inconsistency in the macro process type. Again, the user should be notified about this, enabling him to redefine the macro transition types that link the respective macro step type. In general, every time the user changes a component, respective correctness checks should be performed automatically in order to be able to guide the user in resolving potential inconsistencies.

Since changes of one component might trigger changes of others, the user will not always be aware of the number of components actually changed. Hence, after guiding him through required adaptations of the object-aware process, an overview of all components to be changed should be provided; e.g., such overview could present information about the components to be changed as well as quantitative metrics (e.g., number of components and models to be changed) (cf. Fig. 4e). Finally, the user should explicitly commit the changes, resulting in a new version of the object-aware process schema.

In addition to structural adaptations and structural consistency, active instances must be taken into account; i.e., it should be possible to adapt the running instances according to the changed object-aware process schema. Fig. 4f presents the active instances that may be migrated to the new schema version without causing any run-time error. To foster visualization, instances are grouped according to the underlying object type; i.e., each instance group refers to one particular object type. In our example, the instances refer to an `extension course project`. Then, the user may choose which group of instances he wants to migrate. Further, he may retrieve more detailed information about the respective instances (cf. Fig. 4g). Finally, like in the context of model changes, the user will get an overview of the instances to be migrated.

5 Requirements

Based on our research questions, the sketched vision, and an extensive literature study, we derived major requirements. The requirements of Sect. 5.1 are related to RQ1, while the ones of Sect. 5.2 are related to RQ2. Finally, the requirements of Sect. 5.3 are related to user guidance issues (i.e., RQ3).

5.1 Structural Changes at the Static Level

Requirement 1 (Change primitives). To accomplish structural adaptations of an object-aware process schema, change primitives are required to directly operate on single schema elements. In our context, such primitives denote atomic operations like `add attribute type`, `delete micro step type`, and `add state type`. In general, the set of available change primitives should be *complete* and *minimal* [CCPP98, RD98, RW12]. Completeness means that the available set of change primitives shall allow transforming any object-aware process schema `S` into any other object-aware process schema `S'`. In addition, the core set of provided change primitives should be minimal; i.e., it should not contain any primitive that can be simulated through the combination of other primitives. Finally, for each change primitive, a precise definition of parameters, pre-conditions, and post-conditions (i.e., effects) is required [CCPP98].

Example 2 (Requirement 1: Change primitives). For removing micro step type `date_rejection` of state type `under approval faculty`, change primitives for deleting the micro step type and its related micro transition types are required (cf. Fig. 5).

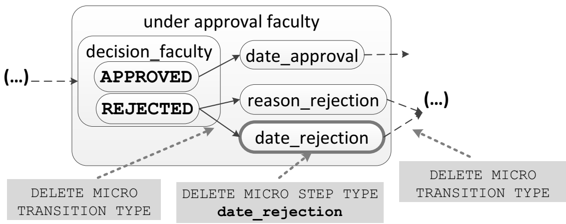

Figure 5: Example of change primitives

Requirement 2 (Cascading effects). Fig. 6 presents the meta model of PHIlharmonicFlows, expressed in terms of an UML class diagram. The dependencies between the different components of the framework are represented as bidirectional associations, compositions and aggregations. Regarding the class diagram, the *bidirectional association* represents components linked with each other in the context of a particular model (e.g., micro step types are linked with micro transition types within a micro process type). A *composition dependency* indicates a "strong" association between components, making one component (i.e., parent component) responsible for the creation and destruction of other components (i.e., child components). For example, an object type is strongly associated with attribute types. If the object type is deleted, all related attributes must be deleted as well. In turn, the *aggregation dependency* constitutes a "weaker" relationship between components: even when deleting the parent component, the child components will not be removed. An example of an aggregation dependency is provided by the relationship between the micro step types and attributes. If a micro step type is deleted, the associated attribute is preserved. Due to these dependencies and associations among different components of the framework, changing one of them might require changes of dependent components as well. In turn, such concomitant changes might again trigger additional changes on other components (i.e., a *cascading* change). In general, mechanisms are required to detect necessary concomitant changes and to guide the user in applying them.

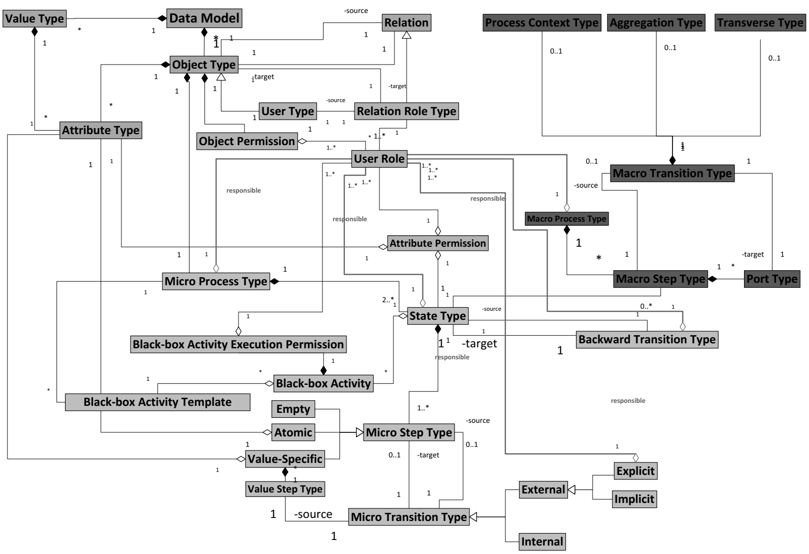

Figure 6: PHILharmonicFlows meta-model

Requirement 3 (Change operations and change patterns). Realizing structural adaptations based on change primitives might introduce errors and inconsistencies. Usually, at such a low level of abstraction, the combined application of several change primitives is required to ensure schema correctness. As an alternative, high-level change operations may be used; e.g., it should be possible to move an entire state type within a micro process type based on a single change operation. Like change primitives, high-level change operations should have pre-conditions. Generally, empirically-grounded change patterns should be defined, which capture the semantics of frequent changes, thus raising the level of abstraction [WRRM08, RW12].

Example 3 (Requirement 3: Change operations). Deleting state type `under approval faculty` (cf. Fig. 5) requires deleting all micro step types associated with this state type as well (i.e., `decision_faculty`, `date_approval`, `reason_rejection`, and `date_rejection`). Further, this deletion includes the micro transition types linking the micro steps and the authorization settings of state type `under approval faculty`. In this context, a change operation allowing for the deletion of the entire state type together with its components would facilitate change definition significantly, and hence reduce errors and inconsistencies of the object-aware process schema.

Requirement 4 (Complex changes). When adapting an object-aware process schema, the integrity and consistency of the various sub-schemas must be preserved; i.e., the changes applied to the sub-schemas will only be applied if this does not result in any inconsistency. Like for database transactions, the changes applied jointly to an object-aware process schema must be treated atomically (i.e., as transaction). Accordingly, modelers must explicitly *commit* complex changes. Finally, multiple users may want to change the same schema version at the same time, requiring proper *concurrency control*.

Requirement 5 (Change traceability). When changing an object-aware process, information on *who* applied *which* changes, *when* and *why* shall be recorded in logs; i.e., change traceability needs to be ensured.

Requirement 6 (Correctness). Changing an object-aware process schema must not result in errors in any of the sub-schemas and not lead to soundness violations (e.g., deadlocks due to data inconsistencies or missing data at run-time). Moreover, a changed object-aware process schema must comply with the correctness criteria established in [Kün13]. Correctness checks are required at two different stages. First, when specifying the various changes of an object-aware process schema, correctness checks are "soft"; i.e., they provide basis to inform the modeler about potential inconsistencies or missing components. Second, correctness needs to be ensured when committing a change transaction; i.e., all sub-schemas forming an object-aware process schema must be correct.

5.2 Changing Active Instances at the Dynamic Level

Requirement 7 (Versioning support). Active instances whose processing started before the schema change must be properly handled. One strategy frequently applied in the context of database and process evolution, is *schema versioning* [Rod96, GdSEM05, KG99]. Every time a schema is changed, a new schema version is created; already active instances continue their processing based on the old schema version. In our context, there are various sub-schemas forming the overall object-aware process schema (i.e., data model, micro and macro process schemas, and authorization settings). Hence, for each object-aware process schema version, the versions of its sub-schemas need to be maintained (cf. Fig. 7). In particular, the instances are linked to a sub-schema version as well as the object-aware process schema version.

If a change is performed, which concerns only a part of the entire object-aware process schema, creating a new version of all sub-schemas involved (even the unchanged ones) will

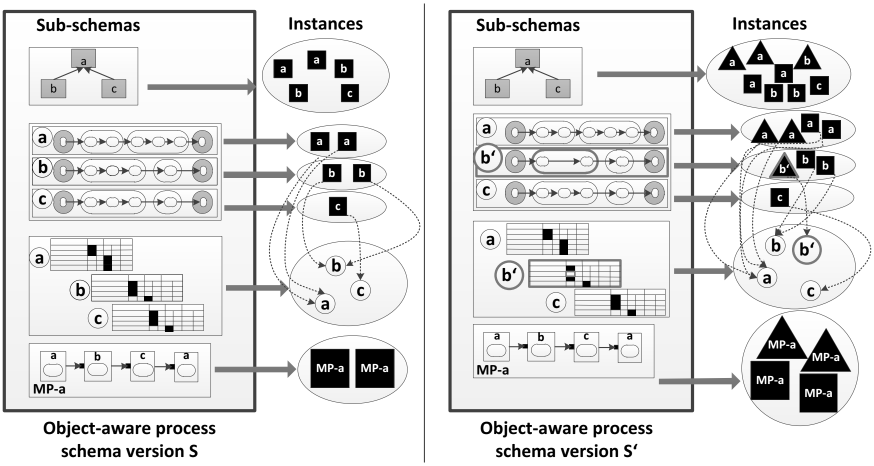

Figure 7: Object-aware process schema versions

not be optimal. Hence, a new version of an object-aware process schema shall comprise the new versions of the changed sub-schemas in combination with the versions of the unchanged sub-schemas. Fig. 7 illustrates this concept. The object-aware process schema S on the left side contains a data model with object types A, B and C. Each object type is associated with a micro process type as well as authorization settings. The latter express who may access which attributes at which stages during process execution. Finally, macro process type MP-a describes the interaction of the object types. The small squares represent the instances created according object-aware process schema version S and being active at the moment. On the right side, a new version S′ of S is depicted; it resulted due to a change of micro process type b. The latter led to a new version b′ of b and a change of related authorization settings. Instead of generating copies for all sub-schemas, S′ comprises the new version b′, the new version of the respective authorization settings, and references to the versions of the unchanged sub-schemas. New instances run according to the new schema version S′, while the older, but still active instances continue running on the old schema version S; i.e., instances running on the two schema versions will co-exist. In Fig. 7, new instances are represented as triangles and old ones as squares.

Requirement 8 (Instance migration). To ensure that active instances may continue running on the old schema version is not sufficient. In addition, it shall be possible to reassign active instances to the new object-aware process schema version if desired. Like in activity-centric PrMS [CCPP98, JH98, RD98, RRMD09], such migration of active instances must be handled in a controlled manner. In general, not all instances can be migrated to the new schema version, particularly if they have progressed too much in their execution. Since there may be numerous concurrently running instances of an object-aware process (i.e., object and micro process instances), the selection of the migratable instances should not handle the instances individually. In the example from Fig. 8a, a new state is inserted in micro process type A. Moreover, the progress of the instances of micro process B now depends on the execution of micro process A; i.e., the instances of B will only reach state s5 if all instances of A′ reach state s7 (cf. Fig. 8b). However, not all active instances of micro process `A` can be migrated to `A'`. More precisely, micro process instances `A1` and `A2` have already completed their executions, which means that they cannot be migrated to `A'`. In turn, micro process instance `A3` may be migrated. However, the individual migration of `A3` will cause a *deadlock* at run-time, since micro process instance `B1` will continue waiting for instances `A1` and `A2` to reach state `s7`. Therefore, a group of instances associated to a particular changed component must not be migrated individually.

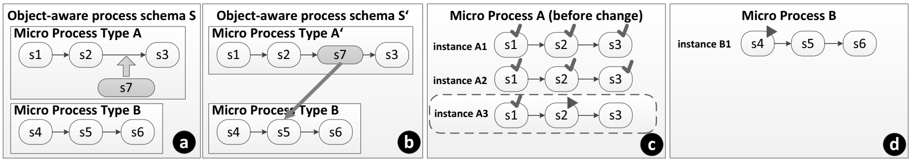

Figure 8: Example of instance migration

Requirement 9 (Data consistency). One of the biggest issues concerning database schema evolution is to prevent data loss when changing a database schema [Ra04]; i.e., the deletion of object types or attributes must not delete the data associated to them, since there may be software systems that still depend on this data. In the context of object-aware processes, data inconsistency might cause run-time errors (e.g., deadlocks). Therefore, it becomes necessary to prevent data inconsistencies (e.g., data loss or missing data important to the logic of the process) relevant for process execution.

5.3 User Requirements

Requirement 10 (User guidance). Changing an object-aware process schema is a non-trivial task from the viewpoint of the user (i.e., process modeler). Our experiences with the change scenarios have shown that user guidance is required to hide this complexity and hence to make schema changes more intuitive and less error-prone (cf. Sect. 4). Moreover, user guidance not only eases the adaptation of an existing schema, but also the modeling of new object-aware process schemas.

Requirement 11 (Change impact analysis and metrics). To better control potential costs of a change, a *change impact analysis* should be performed *before* actually applying the change. Such an analysis must consider the cascading effects; i.e., it must show to users which components are going to be affected by the change. Moreover, metrics help users to evaluate change complexity.

6 Related Work

The described requirements have been partially addressed by existing work. Fig. 9 summarizes which requirements have been addressed by which approach. We investigated data-centric approaches and traditional activity-centric ones.

Data-centric Approaches

Data-driven Process Coordination (COREPRO) [MRH07, MRH08] presents a set of change primitives and operations to change both data and process structures. Since the latter are directly related, the approach automatically adapts the process structure when chang-

+ supported ***o partially supported*** ***- not supported***	COREPRO	Artifact-centric processes	Product-based workflow	FLOWer	ADEPT	YAWL
Req. 1 (Change primitives)	+	+	o	+	+	+
Req. 2 (Cascading effect)	+	+	o	-		
Req. 3 (Change operations and change patterns)	+	+	-	+	+	-
Req. 4 (Complex changes)	+	-	-	+	+	+
Req. 5 (Change traceability)	+	-	-	o	+	+
Req. 6 (Correctness)	+	-	+	o	+	+
Req. 7 (Versioning support)	o	-	-	+	+	+
Req. 8 (Instance migration)	o	-	-	-	+	-
Req. 9 (Data consistency)	o	-	-	-		
Req. 10 (User guidance)	o	-	-	-	+	-
Req. 11 (Change impact analysis and metrics)	-	o	-	-	o	-

Figure 9: Evaluation of different approaches

ing the corresponding data structure. Moreover, it enables change traceability as well as change transactions. Correctness is ensured when changing the structures at static or instance level. In case of inconsistencies, the modeler is notified accordingly. However, even though the data objects are explicitly represented, the control of the data structures is still realized outside the scope of the PrMS. Hence, versioning support and instance migration is only available for the process structures.

Regarding artifact-centric processes, [WW14] proposes an approach for dealing with the change impact analysis of three-level artifact-centric business processes (ACBP). The authors first classify the types of changes that may be applied to an ACBP. This classification provides the basis for the change analysis. For this analysis, a graph representing the different element dependencies is created. Based on this graph, it becomes possible to calculate the direct impact of a change. The approach, however, just covers the changes at static level, without addressing the problems of complex changes and traceability. In turn, [XSY+11] allows for ad-hoc changes on the artifacts' life cycles. Such changes are based rules and declarative constructs such as *skip*, *add* and *replace* and are applied to the tasks. The artifacts, in turn, cannot be changed. Besides, the authors do not provide information on how the active instances should be handled when a change is applied.

In the context of product-based workflows, [RVV10] presents four change primitives enabling changes of the product (i.e., data) structure. These are then reflected in the corresponding process models without need for any manual adaptation. Changes at the process level, however, are not considered. Moreover, information regarding change traceability and run-time issues are neglected.

The case-handling system FLOWer [vdAWG05, MWR08] does not address cascading effects; i.e., inconsistencies in a dependent component caused by a change are not properly handled by the system. Moreover, the system does not use formal correctness criteria in the context of schema evolution [WRRM08]. Regarding schema versioning, FLOWer allows for overwriting a process schema as well as for the co-existence of instance running on different schema versions. Instance migration is not considered.

Activity-centric Approaches

In activity-centric approaches, the processes are described on a single level (i.e., the process model). Additionally, data is managed outside the scope of the PrMS. For these reasons, requirements regarding cascading effects (Req. 2) and data consistency (Req. 9) were not addressed to the analyzed approaches. ADEPT [RD98, RRD04, RW12] is an activity-centric PrMS that enables both schema evolution and ad hoc changes of single process instances. Moreover, it focuses on the *ease of use* of its process support features. Additionally, metrics regarding instance migration are provided. In turn, YAWL supports evolutionary changes in workflows based on Worklets [vdAtH05]. The latter refer to an extensive repertoire of self-contained sub-processes and association rules, which can be inserted into the process model without any system downtime. Even though it provides primitives for changing the process model, there are no change patterns or operations to realize changes at a higher abstraction level. Regarding the user, no guidance or change impact analysis are provided.

7 Summary and Outlook

Our overall vision is to enable schema evolution in object-aware processes. The major challenge lies on the very tight integration of the components of the framework. Such component dependencies might not only affect the object-aware process schema at static level, but active instances as well; i.e., new versions of a particular sub-schema must co-exist with unchanged versions of other sub-schemas. Moreover, we observed that user guidance is crucial to hide the complexity from the modeler and to avoid schema errors. In future work, we will provide detailed insights into our solution tackling the discussed requirements.

References

[CCPP98] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Workflow Evolution. *Data & Know Eng*, 24(3):211–238, 1998.

[GdSEM05] R. Galante, C. Saraiva dos Santos, N. Edelweiss, and A. F. Moreira. Temporal and Versioning Modeling for Schema Evolution in Object-oriented Databases. *Data & Know Eng*, 53(2):99–128, 2005.

[JH98] G. Joeris and O. Herzog. Managing Evolving Workflow Specifications. In *Proc.CoopIS'98*, pages 310–319, 1998.

[KG99] M. Kradolfer and A. Geppert. Dynamic Workflow Schema Evolution Based on Workflow Type Versioning and Workflow Migration. In *Proc. CoopIS'99*, pages 104–114, 1999.

[KR09a] V. Künzle and M. Reichert. Integrating Users in Object-aware Process Management Systems: Issues and Challenges. In *Proc. BPM'09 Workshops*, pages 29–41, 2009.

[KR09b] V. Künzle and M. Reichert. Towards Object-aware Process Management Systems: Issues, Challenges, Benefits. In *Proc. BPMDS'09*, pages 197–210, 2009.

[KR11] V. Künzle and M. Reichert. PHILharmonicFlows: Towards a Framework for Object-aware Process Management. *Journal of Software Mainteinance and Evolution: Research and Practice*, 23(4):205–244, 2011.

[Kün13] V. Künzle. *Object-aware Process Management*. PhD thesis, Ulm University, 2013.

[MRH07] D. Müller, M. Reichert, and J. Herbst. Data-driven Modeling and Coordination of Large Process Structure. In *Proc. CoopIS'07*, pages 131–149, 2007.

[MRH08] D. Müller, M. Reichert, and J. Herbst. A New Paradigm for the Enactment and Dynamic Adaptation of Data-driven Process Structures. In *Proc. CAiSE'08*, pages 48–63, 2008.

[MWR08] B. Mutschler, B. Weber, and M. Reichert. Workflow Management versus Case Handling: results from a Controlled Software Experiment. In *Proc. SAC'08*, pages 82–89, 2008.

[Ra04] Y.-G. Ra. Relational Schema Evolution for Program Independency. In *Proc. CIT 2004*, volume 3356, pages 273–281, 2004.

[RD98] M. Reichert and P. Dadam. ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. *Journal of Intelligent Information Systems*, 10(2):93–129, 1998.

[Rod96] J. F. Roddick. A Model for Schema Versioning in Temporal Database Systems. *Australian Computer Science Communications*, 18:446–452, 1996.

[RRD04] S. Rinderle, M. Reichert, and P. Dadam. Flexible Support of Team Processes by Adaptive Workflow Systems. *Distr. Parallel Databases*, 16(1):91–116, 2004.

[RRMD09] M. Reichert, S. Rinderle-Ma, and P. Dadam. Flexibility in Process-Aware Information Systems. *Trans Petri Nets and other Models of Conc III*, pages 115–135, 2009.

[RVV10] H. A. Reijers, J. Vogelaar, and I. Vanderfeesten. Changing Products, Changing Processes: Dealing with Small Updates in Product-Based Design. In *Proc. eKNOW'10*, pages 56–61, 2010.

[RW12] M. Reichert and B. Weber. *Enabling Flexibility in Process-aware Information Systems: Challenges, Methods, Technologies*. Springer, 2012.

[vdAtH05] W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.

[vdAWG05] W. M. P. van der Aalst, M. Weske, and D. Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data & Know Eng*, 53(2):129–162, 2005.

[WRRM08] B. Weber, M. Reichert, and S. Rinderle-Ma. Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. *Data & Know Eng*, 66(3):438–466, 2008.

[WW14] Y. Wang and Y. Wang. Change Analysis for Artifact-Centric Business Processes. In *Proc. BIS 2014*, pages 98–109, 2014.

[XSY$^+$11] W. Xu, J. Su, Z. Yan, J. Yang, and L. Zhang. An Artifact-centric Approach to Dynamic Modification of Workflow Execution. In *Proc. OTM 2011*, pages 256–273, 2011.

On the Usability of Business Process Modelling Tools – a Review and Future Research Directions

Maria Shitkova

University of Münster - ERCIS
Leonardo Campus 3
48149 Münster
maria.shitkova@ercis.uni-muenster.de

Abstract: The choice of a business process modelling (BPM) tool in combination with the selection of a modelling language is one of the crucial steps in BPM project preparation. Different aspects influence the decision: tool functionality, price, modelling language support, etc. In this paper we discuss the aspect of usability, which has already been recognized as an important topic in software engineering and web design. We conduct a literature review to find out the current state of research on the usability in the BPM field. The results of the literature review show, that although a number of research papers mention the importance of usability for BPM tools, real usability evaluation studies have rarely been undertaken. Based on the results of the literature analysis, the possible research directions in the field of usability of BPM tools are suggested.

1 Introduction

Business process modelling (BPM) has emerged as a popular and relevant practice in information systems (IS) [In09]. While academics tend to review existing and propose new approaches for process modelling and analysis, practitioners apply these approaches to real-world modelling projects. A successful BPM project requires proper management, in which preparatory activities play an important role [RSD11]. These activities start with the identification of relevant perspectives, communication channels and modelling technique, followed by selection of a BPM tool [RSD11].

The choice of a BPM tool is influenced by a number of factors, such as modelling methodology support, overall functionality or collaboration support [Ka07], [RSD11]. Numerous studies were conducted to compare BPM tools based on functionality aspects, such as collaboration support [RHI11], model analysis or process collections management [DRR12], [Ro11]. However, available research publication often neglect actual user experience and quality of user interaction with BPM tools.

Personal user experience during the interaction with a product is studied by performing usability evaluation [AT13]. Usability is an important aspect in such areas of IS as software engineering or web development. Nielsen states that nowadays usability is a necessary condition for survival in a constantly changing web environment [Ni94]. Usability studies are becoming more and more popular in the area of eLearning systems [OKU10], IS design for people with disabilities [Fu05] or clinical information systems [Am09].

In the area of BPM usability plays an equally important role [PCV10], [MR13]. Considering that the target user group of BPM tools is shifting from modelling experts to domain experts from the departments without specific IT knowledge [FT09], [LR12] the usability becomes even more crucial. BPM tools have to become less complex and more understandable [LR12], easy to learn and memorize [Re08], and efficient to use for an overall effective outcome. Usability is seen as the fore factor for technology choice and acceptance [VD00]. Moreover, considering usability goals and guidelines already during the tool design phase reduce the costs of tool support after the product has been released [BB00]. Despite these statements solid usability studies seem to be underrepresented in the BPM community. Therefore the goal of this paper is to evaluate the current state of research on the usability in the field of BPM and identify a set of future research opportunities for BPM tools usability.

The paper is structured as follows: in the related work section we provide the usability definition and brief overview of available evaluation methods, together with the definition of BPM tools. We then proceed with the description of the research method and present the main findings from the literature analysis. We conclude the paper with a discussion of the obtained results and definition of possible research directions.

2 Related Work

2.1 Usability

Usability is a concept widely used in IS, e.g. in evaluation of the interface design and quality of interactive systems [IF08], [Ni94]. Numerous definitions of usability can be found in the literature [AT13]. ISO 9241-11 defines usability as an extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [In98]. [Sh91] and IEEE glossary [In90] extend the above definition by including also speed, time and ease of learn, retention, errors, amount of support needed when working with a software and user specific attitude to the product. Therefore usability is a multi-dimensional concept, composed of the following attributes [Ni94], [In98], [Sh91], [In90]:

- effectiveness – accuracy and completeness of achieving specified goals;
- efficiency – resources expended in relation to effectiveness;
- learnability – ease of learning the functionality of the software;
- memorability – retention of the functionality while using the tool on irregular basis;
- satisfaction – subjective attitude of the user towards the product.

A set of metrics has been defined for usability measurement and evaluation, e.g. task success, number of errors, time to complete a task, user satisfaction. [AT13]. Usability metrics reveal the personal user experience during interaction with a product, characterised by the dimensions presented above [AT13]. At the same time usability does not measure general preferences or attitudes, it only evaluates actual experience when using a particular software product. Usability metrics provide information, helpful for improvement of the product functionality and market position, as well as for comparison of various products [AT13].

The metrics are collected during usability experiments, which might employ quantitative, qualitative or neuroIS methods. By using quantitative methods, the performance aspects of effectiveness, efficiency, learnability and memorability can be estimated. Qualitative methods are mostly employed for measuring satisfaction, a subjective variable, which cannot be revealed by performance evaluation. NeuroIS methods, such as eye-tracking, emotion measurement or analysis of brain activity, despite their unusualness are becoming more and more popular among IS researches as an additional perspective on user behaviour evaluation [BRL11].

2.2 Business Process Modelling Tools

BPM tool is a piece of software, which provides the user with a possibility to create, store, share and analyse business process models. Typical examples of BPM tools used in academia and practice are: Software AG's ARIS Business Process Analysis Platform[1], ADONIS [KK02], APROMORE [Ro11] and a web-based modelling tools, such as Signavio [KW10] or icebricks [Be13].

[Ro11] distinguish between standard and advanced BPM tools functionality. While the first group incorporates such features as creation, modification and deletion of process models, access control, and simple search queries [Ro11] and is provided in both academic and practice-oriented tools, the extended functionality, such as quality and correctness analysis, pattern-based analysis, or reporting functions are at the moment not widely presented in the commercial software, but have a great potential to become a part of it in the nearest future [Ro11].

3 Research Method

To address the research questions a systematic literature review [Br09], [WW02] was chosen as an appropriate research method. We put the focus on analysing and criticising research methods and outcomes regarding usability in the BPM field. We argue that in order to get the full picture of the current state of usability research in BPM, both studies on BPM languages and BPM tools have to be taken into account. It is important to review and compare the usability investigation approaches in both areas. The future research directions derived from the literature analysis are targeted only on BPM tools.

[1] http://www.softwareag.com/de/products/aris/bpa/overview/default.asp. Accessed 19.05.2014

The results of the review should be of value to the academics, who perform design science research in the area of BPM, as well as to the general IS audience interested in BPM and usability.

Based on the key terms defined in the previous chapter, we came up with a set of keywords for the literature search: “usability” combined with “BPM”, “process modelling”, “process management” and “conceptual modelling”. We searched for the keywords in the digital publication databases (ACM, AIS, DBLP, EbscoHost, ScienceDirect and SpringerLink) as well as Google Scholar (first 100 hits). The time frame of the publications was not restricted, so all the peer-reviewed conference papers and journal articles, published before November 2013, which satisfied the search criteria, were taken for the consideration.

As a result of the search we got a set 2402 publications, which was reduced to 386 papers by analysing titles and abstracts. After removing duplicates, ensuring if the terms “process” and “usability” were both present in the full text, as well as checking the possibility of downloading the paper, we have taken 74 papers for detailed analysis. From 74 papers 12 turned out to be not suitable for the review either because the usability was not used in appropriate context, or because BPM was not the main topic. The rest 62 paper were thoroughly analysed and included in the concept matrix.

4 Findings from the Literature Analysis

As a basis for the analysis we have constructed a concept matrix with the following dimensions: (a) year of publication, (b) domain - BPM languages and process models themselves (1) or BPM tools (2), (c) usage of usability term - presence of references to the established usability literature, and (d) research phase, e.g. motivation, requirements specification, evaluation or future steps.

4.1 Year of Publication

All the analysed papers and journal articles were published between the years 2005 and 2013. Therefore the topic of usability in BPM is relatively new. Moreover, the growth of interest to the topic of usability can be noticed starting the year 2010 (see Figure 1).

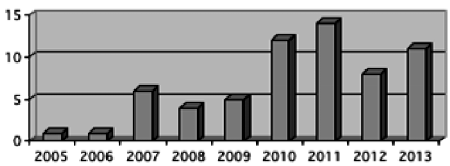

Figure 1. Publications’ distribution by year

4.2 Publication Domains

We have identified the publications on different aspects of process modelling, such as development of new and analysis of existing BPM languages, evaluating quality of BPM models, as well as development of new functions for BPM tools. We then classified all the papers into two main groups: BPM languages and models (1) and BPM tools (2). Some of the papers covered both aspects and thus are included in both groups.

The first group includes 29 papers, which mostly propose new approaches to BPM [BSM11], [Bu10], [Fe10], [GK10], [Hu08], [Li12], [LR12], [SGN07], [Sc11]. Furthermore publications regarding evaluation [GHA11], [Ro09a], [Ro09b], [SG10] and comparison [BKO10], [BO10], [Fe10], [Fi10], [Ka07] [LS07] of already existing modelling languages (e.g. BPMN, UML Activity Diagram) were identified, together with extensions of the existing modelling notations (BPMN) [BFV11], [Ku11], [MB12], [Na11], [Pa11], [WS07], [WSM07]. Moreover, [GD05] proposed a framework for measuring business process quality, in which usability was one of the characteristics, [MRR10] analysed different activity labelling approaches and [FT09], [SA13] proposed a new business process visual query language.

The rest 38 papers either presented new, or analysed existing BPM tools. Most of these publications present prototypes or completely implemented tools for process design [An13], [Be13], [Bu10], [DV11], [Fe10], [JCS07], [KH09], [LWP08], [Mo09], [Re08], [RKG13], [Sc11], [WR06], [WBR10]. However, [Co13], [FT09], [KKR11], [KRR12], [KHO11], [Ku08], [LMR11] proposed functional extensions for existing BPM tools – a piece of some specific functionality, such as model querying or change tracking, recommendation based modelling or automatic labelling of process activities. Two papers were devoted to the architecture of process model repositories [GK10], [WW10] and four specifically investigated the collaboration aspects of BPM tools [AN11], [SCS13], [WO12]. Moreover, requirements for BPM tools were summarised based on empirical research [ASI10], [PCV10] or comparison of existing tools or repositories [EJ12], [MR13], [RSS13].

[KL11] conducted a solid usability evaluation of software process modelling tools. Although the paper does not fully suite the analysed domains, we still included it in the review as a good example of a thorough usability study.

4.3 Usage of the Usability Term

As the next point of analysis we looked if the papers provided definition of usability and used the references to the standard usability literature [In90], [In98], [Ni94], [Sh91]. Only 3 papers used the definitions from these sources [BKO10], [BO10], [Fi10]. Besides this, [Ro09a], [Ro09b], [SG10] used the definition from ISO/IEC 9126 standard[2] on software engineering product quality, where usability is a characteristic of external quality and is actually seen as understandability.

[2] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749 Accessed 20.05.2014

Approximately the same set of publications discussed the metrics, used for usability evaluation [BKO10], [BO10], [GD05], [Ro09a], [Ro09b], [SG10]. In [SG10] the usability (understandability was used as the actual term in the paper) was measured as time, number of correct answers and efficiency, defined as the number of correct answers in relation to time. [Ro09a], [Ro09b] used answer time, success rate, efficiency, subjective evaluation as measures of usability (understandability). [GD05] employed an extended subset of metrics from ISO 9126 for evaluating the quality of process models: understandability (description completeness), operability (cancellability, undoability and monitorability) and attractiveness of interaction. [BKO10], [BO10] used efficiency, effectiveness, and satisfaction measures. "Number of commands used" metric and a standard usability questionnaire for the evaluation were employed in [DV11].

4.4 Usability as a Main Topic of the Study

Only in two publications [BKO10], [BO10] a systematic usability evaluation was presented. Both works compared the usability of BPMN and UML activity diagrams. As already stated in the previous chapter the papers have references to the standard usability literature and use the metrics of efficiency, effectiveness and satisfaction to measure the usability of two modelling languages. Other papers included the concept of usability only in particular steps of their research, as discussed in the next section.

4.5 Research Phase

We have identified six main areas, in which usability was mentioned in analysed publications: comparison of BPM languages or tools; requirements specification; evaluation; ensuring usability before conducting an experiment; and future steps. Figure 2 presents the distribution of papers depending on the research phase, in which the usability was mentioned. The most popular phases were requirements specification, evaluation and future steps. However, the works on BPM languages mention the usability more on in the requirements, and the papers on BPM tools – in future steps.

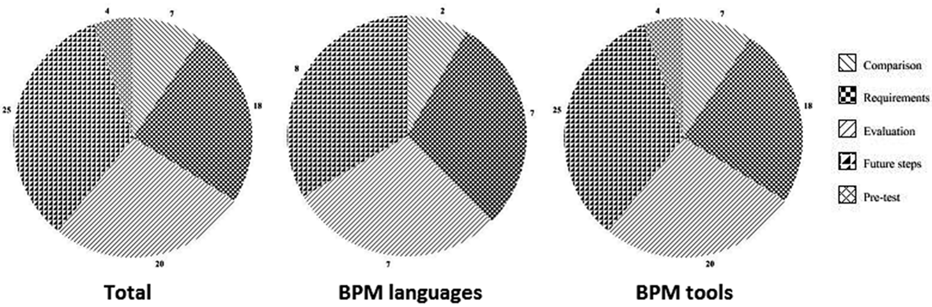

Figure 2. Relevant papers' distribution by research phase

Comparison

The more notations are developed for BPM, the more research publications on comparison of existing ones are made. Each of these publications tend to choose the most appropriate notation for a certain purpose. Thus, [BKO10], [BO10] analysed and compared UML Activity Diagrams and BPMN. One of the criteria for comparison was usability of the notations, measured according to [In98] as effectiveness, efficiency and satisfaction. [Fi10] used the usability definition, provided by [Ni94], and stated that usability is important in two cases: model creation and model interpretation. They further discussed representation of routing elements in BPM languages from the usability perspective. [LS07] compared graphical BPM approaches to the rule-based specifications of business processes, where they understood usability as simplicity of use. And finally [Ka07] proposed a selection framework of BPM method, in which one of the criteria was usability, understood as functional capability to express certain process behaviour.

The comparison of BPM tools is performed quite seldom, which is explained by the fact that research in this area is targeted on creation of new BPM tool functionality. [KL11] compared two software process modelling tools, using ISO metrics and [EJ12] compared existing BPM repositories to identify challenges that affect their usability in supporting re-use of process models [EJ12].

Requirements specification

Usability is seen as important characteristic of BPM languages and tools, and thus should be taken into account when proposing new solutions [KH09], [Sc11], [SA13]. The same statement is supported by empirical research [ASI10], [PCV10], [RSS13], [MR13]. Usability requirements arise mostly due to shifting of the target user group from IS engineers to business users [BSM11], [FT09], [Ku08], [LR12], [Mo09].

Despite the common belief that increasing usability usually means decrease in flexibility of the modelling tools [DV11], [Mo09], [Re08] or expressiveness of modelling approaches [BSM11], the authors still try to find a trade-off [BSM11], [DV11], [Mo09], [Re08]. Common usability requirements demand less complex and more understandable BPM tools and languages [DV11], [Ku11], [LR12], [Mo09], [Na11]; better graphical tool support and user-friendly visual representation of models [FT09], [Ku11], [Ku08], [Mo09], [Ku11]; intuitive interface and improved functionality together with shorten learning period for BPM tools [DV11], [WSM07], [Re08].

Evaluation

From the total 20 publications, which mention usability in the evaluation section, 7 were devoted to the evaluation of BPM notations or process models [BFV11], [Fe10], [GHA11], [Li12], [MB12], [Sc11], [SA13]. The authors used different evaluation approaches, such as conducting user survey [Sc11]; experiment, where the participants were asked to interpret process models [MB12], use two different process model querying techniques [SA13], and comparing how correctly users were solving modelling tasks [GHA11]. [Li12] performed the evaluation with use-cases, when different possible modelling scenarios were implemented using the provided BPM language. [BFV11] evaluated the proposed notation by implementing it in a modelling environment and showing that the proposed constructs and rules are realizable. One of the interesting evaluation approaches was used by [Fe10], where the BPM notation was compared to user interface and its elements to the icons in a software program. Thus, it was possible to use a method of [Ni95], which was originally developed for evaluation of iconographic usability.

The other 15 papers were devoted to the usability of BPM tools. The most popular evaluation approach was conducting an experiment and surveying the participants on their personal experience with the tool [AN11], [Co13], [DV11], [Do13], [KHO11], [Mo09], [SCS13], [WW10], [WR06]. In one of the papers a special attention was put to the fact that "not computer scientists" participated in the interviews. In comparison to subjective evaluation, performance metrics were gathered only in two cases [KHO11], [SCS13], and in one case the clickstream data was analysed [WW10]. Participant observation was used in [An13], [DV11], [Do13], [WO12]. [WW10] and [Be13] based their evaluation on deployment of the tool in different settings and gathering the user opinion after some period of usage. [LWP08] made the "theoretical evaluation" by discussing the functionality of the tool, which were supposed to increase its usability.

Ensuring usability before conducting an experiment

[Cl12], [Pi13], [Pi12a], [Pi12b] ensured the usability of the tool by conducting a pre-test before carrying out an experiment, which was not connected to the usability evaluation. Unfortunately the authors did not provide any details on the pre-test.

Future steps

A great part of the analysed papers mention usability among the future steps. Most of the papers state that they plan to conduct a usability evaluation of the proposed modelling approach or developed modelling tool in empirical study [MRR10], [Pa11] [SGN07], real world usage scenario [GK10], [RKG13], [WBR10] or experiment [FT09], [KRR12], [Sc11]. Others claim they plan to improve the usability by applying user-centric design approach based on prototype iteration [WS07], focusing on usability aspects [KKR11], applying design principles in support of usability and flexibility [Hu08], incorporating usability features based on conducted evaluation [AN11], [An13], [Be13], [LWP08], [Sc11], [WO12], following usability inspection methods [Fe10] or by just analysing and addressing usability and performance of the system while extension [JCS07]. Two publications compared the usability of the proposed approach to the existing ones [LMR11] using Delphi method [Bu10]. And only one author has explicitly specified that usability metrics will be discussed in future publications [RKG13].

Combination of several steps

In some cases usability was mentioned not only in a single research phase, but in a combination of two. [DV11], [Mo09], [SA13] state that usability is an imporatant requirement for a BPM artifact and thus has to be taken into account when developing new solutions. They then perfomed an evaluation of the proposed visual quering language [SA13] and BPM tools [DV11], [Mo09]. Similary [FT09], [Sc11] indentify usability as an important requirement for BPM tool, but as the implementation of the artefact has not been yet finished by the moment of the paper publication the actual evaluation has been postponed to the future steps. Finally usability was mentioned in both *evaluation* and *future steps* sections. [Sc11] proposed a new BPM approach and [AN11], [An13], [Be13], [LWP08], [WO12] presented new BPM tools. All the publications stated, that the usability of the proposed artefact was evaluated and revealed possibilities for improvement, which were postponed to the future steps.

5 Discussion and Future Research Directions

Usability has established itself as an important topic in such areas of IS as web development and software engineering. The goal of this research paper was to evaluate the current state of research on usability in the area of BPM and identify a set of future research opportunities for the topic of BPM tools usability. For this purpose a systematic literature review was conducted, in which we have analysed papers and journal articles, which have either BPM languages or BPM tools as a main topic and at the same time mention usability as one of the aspects of their research. In this section we summarize the main findings from the literature analysis and synthesis and suggest possible research directions on the usability of BPM tools.

Foremost it should be said that usability is a relatively new aspect of research in the area of BPM and its popularity has been growing in the last five years. The empirical research underlines the importance of usability, which can be primarily explained by the shift of target user group of BPM tools from IS engineers to the business users without particular IT knowledge. Therefore it is definitely necessary to continue the research in this area.

However, the review also revealed quite unimpressive results concerning the usage of the term itself – usability is still often used just as a “buzzword”, without providing any concrete definition. Only few publications are referring to the established literature on usability. Taking into account that usability in web and interface design has been investigated already for 25 years, the academic works on BPM should benefit from the available research results by applying already existing usability practices.

The number of thorough usability studies in BPM is still quite small. Mostly these solid studies are conducted for BPM languages, but not for BPM tools. Usability has been successfully employed for comparing existing BPM notations (e.g. BPMN and UML Activity Diagrams), however comparison of BPM tools is rarely conducted. We suggest that more attention should be paid to comparison, evaluation and improvement of existing BPM tools, rather than implementation of similar functionality in new artefacts.

BPM publications without primarily focus on usability still mention it in separate research phases, such as requirements specification, evaluation and future steps. Our analysis shows that there is still a room for improvement at each of these phases. The usability requirements specified for BPM tools are still relatively vague and unstructured. We however argue, that based on the similarity of the available BPM tools’ functionality, the requirements could be summarized in a set of usability guidelines applicable to a generic BPM tool. Furthermore evaluation of the artefacts is nowadays mostly performed with surveys. Although qualitative techniques are valuable for measuring user satisfaction, performance metrics, which require quantitative analysis, should not be disrespected. One of the reasons for this deficiency is the absence of usability metrics tailored for BPM tools. Therefore development of a set of such metrics can be a potential research direction. The future steps part of the paper is always "a fertile ground for buzzwording", however even here more attention can be paid to the aspect of usability by providing details on the possible evaluation approach or methods for ensuring and improving BPM tool usability. Finally, the researchers should not neglect employing the usability methods at several research phases to increase the quality of the developed BPM artefact. BPM tools will only benefit from the usability research and thus improve the acceptance of BPM in both academic and practical environments.

As a limitation of the current research we should mention that the suggested future research directions are just a first research step towards the construction of the final research agenda. In the future research we plan to evaluate the suggested research directions with BPM experts from academia and practice, prioritize them according to their importance and finally compose a research agenda for the topic of BPM tools usability.

References

[ASI10] Abdullah, N.; Sadiq, S.; Indulska, M.: Emerging Challenges in Information Systems Research for Regulatory Compliance Management. In: Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2010; pp.251-265.

[AT13] Albert, W.; Tullis T.: Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics, Newnes, 2013.

[Am09] Amarasingham, R.; Pronovost, P. J.; Diener-West, M.; Goeschel, C.; Dorman, T.; Thiemann, D.R.; Powe, N.R.: Measuring Clinical Information Technology in the ICU Setting: Application in a Quality Improvement Collaborative. In: Journal of the American Medical Informatics Association : JAMIA (14:3), 2009; pp. 288–94.

[AN11] Amiyo, M.; Nabukenya, J.: A Collaboration Support Environment for Decision Enhancement in Business Process Improvement. In: Collaboration and Technology, Springer Berlin Heidelberg, 2011; pp.175-190.

[An13] Antunes, P.; Simões, D.; Carriço, L.; Pino, J.A.: An End-User Approach to Business Process Modeling. In: Journal of Network and Computer Applications (36:6) 2013. Elsevier; pp. 1466–1479.

[BB00] Bevan, N.; Bogomolni, I.: Incorporating User Quality Requirements in the Software Development Process. In: Proceedings of the 4th International software Quality Week Europe, 2000; pp.1192-1204

[Be13] Becker, J.; Clever, N.; Holler, J.; Püster, J.; Shitkova, M.: Integrating Process Modeling Methodology, Language and Tool – A Design Science Approach. In: The Practice of Enterprise Modeling, 2013. Springer Berlin Heidelberg; pp. 221–235.

[BKO10] Birkmeier, D.; Klöckner, S.; and Overhage, S.: An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users. In: 18th European Conference on Information Systems, 2010; pp. 1–12

[BO10] Birkmeier, D.; Overhage, S.: Is BPMN Really First Choice in Joint Architecture Development? An Empirical Study on the Usability of BPMN and UML Activity Diagrams for Business Users. In: Research into Practice–Reality and Gaps. Springer Berlin Heidelberg, 2010; pp. 119–134.

[BSM11] Borgert, S.; Steinmetz, J.; Mühlhäuser, M.: ePASS-IoS 1.1: Enabling Inter-Enterprise Business Process Modeling by S-BPM and the Internet of Services Concept. In: S-BPM ONE-Learning by Doing-Doing by Learning. Springer Berlin Heidelberg, 2011; pp. 190-211.

[BFV11] Brambilla, M.; Fraternali, P.; Vaca, C.: A Notation for Supporting Social Business Process Modeling. In: Business Process Model and Notation. Springer Berlin Heidelberg, 2011; pp. 88–102.

[BRL11] Vom Brocke, J.; Riedl, R.; Léger, P.-M.: Neuroscience in Design-Oriented Research: Exploring New Potentials. In: Proceedings of the 6th International Conference on Design Science Research in Information Systems and Technology. USA, 2011; pp. 427–439.

[Br09] Vom Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Plattfaut, R.; Cleven, A.: Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: ECIS 2009 Proceedings, 2009; pp. 2206–2217.

[Bu10] Businska, L.: Multidimensional Business Process Modeling Approach. In: Advances in Databases and Information Systems. Springer Berlin Heidelberg, 2010; pp. 247–256.

[Cl12] Claes, J.; Vanderfeesten, I.; Reijers, H.A.; Pinggera, J.; Weidlich, M.; Zugal, S.; Fahland, D.; Weber, B.; Mendling, J.; Poels, G.: Tying Process Model Quality to the Modeling Process: the Impact of Structuring, Movement, and Speed. In: Business Process Management. Springer Berlin Heidelberg, 2012; pp. 33–48.

[Co13] Conforti, R.; La Rosa, M.; Fortino, G.; ter Hofstede, A.H.M.; Recker, J.; Adams, M.: Real-Time Risk Monitoring in Business Processes: A Sensor-Based Approach. In: Journal of Systems and Software 86(11). Elsevier, 2013; pp. 2939–2965.

[DV11] Dengler, F.; Vrandečić, D.: Wiki-Based Maturing of Process Descriptions. In: Business Process Management. Springer Berlin Heidelberg, 2011; pp. 313–328.

[DRR12] Dijkman, R.; La Rosa, M.; Reijers, H.A.: Managing Large Collections of Business Process Models - Current Techniques and Challenges. In: Computers in Industry 63(2), 2012; pp. 91–97.

[Do13] Döweling, S.; Tahiri, T.; Schmidt, B.; Nolte, A.; Khalilbeigi, M.: Collaborative Business Process Modeling on Interactive Tabletops. In: ECIS 2013 Proceedings, 2013.

[EJ12] Elias, M.; Johannesson, P.: A Survey of Process Model Reuse Repositories. In: Information Systems, Technology and Management. Springer, 2012; pp. 64–76.

[Fe10] Fernández, H.F.; Palacios-González, E.; García-Díaz, V.; Pelayo G.; Bustelo, B.C.; Sanjuán Martínez, O.; Cueva Lovelle, J.M.: SBPMN — an Easier Business Process Modeling Notation for Business Users. In: Computer Standards & Interfaces 32(1). Elsevier, 2010; pp. 18–28.

[Fi10] Figl, K.; Mendling, J.; Strembeck, M.; Recker, J.: On the Cognitive Effectiveness of Routing Symbols in Process Modeling Languages. In: Business Information Systems. Springer Berlin Heidelberg, 2010; pp. 230–241.

[FT09] Di Francescomarino, C.; Tonella, P.: Crosscutting Concern Documentation by Visual Query of Business Processes. In: Business Process Management Workshops. Springer Berlin Heidelberg, 2009; pp. 18–31.

[Fu05] Fukuda, K.; Saito, S.; Takagi, H.; Asakawa, C.: Proposing New Metrics to Evaluate Web Usability for the Blind. In: CHI ’05 extended abstracts on Human factors in computing systems. ACM, 2005; p. 1387-1390.

[GK10] Gao, S.; Krogstie, J.: A Repository Architecture for Business Process Characterizing Models. In: The Practice of Enterprise Modeling. Springer, 2010; pp. 162–176.

[GHA11] Genon, N.; Heymans, P.; Amyot, D.: Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In: Software Language Engineering, 2011; pp. 377–396.

[GD05] Guceglioglu, A.S.; Demirors, O.: Using Software Quality Characteristics to Measure Business Process Quality. In: Business Process Management. 2005; pp. 374–379.
[Hu08] Hull, R.: Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges. In: On the Move to Meaningful Internet Systems: OTM 2008. Springer Berlin Heidelberg, 2008; pp. 1152–1163.
[In09] Indulska, M.; Recker, J.; Rosemann, M.; Green, P.: Business Process Modeling: Current Issues and Future Challenges. In: Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2009; pp. 501-514.
[IF08] Insfran, E.; Fernandez, A.: A Systematic Review of Usability Evaluation in Web Development. In: Web Information Systems Engineering–WISE 2008 Workshops. Springer Berlin Heidelberg, 2008; pp. 81-91.
[In90] Institute for Electrical and Electronics Engineers (IEEE): Standard Glossary of Software Engineering Terminology. 1990
[In98] International Organization for Standardization (ISO): International Standard Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)-Part 11: Guidance on Usability. Geneve, Switzerland; 1998.
[JCS07] Jung, J.; Choi, I.; Song, M.: An Integration Architecture for Knowledge Management Systems and Business Process Management Systems. In: Computers in Industry 58(1), 2007; pp. 21–34.
[KKR11] Kabicher, S.; Kriglstein, S.; Rinderle-Ma, S.: Visual Change Tracking for Business Process Models. In: Conceptual Modeling–ER 2011 Springer, 2011; pp. 504-513.
[KK02] Karagiannis, D.; Kühn, H.: Metamodelling Platforms. In: Proceedings of the 3rd International Conference EC-Web 2002, 2002; pp. 182–196.
[Ka07] Kaschek, R.; Pavlov, R.; Shekhovtsov, V.; Zlatkin, S.: Characterization and Tool Supported Selection of Business Process Modeling Methodologies. In: Technologies for Business Information Systems. Springer Netherlands, 2007; pp. 25-37.
[KL11] Kerzazi, N.; Lavallee, M.: Inquiry on Usability of Two Software Process Modeling Systems Using ISO/IEC 9241. In: Electrical and Computer Engineering (CCECE), 2011. IEEE, 2011. pp. 773-776.
[KRR12] Kolb, J.; Rudner, B.; Reichert, M.: Towards Gesture-Based Process Modeling on Multi-Touch Devices. In: Advanced Information Systems Engineering Workshops. Springer Berlin Heidelberg, 2012; pp. 280-293.
[KHO11] Koschmider, A.; Hornung, T.; Oberweis, A.: Recommendation-Based Editor for Business Process Modeling. In: Data & Knowledge Engineering 70(6), 2011; pp. 483-503.
[Ku11] Kunz, S.; Fickinger, T.; Prescher, J.; Spengler, K.: Managing Complex Event Processes with Business Process Modeling Notation. In: Business Process Modeling Notation. Springer Berlin Heidelberg, 2011; pp. 78-90.
[KW10] Kunze, M.; Weske, M.: Signavio-Oryx Academic Initiative. In: BPM 2010 Demonstration Track, 2010; pp. 6.
[Ku08] Küster, J.M.; Gerth, C.; Förster, A.; Engels, G.: Detecting and Resolving Process Model Differences in the Absence of a Change Log. In: Business Process Management. Springer Berlin Heidelberg, 2008; pp. 244-260.
[KH09] Küster, T.; Heßler, A.: Towards Transformations from BPMN to Heterogeneous Systems. In: Business Process Management Workshops. Springer, 2009; pp. 200-211.
[LWP08] Lee, J.; Wyner, G.M.; Pentland, B.T.: Process Grammar as a Tool for Business Process Design. In: MIS Quarterly 32(4), 2008; pp. 757–778.
[LMR11] Leopold, H.; Mendling, J.; and Reijers, H.A.: On the Automatic Labeling of Process Models. In: Advanced Information Systems Engineering. Springer, 2011; pp. 512-520.
[Li12] Liptchinsky, V.; Khazankin, R.; Truong, H.-L.; Dustdar, S.: A Novel Approach to Modeling Context-Aware and Social Collaboration Processes. In: Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2012; pp. 565–580.

[LR12] Lohrmann, M.; Reichert, M.: Modeling Business Objectives for Business Process Management. In: S-BPM ONE–Scientific Research. Springer, 2012; pp. 106-126.

[LS07] Lu, R.; Sadiq, S.: A Survey of Comparative Business Process Modeling Approaches. In: Business Information Systems. Springer Berlin Heidelberg, 2007; pp. 82-94.

[MB12] Malik, S.; Bajwa, I.S.: A Rule Based Approach for Business Rule Generation From Business Process Models. In: Rules on the Web: Research and Applications. Springer Berlin Heidelberg, 2012; pp. 92-99.

[MRR10] Mendling, J.; Reijers, H.A.; Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. In: Information Systems 35(4), 2010; pp. 467–482.

[Mo09] Montani, S.: Prototype-Based Management of Business Process Exception Cases. In: Applied Intelligence 33(3), 2009; pp. 278–290.

[MR13] Mutschler, B.; Reichert, M.: Understanding the Costs of Business Process Management Technology. In: Business Process Management. Springer, 2013; pp. 157-194.

[Na11] Natschläger, C.: Deontic BPMN. In: Database and Expert Systems Applications. Springer Berlin Heidelberg, 2011; pp. 264-278.

[Ni94] Nielsen, J.: Usability engineering, Elsevier, 1994.

[Ni95] Nielsen, J.: Icon Usability for the 1995 Sun Microsystems' Website, 1995. http://www.nngroup.com/articles/icon-usability-1995-sun-microsystems-website/ . Accessed 19.05.2014

[OKU10] Oztekin, A.; Kong, Z.J.; Uysal, O.: UseLearn: A Novel Checklist and Usability Evaluation Method for eLearning Systems by Criticality Metric Analysis. In: International Journal of Industrial Ergonomics 40(4), 2010; pp. 455–469.

[PCV10] Patig, S.,;Casanova-Brito, V.; Vögeli, B.: IT Requirements of Business Process Management in Practice–an Empirical Study. In: Business Process Management. Springer Berlin Heidelberg, 2010; pp. 13-28.

[Pa11] Pautasso, C.: BPMN for REST. In: Business Process Model and Notation. Springer Berlin Heidelberg, 2011; pp. 74-87.

[Pi13] Pinggera, J.; Furtner, M.; Martini, M.; Sachse, P.; Reiter, K.; Zugal, S.; Weber, B.: Investigating the Process of Process Modeling with Eye Movement Analysis. In: Business Process Management Workshops. Springer, 2013; pp. 438-450.

[Pi12a] Pinggera, J.; Soffer, P.; Zugal, S.; Weber, B.; Weidlich, M.; Fahland, D.; Reijers, H.A.; Mendling, J.: Modeling Styles in Business Process Modeling. In: Business-Process and Information Systems Modeling. Springer Berlin Heidelberg, 2012; pp. 151-166.

[Pi12b] Pinggera, J.; Zugal, S.; Weidlich, M.; Fahland, D.; Weber, B.; Mendling, J.; Reijers, H. A.: Tracing the Process of Process Modeling with Modeling Phase Diagrams. In: Business Process Management Workshops. Springer, 2012; pp. 370-382.

[RSS13] Reijers, H.A.; Slaats, T.; Stahl, C.: Declarative Modeling–An Academic Dream or the Future for BPM?. In: Business Process Management. Springer, 2013; pp. 307-322.

[Re08] Ren, C.; Wang, W.; Dong, J.; Ding, H.; Shao, B.; Wang, Q.: Towards a Flexible Business Process Modeling and Simulation Environment. In: Simulation Conference, 2008. IEEE, 2008; pp. 1694-1701.

[RHI11] Riemer, K.; Holler, J.; Indulska, M.: Collaborative Process Modelling - Tool Analysis and Design Implications. In: ECIS 2011: European Conference on Information Systems-ICT and Sustainable Service Development. Association for Information Systems, 2011.

[Ro09a] Rolón, E.; Cardoso, J.; García, F.; Ruiz, F.; Piattini, M.: Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models. In: Enterprise, Business-Process and Information Systems Modeling. Springer, 2009; pp. 58-70.

[Ro09b] Rolon, E.; Sanchez, L.; Garcia, F.; Ruiz, F.; Piattini, M.; Caivano, D.; Visaggio, G.: Prediction Models for BPMN Usability and Maintainability. In: Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on. IEEE, 2009; pp. 383-390.

[Ro11] La Rosa, M.; Reijers, H.A.; van der Aalst, W.M.P.; Dijkman, R.M.; Mendling, J.; Dumas, M.; García-Bañuelos, L.: APROMORE: An Advanced Process Model Repository. In: Expert Systems with Applications 38(6), 2011; pp. 7029–7040.

[RSD11] Rosemann, M.; Schwegmann, A.; Delfmann, P.: Preparation of Process Modeling. In: Process Management: A Guide for the Design of Business Processes. J. Becker, M. Kugeler, and M. Rosemann (eds.), Springer Berlin / Heidelberg, 2011; pp. 41–90.

[RKG13] Rychkova, I.; Kirsch-Pinheiro, M.; Le Grand, B.: Context-Aware Agile Business Process Engine: Foundations and Architecture. In: Enterprise, Business-Process and Information Systems Modeling. Springer Berlin Heidelberg, 2013; pp. 32-47.

[SGN07] Sadiq, S.; Governatori, G.; Namiri, K.: Modeling Control Objectives for Business Process Compliance. In: Business process management. Springer, 2007; pp. 149-164.

[SG10] Sánchez-González, L.; García, F.: Quality Assessment of Business Process Models Based on Thresholds. In: On the Move to Meaningful Internet Systems: OTM 2010. Springer Berlin Heidelberg, 2010; pp. 78-95.

[Sc11] Schnabel, F.; Gorronogoitia, Y.; Radzimski, M.; Lécué, F.; Mehandjiev, N.; Ripa, G.; Abels, S.: Empowering Business Users to Model and Execute Business Processes. In: Business Process Management Workshops. Springer, 2011; pp. 433-448.

[SCS13] Scholtz, B.; Calitz, A.; Snyman, I.: The Usability of Collaborative Tools: Application to Business Process Modelling. In: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference. ACM, 2013; pp. 347-358.

[Sh91] Shackel, B.: Usability - Context, Framework, Definition, Design and Evaluation. In: Human factors for informatics usability, 1991; pp. 21-37.

[SA13] Störrle, H.; Acretoaie, V.: Querying Business Process Models with VMQL. In: Proceedings of the 5th ACM SIGCHI Annual International Workshop on Behaviour Modelling-Foundations and Applications. ACM, 2013; pp. 4.

[VD00] Venkatesh, V.; Davis, F.D.: A Theoretical Acceptance Extension Model: Field Four Studies of the Technology Longitudinal. In: Management Science (46:2), 2000, pp. 186–204.

[WO12] Wachholder, D.; Oppl, S.: Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes. In: S-BPM ONE–Scientific Research. Springer Berlin Heidelberg, 2012; pp. 145-162.

[WW02] Webster, J.; Watson, R.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. In: MIS Quaterly (26:2), 2002; pp. XIII–XXIII.

[WW10] Wang, H.J.; Wu, H.: Supporting Process Design for E-Business via an Integrated Process Repository. In: Information Technology and Management 12(2), 2010; pp. 97–109.

[WR06] Wang, J.; Rosca, D.: Dynamic Workflow Modeling and Verification. In: Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2006; pp. 303-318.

[WBR10] West, S.; Brown, R.A.; Recker, J.: Collaborative Business Process Modeling Using 3D Virtual Environments. In: Proceedings of the 16th Americas Conference on Information Systems: Sustainable IT Collaboration around the Globe. Association for Information Systems (AIS), 2010; pp.12-15.

[WS07] Wolter, C.; Schaad, A.: Modeling of Task-Based Authorization Constraints in BPMN. In: Business Process Management. Springer Berlin Heidelberg, 2007; pp. 64-79.

[WSM07] Wolter, C.; Schaad, A.; Meinel, C.: Deriving XACML Policies from Business Process Models. In: Web Information Systems Engineering–WISE 2007 Workshops. Springer Berlin Heidelberg, 2007; pp. 142-153.

Process Implementation

Visual Analytics for Supporting Manufacturers and Distributors in Online Sales

Olivier Parisot [1], Gero Vierke [2], Thomas Tamisier [1], Yoanne Didry [1], Helmut Rieder [2]

[1] Public Research Centre - Gabriel Lippmann
41, rue du Brill, L-4422 Belvaux
[2] InfinAIt Solutions SA
2a, Ennert dem Bierg L-5244 Sandweiler

Abstract: This article presents the basic concepts of OPTOSA, a Visual Analytics solution for the optimization of online sales, designed to support manufacturers in all phases of the online sales process from the product specification to the price fixing and more. OPTOSA combines a data processing module that builds and constantly updates operational knowledge related to sales positioning with a decision assistant that uses relevant aspects of the knowledge for helping the tasks of the different teams along the integrated chain of the sales. After reviewing the challenges of the approach, we discuss the first significant experiments with OPTOSA on some formalized use-cases.

1 Introduction

Everybody realize that the future of selling lies on the Internet. Therefore, most manufacturers and uncountable merchants devote themselves in a hard and continuous competition over the online marketplaces. However, many of them do not develop any structured approach to increase their market shares. Others use rather naive procedures, commonly based on lowering prices and thus reducing sales margins. An intelligent approach, based on timely usage of dynamic information directly available from the Internet will give us unique and decisive advantage.

Generally speaking, one of the greatest challenges for the manufacturers is that in online sales the distributors in the online market do not fulfill the same function as in the stationary market. In particular, in the stationary market the distributor, especially the procurement manager, supports the manufacturer with his know how on the consumer needs, namely the optimal specification and presentation of products. In contrast, in the online distribution, the manufacturer is given full freedom to manage all aspects of the sale process.

There is therefore a strong need to provide the sales business with an alternate solution that could make up for the human competence that is not at hand any more. This is the purpose of the OPTOSA system, developed in a partnership with researchers and practitioners, and specified after a thorough operational state of the art driven by three progressive investigation steps:

- In what way it would be possible to rely on data nowadays at disposal with a view to maximizing turnover and margin?
- Based on this knowledge from the data, which processes in online sales could be automated?
- Last, which decisions within the other processes could be supported by a knowledge based assistant?

With regard to the data, there has been a lot of research done and some tools are available in the domain of Search Engine Optimization (SEO), in particular [Sis, Wis], or tools from the search engines themselves like Google Analytics. However, those tools focus on search engines like Google and not on online market platforms. The ranking mechanism of search engines and market places are similar but not identical, the essential difference being that search engines aim to deliver most relevant content while market places aim to deliver products that are most likely to be bought. The goal of OPTOSA goes beyond this current state of the art, in order to figure out those differences and to develop algorithms that for instance analyze the meta-content of successful products and to extract the most relevant key-words and to compute the optimal structure for our meta-content.

Complementary, a lot of data sources like [Adv] propose business information that could be formalized to automatic handle or optimize several tasks in the sale process. Made of static knowledge about common best practices, this information can be outdated when applied to the fast evolving domain of online sales. In contrast, the OPTOSA system is based on the monitoring of live data in view of delivering to the sales collaborators an accurate business handling and a pro-active decision support.

In this paper, we present a software architecture that aims to optimize online sales. This architecture integrates several building blocks for data management, data mining, visualization, knowledge extraction and endly prediction.

The rest of this article is organized as follows. Firstly, challenges are presented. Secondly, the management of data is described in details. Finally, the architecture of the software is presented, and the results of experiments are discussed.

2 Challenges

Within the preliminary state of the art, we identified two main challenges towards a system that could cure the lack of expert knowledge.

The first one is to extract business information from the available data. A lot of sales data are available from online marketplaces, either directly to the general public or to registered users. As such, the raw data cannot be straightforwardly used to solve any concrete problem such as optimizing the sales through a given platform. We need to understand the data in their specific context of use, and then to mine them according to the knowledge we intend to build: remove outliers, find correlations, etc. Moreover, user control and interaction should be possible during the processing of so diversified bunches of data: we follow here the *Visual Analytics* paradigm, a way of addressing such complexity through an efficient combining of the respective powers of the computer and the human brain [HA08].

The second challenge is to process this business information within concrete usecases by means of a collaborative expert system and a transparent data model. The OPTOSA system primarily intends to generate recommendations for operating on an online platform, dedicated to the sales collaborators. The recommendations must be in any case supported by a clear and suitable justification. The justification exhibits both the trace of the reasoning and available references to knowledge extracted from the data. The traceability brings constraints on the architecture and the technology used to implement the system. In particular, OPTOSA will use a referencing mechanism associates rules with documentary sources. Also, when updating the knowledge base, the references allow identifying the impact of a given knowledge source on the rules model, in order to replace them.

Sales processes are modeled through different procedures and algorithms. We structured the functionalities of the OPTOSA system according to separate optimization needs that could be refined independently:

- Direct optimization of the sale of a product.
- Optimization in relation to sales and delivery channels: How to deal with competitors?
- Reporting and analysis: How to improve the offer through the consolidation of the sale channels?

As mentioned above some of the functionalities can be fully automated, given the values of parameters asked by the system. An example is the dynamic price adjustment: for each product to be sold, the user can specify a minimal and maximal price. OPTOSA will autonomously monitor the market situations, especially the prices of the competitors, and finally adapt the product price in such a way that turnover and margin is optimized. Other functionalities can be only partially automated and serve as decision support for the user. For instance, when description text or search tags are to be defined, OPTOSA will scan other, comparable product descriptions on the sales platforms, and evaluate which key words correspond to sales success. From this it computes suggestions that are presented to the user. In any case, the input data processed by the rules is information extracted through data mining from recorded characteristics of past sales. The business knowledge to process the data is specified and tested in partnership with intended users of the system.

3 Data handling

In order to tackle the optimization of online sales, data about sellers and products have to be collected, preprocessed, analyzed and finally used for extracting knowledge models for decision support.

In fact, the OPTOSA platform aims at providing decision support features [TPD+11] by combining:

- The insights that are automatically discovered from data (for instance: 'the price of a given product should be lower than the price of similar concurrent products, with at least a difference of X%').
- The business know-how that is brought by the user (for instance: some rules on prices to take into account margin).

As a result, the goal is to provide *predictive models* to help *price positioning*: a typical output can be a model that evaluates the price of given product (according to the product features and the similar products prices). The following sections briefly describe each step of the process that supplies the OPTOSA platform (Figure 1).

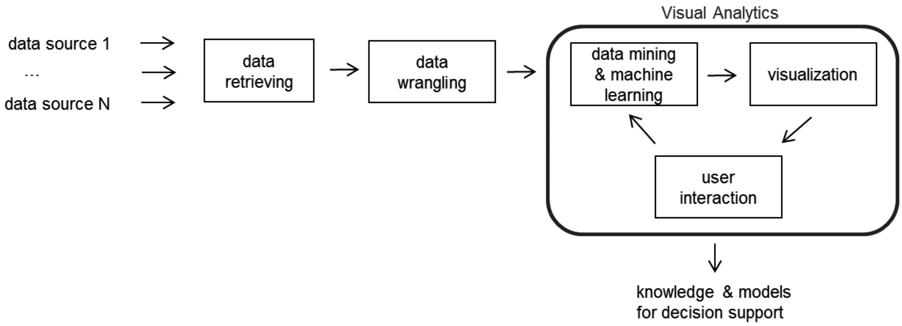

Figure 1: The worflow that is applied into the OPTOSA platform for data handling.

Recent papers about *data analysis* argue that *Visual Analytics* is needed to build accurate models by considering both data and *business knowledge* [KAF$^+$08, KHP$^+$11]. But is it realistic to apply the straightforward *Visual Analytics* approach in a real-time system which continuously manages huge heterogeneous data? The OPTOSA platform aims at combining state-of-the art techniques and pragmatical choices in order to provide operational results.

3.1 Data retrieving

By using online marketplace platforms like *eBay* or *Amazon*, it is possible to retrieve and aggregate a lot of data about products and sellers. The best way is to use the API that are provided by these systems. Generally stable and well-documented, these API allow to get data easily in order to store them into a *local* database for further analysis. In addition, the produced database can be completed by using services provided by dedicated data providers like icecat ([ice]).

3.2 Data wrangling

Traditionally used as a preliminary step in data visualization and knowledge extraction, data wrangling is a semi-automated process that aims at transforming data in order to facilitate further usage of them [KHP$^+$11]. More precisely, data wrangling consists in iteratively applying different kinds of preprocessing tasks like *inconsistent data cleaning*, *missing values imputation* or *sampling* [ET98]. As it induces modifications and information loss, data wrangling has to be carefully applied; in addition, it is a painful and time-consuming task that requires efficient tools [KPHH11].

We plan to use different tools to wrangle data: Excel for quick transformation prototyping, and specific modules for more complexes tasks into the OPTOSA system.

3.3 Decision support & Visual Analytics: data mining, machine learning, visualization, user interaction

Once the data are ready to be analyzed, a classical *cascading process* can be implemented in order to build efficient predictive models [CTTB06]: in order words, the idea is to mix unsupervised and supervised machine learning methods in order to obtain efficient models for each kind of products. In a few words, the process is the following:

- *Features selection* in order to filter the data that are the most significant [YL03].
- *Clustering* in order to regroup similar objects and to detect/ignore outliers. [Jai10].
- Computation of *predictive models* for each cluster. In this field, regression trees are helpful and popular because they use an intuitive formalism that is easy to understand for domain experts [B$^+$84, Mur98]. Moreover, they can be built from data by using well-known algorithms like CART [B$^+$84] and M5 [Qui92] (Figure 2).

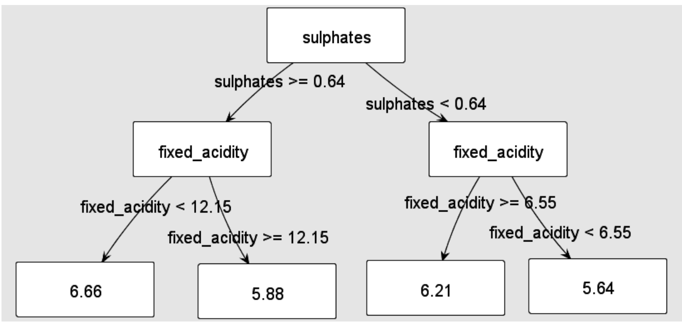

Figure 2: A regression tree obtained from the 'winequality-red' dataset [BL13], to evaluate the wine quality score: a score of 0 represents a poor wine and a score of 10 represents an excellent wine.

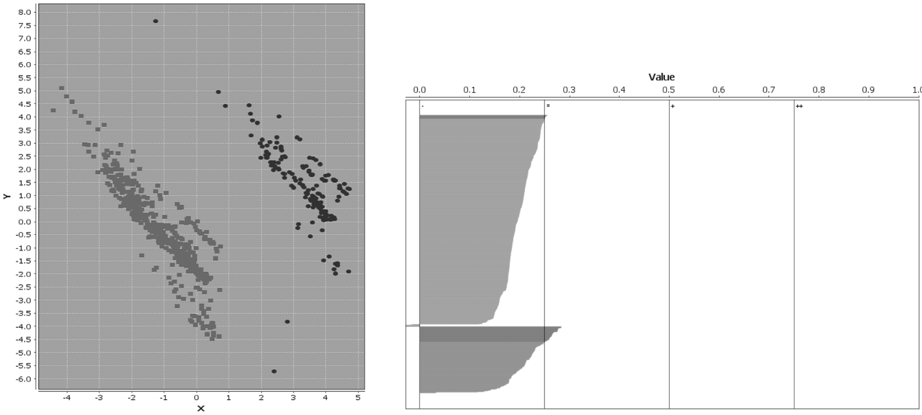

Figure 3: Visualizations to inspect a clustering result ('*credit-a*' dataset, clustered with k-means). At the left, a colored PCA projection: the x axis represents the first principal component, and the y axis represents the second principal component. At the right, a Silhouette plot: the x axis represents the Silhouette value, and the y axis represents the elements (sorted by clusters and Silhouette values). In practice: larger Silhouette values indicate a better quality of the clustering result.

Even if this process is well-known, obtaining operational results is not straightforward: in fact, various parameters can have a great influence on the final results. For instance, the usage of clustering is not trivial and depends of the final application [AR13]: what kind of clustering algorithms should be used? Which settings have to be applied? Should some data be manually processed? Generally speaking, applying machine learning can be considered as a *black art* that requires a strong background and experiment [Dom12].

However, pragmatical solutions can be considered. For instance, by following the *Visual Analytics* approach [KAF$^+$08], and by providing interactive features to the final user, results can be produced, evaluated, visually inspected and refined in order to obtain clusters are simple to interpret and predictive models that are efficient.

More precisely:

- Clustering can be adapted to refine the predictive models: in an automatic way [PDBT14], or by taking into account the user domain knowledge.
- Predictive models can be simplified by using various simplification techniques: for instance, a recent technique has been proposed to simplify them by using data preprocessing [PBDT13].
- Results can be visualized using simple scatter plots or more sophisticated visualizations like MDS projections [KW78].

As a result, the predictive models can be used to support decision regarding the *price positioning* objective.

4 The OPTOSA platform

A first prototype has been implemented in Java as a standalone tool. The core system is based on several libraries that provide building blocks for *Visual Analytics* and Decision Support:

- Data wrangling, data mining and machine learning features are provided by Weka, a widely-used data mining library [WFH11]. More precisely, it provides several implementations for features selection, regression trees induction and clustering.
- As the Weka licensing policy can cause further issues in the case of commercial uses, some alternatives to Weka have been tested and integrated: Sandia Cognitive Foundry ([BBD08]) and apache-commons ([Com]). These libraries are two main advantages: they provide implementations of the state-of-the-art techniques, and they are distributed under licences that are permissible and compliant to potential commercial use.
- For the graphical representation of the price evaluation model, the tool uses Jung, a robust and well-known graphical library [OFWB03].

In addition to these building block components, the prototype is completed with a graphical interface for manipulating data, computing predictive models, exploring the results (statistics about the datasets, visual representations of the clusters and the predictive models, etc.) and taking into account the user knowledge (Figure 4).

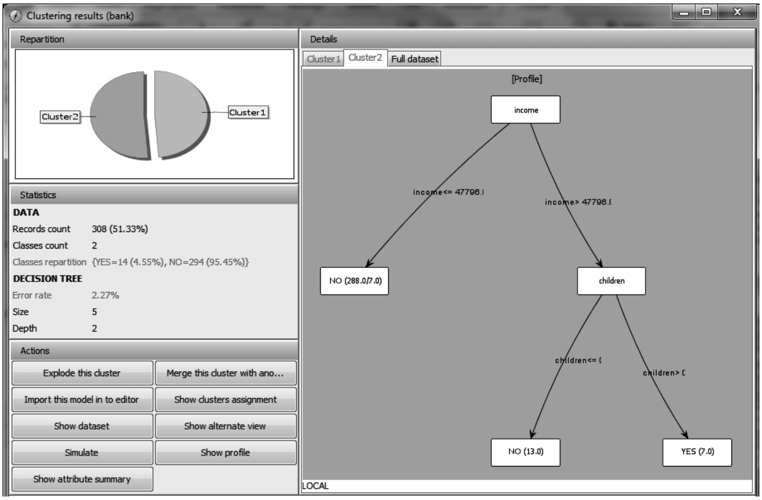

Figure 4: The prototype module that allows to check and refine if needed the clustering results.

5 Use-case

For our first technical trials, we processed datasets relevant to different business use cases. As explained in the previous sections, the knowledge is formalized at the end on the shape of decision trees and regression trees [Kot13], based on a user readable formalism, the trees being directly extracted from real life data.

Data considered as input are: the list of products and information about each product (features description, prices, sales ranks, etc.). The information can be structured in the shape of clusters of products, and for each cluster the products profile is defined (core features, prices ranges, sales rank ranges, etc.). The use cases are groups of tasks defining some concrete sales activity, such as the following examples. Some use cases are principally handled by data mining (on relevant data) such as 'Price fixing' based on similar products. Other ones contain more domain related knowledge (from vendor or buyer point of view) such as 'Calculation of a similarity index between products'.

In Figure 5, we illustrate the 'Price positioning' use case for a laptop. First, from a dataset, a decision tree is computed to determine the price range according to the features of the laptop. The decision model is then applied to evaluate and justify the price of a given product. Second, the knowledge is used for the computation of a clustering of the laptops recorded on a dataset, according to their main features. A planar projection of the laptops shows the correspondence between the features (horizontal axis) and the price (vertical).

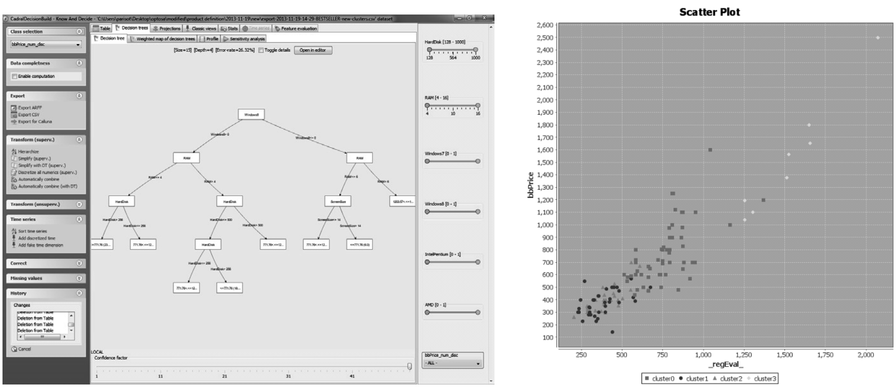

Figure 5: *Visual Analytics* for price positioning.

By using these models, a user interface has been built to interactively evaluate the price of a given product (Figure 6). In this interface, the evaluation is completed by a justification: here, the justification corresponds to the rules that are *verified* for the given product. In addition, several data mining features have been added to complete the evaluation: for instance, the 'k-nearest neighbours' technique allows to automatically find the products that are the most similar to the given product, so the price positioning result can be checked.

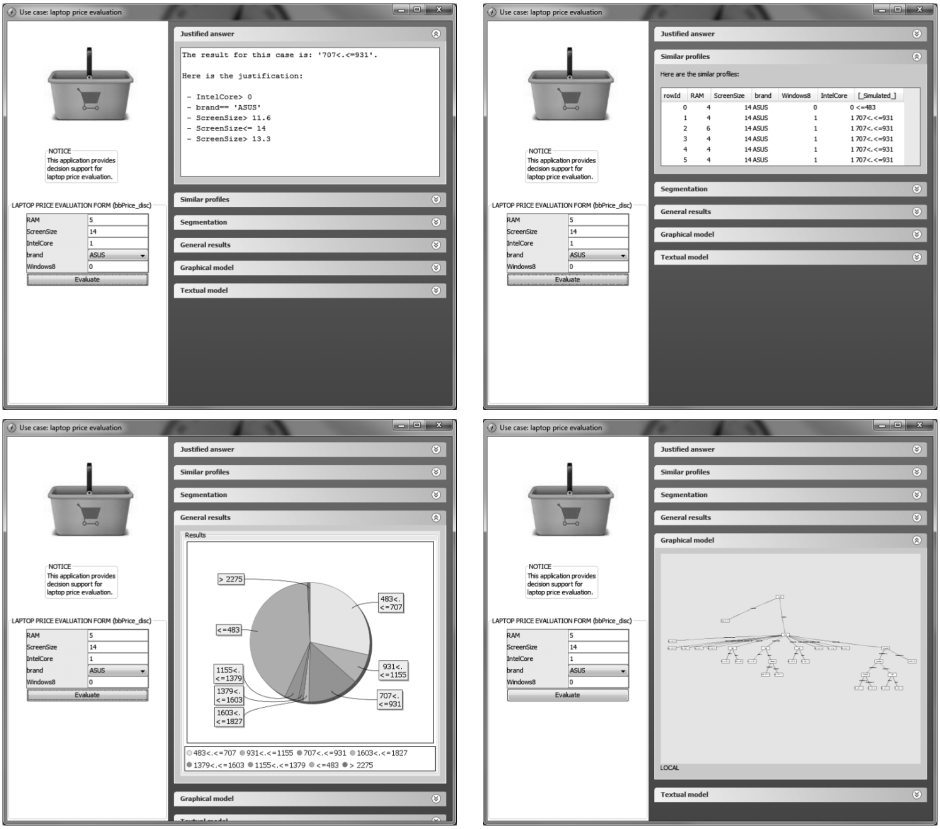

Figure 6: User interface to evaluate the price of a given product. At the top left, the price and the justification are given. At the top right, the similar products are shown, in order to check if the evaluated price is compliant with the others. At the bottom left, the clusters are represented into a pie chart. At the bottom right, the graphical model is presented.

The processing of the use cases shows the possibility to extract knowledge, track evolutions and build decision models. From the debriefing on the data mining results, it was also clear that a lot of business information must be taken into account to reinterpret the results, in order to be able to formalize rules that could automate the handling of marketing activities.

Finally, the integration of expert of business knowledge is a critical step for decision support: to handle this issue, the prototype provides a module to edit the price evaluation model (Figure 7). By using it, the user is able to adapt and refine the price positioning module according to the business requirements.

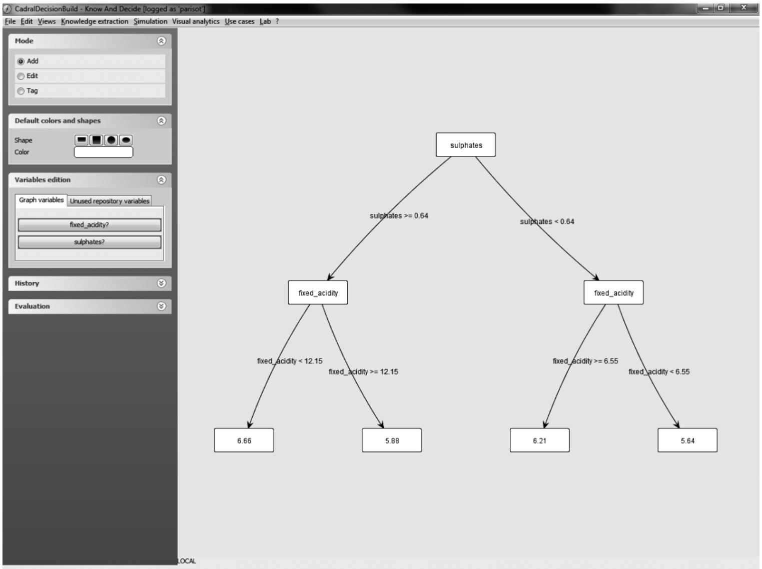

Figure 7: Graphical user interface for the edition of the price evaluation model: the structure of the regression tree can be changed and/or the rules can be modified by the expert.

6 Conclusion

In this paper, a software architecture has been proposed to optimize the online sales: the OPTOSA platform aims at integrating both clever methods for data processing and visualization techniques in order to provide an efficient and useful decision support.

Drawing from these promising results, the innovative approach of OPTOSA will be carried on and refined to efficiently support the work of both manufacture companies and distribution channels. The monitoring of online sales include many time consuming tasks for the human operator that could be automated by enriching OPTOSA with business knowledge. The user will thus be free to concentrate on crucial parts of the process through relevant interaction with the system. Thanks to our operational partnership, further steps towards such integrated tool will continue to be regularly assessed on live data from market places.

7 Acknowledgments

The project OPTOSA is supported by a grant from the Ministry of Economy and External Trade, Grand-Duchy of Luxembourg, under the RDI Law.

References

[Adv] Channel Advisor. http://www.channeladvisor.com.

[AR13] C.C. Aggarwal and C.K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC D.M. and K.D. Series. Taylor & Francis, 2013.

[B+84] Leo Breiman et al. *Classification and Regression Trees*. Chapman & Hall, 1984.

[BBD08] Justin Basilico, Zachary Benz, and Kevin R Dixon. The cognitive foundry: A flexible platform for intelligent agent modeling. In *Proceedings of the 2008 Behavior Representation in Modeling and Simulation (BRIMS) Conference*, 2008.

[BL13] K. Bache and M. Lichman. UCI Machine Learning Repository, 2013.

[Com] Apache Commons. http://commons.apache.org/.

[CTTB06] Laurent Candillier, Isabelle Tellier, Fabien Torre, and Olivier Bousquet. Cascade evaluation of clustering algorithms. In *Machine Learning: ECML 2006*, pages 574–581. Springer, 2006.

[Dom12] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[ET98] Robert Engels and Christiane Theusinger. Using a Data Metric for Preprocessing Advice for Data Mining Applications. In *ECAI*, pages 430–434, 1998.

[HA08] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. *Information visualization*, 7(1):49–62, 2008.

[ice] icecat. http://icecat.lu/.

[Jai10] Anil K Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008.

[KHP+11] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Inf. Vis.*, 10(4):271–288, 2011.

[Kot13] S.B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

[KPHH11] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive Visual Specification of Data Transformation Scripts. In *Proceedings of the SIGCHI Conference on H.F.C.S.*, CHI '11, pages 3363–3372, NY, USA, 2011.

[KW78] Joseph B Kruskal and Myron Wish. *Multidimensional scaling*, volume 11. Sage, 1978.

[Mur98] Sreerama K. Murthy. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Min. Knowl. Discov.*, 2(4):345–389, 1998.

[OFWB03] J. O'Madadhain, D. Fisher, S. White, and Y. Boey. The JUNG (Java Universal Network/Graph) Framework. Technical report, UCI-ICS, October 2003.

[PBDT13] Olivier Parisot, Pierrick Bruneau, Yoanne Didry, and Thomas Tamisier. User-Driven Data Preprocessing for Decision Support. In Yuhua Luo, editor, *CDVE*, volume 8091 of *LNCS*, pages 81–84. Springer Berlin Heidelberg, 2013.

[PDBT14] Olivier Parisot, Yoanne Didry, Pierrick Bruneau, and Thomas Tamisier. Data visualization using decision trees and clustering. In *5th International Conference on Information Visualization Theory and Applications (IVAPP 2014), Lisbon, Portugal*, 2014.

[Qui92] John R Quinlan. Learning with continuous classes. In *Proceedings of the 5th Australian joint Conference on A.I.*, volume 92, pages 343–348. Singapore, 1992.

[Sis] Sistrix. `http://www.sistrix.com`.

[TPD+11] Thomas Tamisier, Olivier Parisot, Yoann Didry, Jérôme Wax, and Fernand Feltz. Adapting decision support to business requirements through data interpretation. In *Cooperative Design, Visualization, and Engineering*, pages 82–85. Springer, 2011.

[WFH11] Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

[Wis] Wiseseo. `http://www.wiseseo.de`.

[YL03] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

Business Process as a Service – Status and Architecture

Thomas Barton[1], Christian Seel[2]

[1]University of Applied Sciences Worms
Erenburgerstr. 19
67549 Worms, Germany
barton@hs-worms.de
[2]University of Applied Sciences Landshut
Am Lurzenhof 1
84036 Landshut, Germany
seel@haw-landshut.de

Abstract: A Cloud Service is usually classified as Infrastructure as a Service, Platform as a Service or Software as a Service. This classification ist not sufficient, when Business Processes are considered. Therefore, Business Process as a Service (BPaaS) as next level of abstraction is in discussion. BPaaS is already an important topic for analysts and cloud providers. In the scientific community a discussion about BPaaS has been started. This contribution presents a literature analysis of the current state-of-the-art in BPaaS. In order to investigate how a Business Process can be built on top of a cloud service, a prototpye of an external application is presented, which is built on top of a cloud service using a RESTful API. For the realization of BPaaS existing architectures for cloud computing are discussed. A proposal for an advanced Architecture for Business Process as a Service is presented. Based on these findings, a brief outlook on future research questions concerning Business Process as a Service is derived.

1 Business Process as a Service

A common classification of cloud services has been developed by the National Institute of Standards and Technology (NIST), which assigns cloud services to three layers [LTMBMBL11]

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS) and
- Software as a Service (SaaS).

This classification starts with IaaS, which refers to basic services e.g. for storage and reaches up to SaaS, where applications are provided as a service. In this classification each level abstracts from the functionality of the level below. The next level of abstraction above SaaS is Business Process as a Service (BPaaS) [SPBB12]. BPaaS means not only that software is provided to the user as a service. BPaaS means as well to deliver the logic and control flow of the business process the user wants to execute as a service. As a result, a classification of services is obtained as shown in Figure 1.

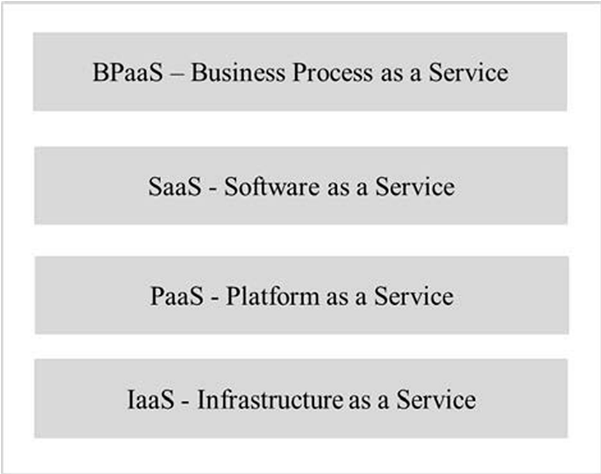

Figure 1: Classification of Cloud Services

Despite of the fact that consulting and software companies as well as market analysts are using the term BPaaS frequently, neither the exact meaning nor a concrete architecture of a BPaaS system are clearly defined in the scientific literature. Therefore this paper presents a state-of-the-art analysis of BPaaS, which starts with a literature analysis. As Business Processes as a Service have to be set up on existing cloud services, a protoype of an app is presented, that runs on top of a cloud service. Based on that analysis and the discussion of different architectures for cloud computing, an extension of current architectures is presented.

2 BPaaS in Literature

The state-of-the-art analysis of BPaaS starts with a review of the scientific literature on that topic.

2.1 Methodology

The lack of rigour of scientific literature analysis in the information systems discipline is often criticized. This criticism is especially related to the missing documentation and traceability of the analysis. Based on that analysis a procedure model for a literature analysis was proposed that consists of five steps [vBSNRPC09]: Step 1 to 3 addresses the preparation and literature selection and step 4 and 5 the analysis and classification of the results. In order to achieve traceability the literature analysis is based on that approach. The first step is the definition of the research questions for the literature analysis. This paper's research questions are:

- Question 1: How common is BPaaS in the scientific literature?
- Question 2: What are the research perspectives on BPaaS?

Step 2 is the conceptualization of the literature search process. Therefore search terms and context has to be defined [Co88]. For this analysis "Business Process as a Service" and "BPaaS" are defined as search terms. The context in which the search takes place is the international scientific literature in the field of information systems. Step 3 is the selection of the literature base in which the analysis should be done. This decision is directly related to the context defined in the step before. For this analysis these well-known two portals and two journals are chosen: SpringerLink, EBSCO, Business & Information Systems Engineering (BISE), MIS Quarterly (MISQ). Publications in English and German language have been considered in the search. Step 4 and 5 of the analysis are the literature search itself and the analysis of the publications found. The results of these steps are presented in the next section.

2.1 Results

The literature search, step 4, reveals that the keywords "Business Process as a Service" and "BPaaS" match with 40 publications in SpringerLink and with 59 in the EBSCOHost. Figure 2 shows the matches by keyword and portal or journal:

Searched in	"Business Process as a Service"	"BPaaS"
SpringerLink	40	24
EBSCO	59	36
BISE	0	0
MISQ	0	0

Figure 2: Literature overview

For step 5 the search results have been refined by an analysis of the title and the abstract of each publication that was found. During this process all publications that don't have a strict relation to BPaaS and all publications that were found in non-scientific media have been eliminated. Especially on EBSCOHost many publications from analysts or newspapers have been found. The result of this process is 16 BPaaS publications. In order to answer the first research question all publications have been displayed in a chart that shows the number of publications per year on BPaaS (Figure 3). The chart illustrates that only two relevant publications have been published before 2011. The largest number of publications is found in 2013. These numbers indicate that BPaaS is an actual topic in nowadays and starts to get in the focus of scientific research at the moment.

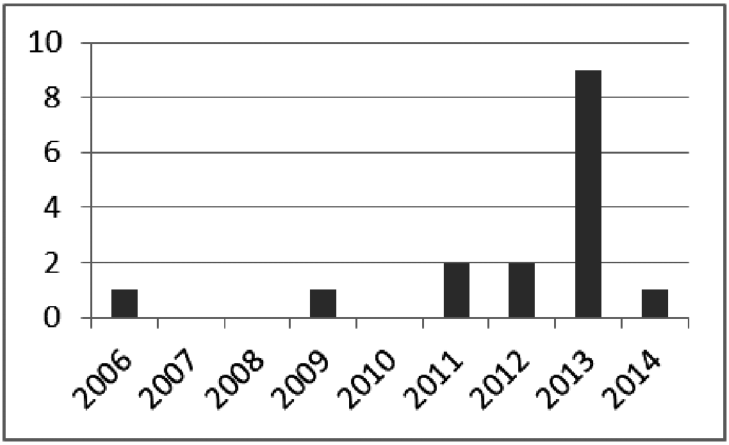

Figure 3: Publication on BPaaS per year

The second research question is about the perspective on BPaaS that the analyzed publications take. All found publications have been classified into four perspectives that are shown in Figure 4:

Publication	Classification
[YW13]	Business Perspective
[AR12]	Business perspective
[HHLM13]	Business perspective
[LM06]	Business perspective
[Be14]	Business perspective
[ME13]	Business perspective
[Fe 13]	Business perspective
[GWvLJ13]	Development
[HSGMB09]	Development
[WITZYHL12]	Development
[OOM13]	Development
[Ra13])	Development
[MH11]	EA in the Cloud
[Ma11]	EA in the Cloud
[MA13]	Security
[Pe13]	Security

Figure 4: Classification of selected BPaaS publications

The following paragraphs describe the specific classifications in more detail:

Business Perspective

The seven publications that regard BPaaS form a business perspective focus on economic aspects of the business software. The major topic is outsourcing of business processes into the cloud or special business tools that are provided in the cloud. All these publications have in common that they stress the impact of BPaaS or its measurement and not the architecture or implementation of BPaaS.

Development

The five publications that have been classified in the development perspective focus just the opposite of the business perspective publications. Their focus is on the architecture, implementation or on technology stacks that could be used to implement cloud services and BPaaS. The work presented in these papers is often based on standards like BPEL or RESTful services.

EA in the Cloud

The two publications that are classified in the enterprise architecture perspective regards how cloud services can be integrated in the software architecture of a company. The found publications in that category describe fundamentals and basic principles on the topic. In relation to the publications form the business and the development perspective these papers are in-between these two categories. They are less technical than the development publications but are more focused on the realization und integration of BPaaS in a company than the business perspective publications.

Security

The last category is security. The two publications in that category describe security lags of processes in the cloud and privacy and security concerns. They analysis what security problem occur by outsourcing business processes into the cloud. But a final solution to this problem is still an unsolved research question.

The literature analysis reveals that BPaaS has gained importance in the scientific literature in 2013. Despite of the increasing impact most publications neither define the term BPaaS nor do they propose how to build BPaaS applications. The development of BPaaS applications requires to develop applications on top of cloud services. Therefore, the following sections start with the presentation of an application, which was built on top of a cloud service. Finally, the architecture of BPaaS applications is considered, where a proposal for an advanced Architecture for Business Process as a Service is presented.

3 Building Applications on top of Cloud Services

This chapter follows the question how to build an applications on top of a cloud service. In order to show how an external application can be built, an application is presented, which has been developed on top of the popular OpenStack software using so called RESTful APIs.

3.1 RESTful APIs

APIs are used to connect different applications. And, new applications are developed by integrating existing applications with each other. Such a new application is called mashup. A lot of APIs, which are used in order to develop mashups, rely on an architectureal style, which is called REST (REpresentational State Transfer). The main principles of RESTful Services are as follows [Fi00, Ti11, Ba14]:

- RESTful Services are based on resources, which are identified by URIs. In general, any source of information, which serves a special purpose, describes a resource. The weather forecast of a city or the descriptions of a product are examples of a resource. A collection of resources e.g. all products of a specific product group is also a resource.
- RESTful Services have a uniform interface. In case, data have to be retrieved (read only) within an order-management application, there is only one single method, which has to be carried out. As an example, both order and product data are retrieved with the same method.
- For every Resource, data should be available in different data formats. As an example, a consumer should be able to request information for a product in an XML or JSON format.
- The communication within a RESTful Service is stateless. A change of state may be recorded by a provider by means of a changed resource. On the other hand, it may be the task of a consumer to handle the state of an application.
- A RESTful Service uses Hypermedia to carry out follow-up processes. In case, a consumer has carried out an order, the Service reply should contain hyperlinks which allow e.g. to check or cancel this order.

In case of internet applications HTTP is commonly used as protocol. As a consequence, the uniform interface is formed by the well-known HTTP methods GET, POST, PUT and DELETE.

3.2 The OpenStack project

There are a lot of projects, which offer RESTful APIs. In the area of Infrastructure as a Service, OpenStack is a very popular example. OpenStack is an open source solution for a standard cloud computing infrastructure for both public and private clouds [OS].

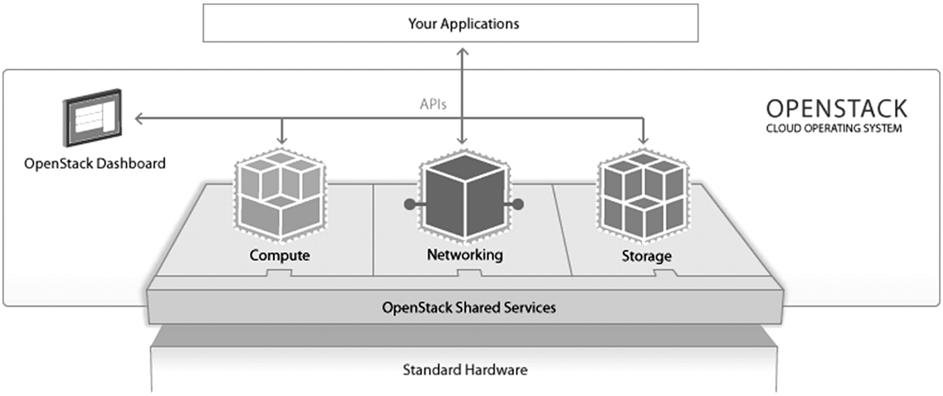

Figure 5: Overview of the OpenStack Software [OS]

The OpenStack software, as shown in Figure 5, is organised around three major concepts: compute, networking and storage. Additionally, there are shared services like e.g. identity management. The OpenStack dashboard is the graphical interface for administrators and users. Different OpenStack components are integrated with each other by means of RESTful APIs. For this purpose every OpenStack project offers a RESTful API [OSAPIs]. Therefore, it is on one hand easy to continuously enhance the functionality of OpenStack by means of further projects. On the other hand, building applications on top of cloud services is simplified, when RESTful APIs are available.

3.3 Building an app on top of OpenStack

OpenStack contains „Shared Services", which offer several functions for all components via RESTful services. In case of functions for authentification and user administration the Identify API is responsible to answer internal and external requests. The main resources for a user administration and there relation to the unifrom interface, which consists of four methods, are represented in Figure 6. The resources are as follows: The accumulative resource „/v2.0/users" stands for all users. The resource „/v2.0/users/{userId}" identifies a single user with identification numberer {userId}. All operations, which are carried out for a single user, like e.g the change of the password, occur at resource „/v2.0/users/{userId}". Method GET serves for requesting information for a resource. GET is used with the accumulative resource to list up all users and to receive the details of a single user, when it is applied to the resource, which identifies a single user. By means of method PUT a single resource is updated, it is not used in relation to a accumulative resource. The creation of a new user is carried out, when applying method POST on the accumulative resource „/v2.0/users". POST is not used in relation to a single resource. DELTE serves to delelte single users. The deletion of all users via the accumulative resource is not implemented in the Identify API.

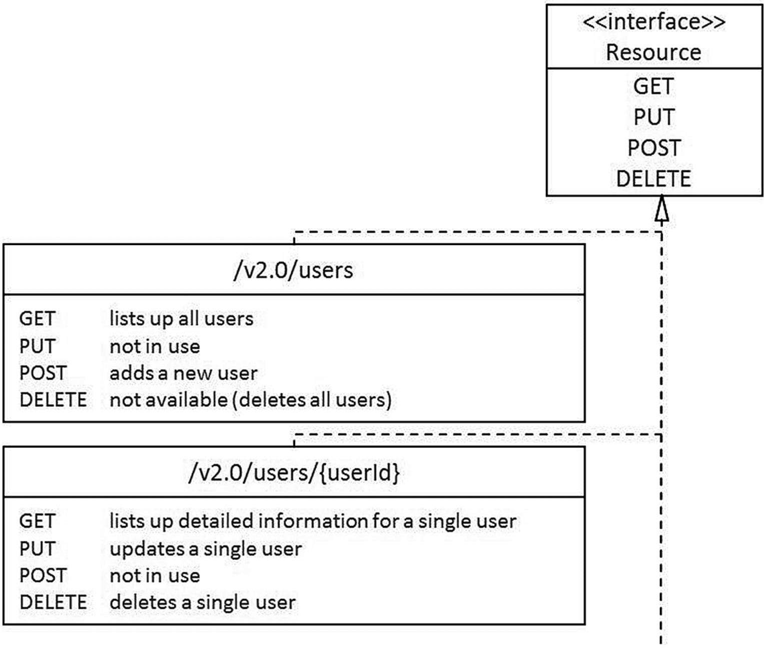

Figure 6: Main resources for user administration

The Identity API is used to build an external application on top of OpenStack. This application runs on a Android smartphone. It has been developed in Java using Eclipse as an integrated devevloment enviroment. In order to develop this application the Identify API is applied to the resources, which are shown in in Figure 6. The application contains four functions, which are shown in a screen shot (Figure 7).

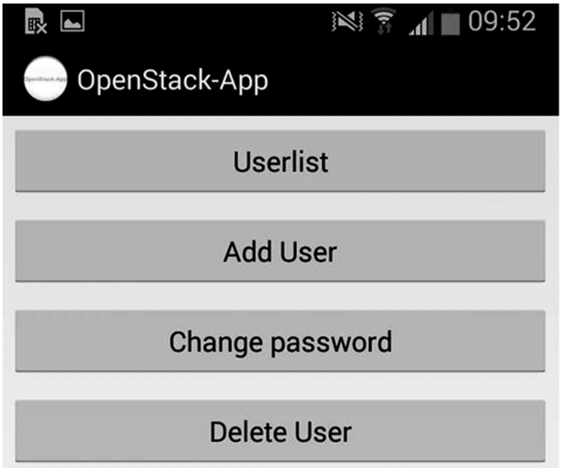

Figure 7: App for user administration (overview)

The application allows to list up all users, to add a single user, to change the password of an existing user and to delete a single user. The application interacts as an external application with a Identify service in OpenStack by means of a RESTful API.

4 BPaaS Architectures

This chapter contains the closing part of the state-of-the-art analysis. It examines the architecture for cloud computing in order to realize BPaaS.

4.1 Cloud Computing Reference Architecture

The US National Institute of Standards and Technology (NIST) define a Cloud Computing Reference Architecture. The main actors are Cloud Consumer and Cloud Provider. A Cloud Consumer is an individual or organization that acquires and uses cloud products and services. A Cloud Provider is a person, an organization or an entity responsible for making a service available to a Cloud Consumer [LTMBMBL11]. Within this work a scenario is considered, where only the main actors Cloud Consumer and Cloud Provider are involved. Other actors are not considered here.

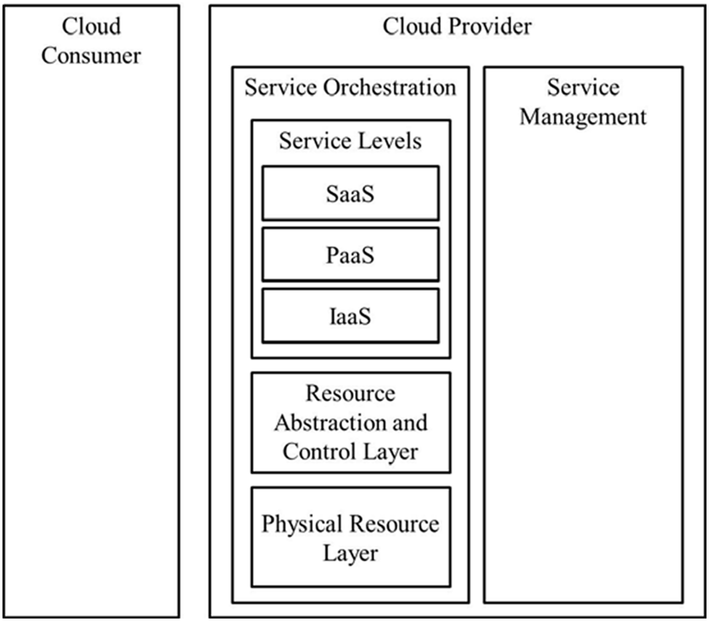

Figure 8: Representation of the NIST cloud computing reference architecture (only two actors and two activities are shown)

Figure 8 shows an overview of the NIST cloud computing reference architecture with the following restrictions: Only Cloud Consumer and Cloud Provider are considered as actors. In addition only two activities are shown: Service Orchestration and Service

Management. Service Orchestration refers to the arrangement, coordination and management of cloud infrastructure to provide different cloud services in order to meet IT and business requirements. The service layers within a Cloud Provider define the interfaces for Cloud Consumers to access cloud services. The resource abstraction and control layer contains system components to provide and manage access to physical computing resources. The physical resource layer includes all computing resources. Service Management implies all the service-related functions that are necessary for the management and operations of services for Cloud Consumers [LTMBMBL11]. Within it's Cloud Computing Reference Architecture IBM has defined an own reference architecture for cloud computing. In addition to the existing three Service Models IaaS, PaaS and SaaS, IBM defines a fourth Service Model named Business Process as a Service (BPaaS) [SPBB12].

4.2 BPaaS Architecture and RESTful Services

In order to set up an architecture diagram for Business Process as a Service the cloud computing reference architecture, as suggested by NIST and as shown in Figure 8, is taken as a starting point. To consider BPaaS, another layer has to be introduced. In contrast to IBM's cloud computing reference architecture BPaaS should not extend the already existing layer for the Service Levels, which belong to the Cloud Provider, because a layer for Business Process as a Service cannot be strictly allocated to either a Cloud Provider or a Cloud Consumer or any other actor.

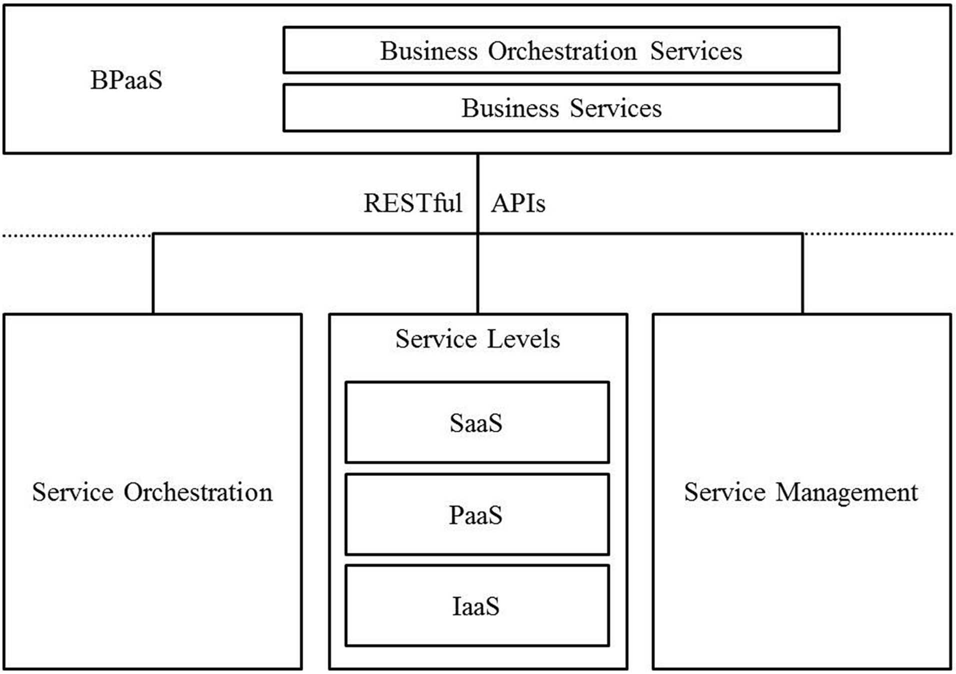

Figure 9: Architecture for Business Process as a Service

There has to be always the choice for an enterprise to determine where a layer for Business Process as a Service should be located. Certainly, a lot of CIOs would not agree to delegate Business Processes to any actor outside their companies. Therefore, a layer for BPaaS should extend already existing layers without any assignment to an actor. As a consequence Cloud Consumer and Cloud Provider cannot be included in an architecture diagram for BPaaS. Business Process as a Service requires functions to set up Business Services and to orchestrate Business Services within on demand applications. As a sufficient standardization RESTful APIs shall be available in order to connect to different applications and projects. RESTful Services could serve as enabler to connect software components with each other (Figure 9).Furthermore the BPaaS layer on top of the three standard NIST layers can be used to integrate different software business models. In order to integrate different software business models in order execute one Business Process the following scenario is considered (Figure 10). It shows one on-demand application in the area of Customer Relationship Management (CRM) and one on-premise application for Enterprise Resource Planning (ERP). The BPaaS layer on top of the SaaS layer can be used to integrate these different application business models. This can be realized by an orchestration service on the BPaaS layer. And, the BPaaS layer can either be controlled by the cloud provider or by the cloud Customer or in a hybrid way, where the control is distributed between these two parties.

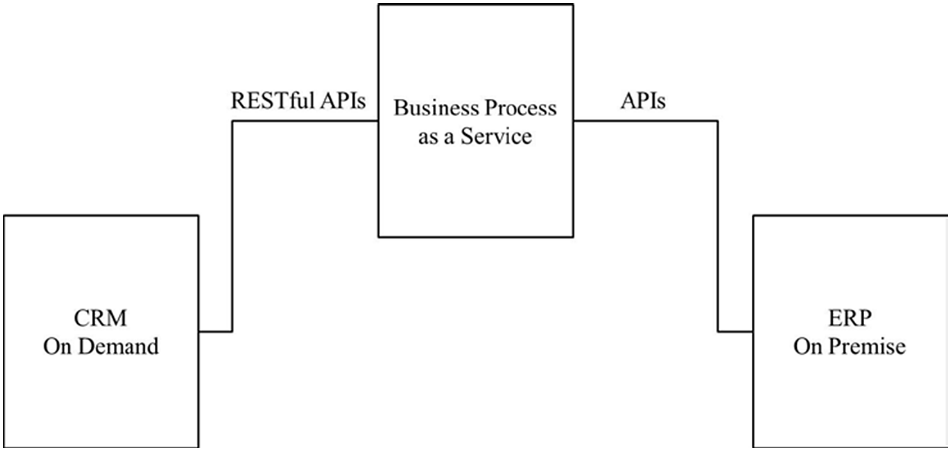

Figure 10: BPaaS scenario for an on-demand and an on-premise application

5 Summary and Outlook

The literature analysis in chapter 2 reveals that the number of papers on BPaaS has increased during the last two years, which indicates that BPaaS has been put on the research agenda.

By means of an app for OpenStack, as shown in chapter 3, a proof of concept has been presented, that an external application can be built on top of a cloud service, where

REST APIs are used as interfaces to connect the external application with a cloud service.

IBM's Cloud Computing Reference Architecture has been analyzed. As a result, an additional BPaaS layer on top of the Cloud Computing Reference Architecture, as suggested by NIST, has been included, which is not necessarily under control of the cloud provider. RESTful APIs shall be used to connect services on a BPaaS layer with cloud services on IaaS, BPaaS or SaaS layers. The new BPaaS layer may be used to integrate the two different software business models, on demand and on premise.

Despite of the research presented in the reviewed paper and the architecture discussion presented here, there are further open requirements, which should be subjects of future research:

- First of all, a consistent description on Business Process as a Service with clear definition of the term has to be formulated.
- Secondly, a concept for distributed REST based modeling and design of Business Processes has to be developed.
- At third, it has to be clarified, how the orchestration of such a Business Process that is located into a BPaaS layer on top of the SaaS layer is carried out. Furthermore the prototypic implementation of such a Business Process as a Service system has to be considered.

References

[AR12] Abelló, A.; Romero, O.: Service-Oriented Business Intelligence. Lecture Notes in Business Information Processing Volume 96, 2012, pp 156-185.

[Ba14] Barton, T.: E-Business mit Cloud Computing. Springer Wiesbaden, 2014.

[Be14] Bensberg, F.: Cloud-Lösungen rational bewerten. In: Controlling & Management Review, March 2014, Volume 58, Issue 7, pp 66-73.

[CF] Cloud Foundry. http://docs.cloudfoundry.org, accessed on 05/20/2014.

[Co88] Cooper, H.: Organizing knowledge syntheses: A taxonomy of literature reviews. In: Knowledge in Society, Spring 1988, Volume 1, Issue 1, pp 104-126.

[Fe13] Ferreira, D.: Evolution of Enterprise Systems. In: Enterprise Systems Integration. 2013, pp 3-14.

[Fi00] Fielding, R. T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine, 2000.

[GWvLJ13] Gey, F.;Walraven, S.;van Landuyt, D.; Joosen, W.: Building a Customizable Business-Process-as-a-Service Application with Current State-of-Practice Software Composition. In: Lecture Notes in Computer Science Volume 8088, 2013, pp 113-127.

[Go] Google App Engine. https://cloud.google.com/products/app-engine/, accessed on 05/20/2014.

[He] Heroku. https://www.heroku.com/, accessed on 05/20/2014.

[HHLM13] Hill, R.; Hirsch, L.; Lake, P. Moshiri, S.: Enterprise Cloud Computing. Guide to Cloud Computing. In: Computer Communications and Networks 2013, pp 209-222.

[HSGMB09] Höing, A.; Scherp, G.: Gudenkauf, S.; Meister, D.; Brinkmann, A.: An Orchestration as a Service Infrastructure Using Grid Technologies and WS-BPEL. In: Service-Oriented Computing, Lecture Notes in Computer Science Volume 5900, 2009, pp 301-315.

[LTMBMBL11] Liu, F.; Tong, L.; Mao, J.; Bohn, R.; Messina, J.; Badger, L.; Leaf, D.: NIST Cloud Computing Reference Architecture. NIST Special Publication 500-292, 2011 (http://www.cs.cmu.edu/~garth/15719/papers/nist_cloud_computing_reference.pdf, accessed on 05/20/2014).

[LM06] Longo, A.; Motta, G.: Design Processes for Sustainable Performances: A Model and a Method Business Process Management Workshops. In: Lecture Notes in Computer Science Volume 3812, 2006, pp 399-407.

[Ma11] Mahmood, Z.: Cloud Computing for Enterprise Architectures: Concepts, Principles and Approaches. In: Cloud Computing for Enterprise Architectures. Computer Communications and Networks 2011, pp 3-19.

[ME13] Meyer, K.; Eul, M.: IT-Outsourcing-Trends – und wie Unternehmen erfolgreiche Partnerschaften gestalten. In: IT-Outsourcing, 2013, pp 71-89.

[MH11] Moshiri, S.; Hill, R.: Enterprise Architecture Fundamentals. In: Cloud Computing for Enterprise Architectures. Computer Communications and Networks 2011, pp 21-41.

[MA13] Müller, G.; Accorsi, R.: Why Are Business Processes Not Secure? In: Number Theory and Cryptography. Lecture Notes in Computer Science Volume 8260, 2013, pp 240-254.

[OOM13] Okuda, H.; Ogata, S.; Matsuura, S.: Experimental development based on mapping rule between requirements analysis model and web framework specific design model. In: SpringerPlus, March 2013, 2:123.

[OS] OpenStack. http://www.openstack.org, accessed on 05/20/2014.

[OSAPI] OpenStack API Complete Reference. http://api.openstack.org/api-ref.html, accessed on 05/20/2014.

[Pe13] Pearson, S.: Privacy, Security and Trust in Cloud Computing. In: Privacy and Security for Cloud Computing. Computer Communications and Networks 2013, pp 3-42.

[Ra13] Ramachandran, M.: Business Requirements Engineering for Developing Cloud Computing Services. In: Software Engineering Frameworks for the Cloud Computing Paradigm. Computer Communications and Networks 2013, pp 123-143.

[SPBB12] Stifani, R.; Pappe, S.; Breiter, G.; Behrendt, M.: IBM Cloud Computing Reference Architecture. IBM Academy of Technology Volume 3, Number 1, 2012. http://www-05.ibm.com/it/cloud/downloads/Cloud_Computing.pdf, accessed on 05/20/2014.

[Ti11] Tilkov, S.: REST und HTTP, 2nd edition, dpunkt, Heidelberg, 2011.

[vBSNRPC09] Vom Brocke, J; Simons, A; Niehaves, B; Riemer, K; Plattfaut, R; Cleven, A: Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: Proceedings of the 17. European Conference On Information Systems (ECIS), Verona 2009.

[WITZYHL12] Wu, Z.; Itälä, T.; Tang, T.; Zhang, C.; Ji, Y.; Hämäläinen, M.; Liu, Y.: A web-based two-layered integration framework for smart devices. In: A web-based two-layered integration framework for smart devices, 2012.

[YW13] Yang, T.; Wu, Y.: What to Share and Why to Share? A Case Study of Cross-Boundary Information Sharing in Taiwan e-Government. Journal of Library & Information Studies. Jun 2013, Vol. 11 Issue 1, p25-53. 29p.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik – Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwick-lung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^2CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture
Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010 co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010 Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010) Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010 Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010 Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010 Service Science – Neue Perspektiven für die Informatik Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010 Service Science – Neue Perspektiven für die Informatik Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010 Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010 Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011 Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011 Workshopband (inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf
INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)
INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)
Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)
Automotive – Safety & Security 2012
Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)
Massendatenmanagement in der Agrar- und Ernährungswirtschaft
Erhebung - Verarbeitung - Nutzung
Referate der 33. GIL-Jahrestagung
20. – 21. Februar 2013, Potsdam

P-212 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2013
Proceedings of the 12th International Conference of the Biometrics Special Interest Group
04.–06. September 2013
Darmstadt, Germany

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)
Software Engineering 2013
Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013
13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.)
Software Engineering 2013
Workshopband
(inkl. Doktorandensymposium)
26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband
11. – 12. März 2013, Magdeburg

P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
6. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung
03.–04. Juni 2013, Erlangen

P-218 Andreas Breiter, Christoph Rensing (Hrsg.)
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
8. – 11. September 2013, Bremen

P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)
Informatik erweitert Horizonte
INFOS 2013
15. GI-Fachtagung Informatik und Schule
26. – 28. September 2013

P-220 Matthias Horbach (Hrsg.)
INFORMATIK 2013
Informatik angepasst an Mensch, Organisation und Umwelt
16. – 20. September 2013, Koblenz

P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.)
Electronic Government and Electronic Participation
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013
16. – 19. September 2013, Koblenz

P-222 Reinhard Jung, Manfred Reichert (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2013)
St. Gallen, Switzerland
September 5. – 6. 2013

P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2013
10. – 11. September 2013
Kloster Banz, Germany

P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)
Vorgehensmodelle 2013
Vorgehensmodelle – Anspruch und Wirklichkeit
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.
Lörrach, 2013

P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)
Modellierung 2014
19. – 21. März 2014, Wien

P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement
Referate der 34. GIL-Jahrestagung
24. – 25. Februar 2014, Bonn

P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs
Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland

P-228 Stefan Katzenbeisser, Volkmar Lotz,
Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des
Fachbereichs Sicherheit der
Gesellschaft für Informatik e.V. (GI)
19. – 21. März 2014, Wien

P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International
Conference of the Biometrics Special
Interest Group
10. – 12. September 2014 in
Darmstadt, Germany

P-231 Paul Müller, Bernhard Neumair,
Helmut Reiser, Gabi Dreo Rodosek
(Hrsg.)
7. DFN-Forum
Kommunikationstechnologien
16. – 17. Juni 2014
Fulda

P-232 E. Plödereder, L. Grunske, E. Schneider,
D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart

P-233 Stephan Trahasch, Rolf Plötzner, Gerhard
Schneider, Claudia Gayer, Daniel Sassiat,
Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg

P-234 Fernand Feltz, Bela Mutschler, Benoît
Otjacques (Eds.)
Enterprise Modelling and Information
Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014

P-235 Robert Giegerich,
Ralf Hofestädt,
Tim W. Nattkemper (Eds.)
German Conference on
Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de