

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 co-operation 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 978-3-88579-642-8

EMISA 2015 is the sixth international workshop in a series that provides a key forum for researchers and practitioners in the fields of enterprise modeling 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 2015 will provide an international forum to explore new avenues in enterprise modeling and the design of IS architectures by combining the contributions of different schools of Information Systems, Business Informatics, and Computer Science.



GI-Edition

Lecture Notes in Informatics

Jens Kolb, Henrik Leopold,
Jan Mendling (Eds.)

Enterprise Modelling and Information Systems Architectures

Proceedings of the 6th Int. Workshop on
Enterprise Modelling and Information
Systems Architectures, Innsbruck, Austria

September 3-4, 2015

J. Kolb, H. Leopold, J. Mendling (Eds.): EMISA 2015





Jens Kolb, Henrik Leopold, Jan Mendling (Eds.)

**Enterprise Modelling and
Information Systems Architectures**

**Proceedings of the 6th International Workshop on Enter-
prise Modelling and Information Systems Architectures**

**September 3-4, 2015
Innsbruck, Austria**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-248

ISBN 978-3-88579-642-8

ISSN 1617-5468

Volume Editors

Jens Kolb

Universität Ulm, 89069 Ulm, Germany

Email: jens.kolb@uni-ulm.de

Dr. Henrik Leopold

VU Amsterdam, 1081 HV Amsterdam, The Netherlands

Email: h.leopold@vu.nl

Prof. Dr. Jan Mendling

Wirtschaftsuniversität Wien, 1020 Wien, Austria

Email: jan.mendling@wu.ac.at

Series Editorial Board

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria

(Chairman, mayr@ifit.uni-klu.ac.at)

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Sigrid Schubert, Universität Siegen, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations

Steffen Hölldobler, Technische Universität Dresden, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2015

printed by Köllen Druck+Verlag GmbH, Bonn

Preface

The strategic importance of enterprise modelling has been recognized by an increasing number of companies and public agencies. Enterprise modelling delivers the ‘blueprints’ for co-designing and aligning business and enterprise information systems such that they complement each other in an optimal way. As example consider the support of business processes by process-aware information systems. Achieving such interplay requires a multi-perspective approach taking organizational, economic, and technical aspects into account. In a world of cloud, social and big data, additional challenges for enterprise modelling and the design of information systems architectures are introduced, e.g., in respect to the design of data-driven processes or processes enabling cross-enterprise collaboration. To deal with these challenges, a close cooperation of researchers from different disciplines such as Information Systems, Business Informatics, and Computer Science will be required.

EMISA 2015 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 2015 will provide an international forum to explore new avenues in enterprise modeling and the design of IS architectures by combining the contributions of different schools of Information Systems, Business Informatics, and Computer Science.

These proceedings feature a selection of high-quality contributions from academia and practice on enterprise modelling, enterprise architectures, business process modelling and process model matching. We received 14 submissions that were thoroughly reviewed by at least three selected experts of the program committee. Seven contributions were selected for presentation at the workshop and publication in these proceedings. For the first time, EMISA featured the Process Model Matching Contest. Twelve matchers took part in this competition.

We would like to thank the members of the program committee and the reviewers for their efforts in selecting the papers and helping us to compile a high-quality program. We would also like to thank the local organization, in particular Cornelia Haisjackl and Ilona Zaremba who served as local organization chairs at the University of Innsbruck. We also thank Agnes Koschmider for her keynote. We hope you will find the papers in this proceedings interesting and stimulating.

August 2015

Jens Kolb, Henrik Leopold, and Jan Mendling

Program Co-Chairs

Jens Kolb, Universität Ulm, Germany

Henrik Leopold, VU Amsterdam, The Netherlands

Jan Mendling, Wirtschaftsuniversität Wien, Austria

Program Committee

Antonia Albani, University of St. Gallen, Switzerland

Jörg Becker, European Research Center for Information Systems, Germany

Patrick Delfmann, European Research Center for Information Systems, Germany

Jörg Desel, FernUniversität in Hagen, Germany

Michael Fellmann, University of Rostock, Germany

Fernand Feltz, Centre de Recherche Public - Gabriel Lippmann, Luxembourg

Ulrich Frank, Universität of Duisburg Essen, Germany

Andreas Gadatsch, Hochschule Bonn-Rhein-Sieg, Germany

Wilhelm Hasselbring, Kiel University, Germany

Reinhard Jung, University of St. Gallen, Switzerland

Dimitris Karagiannis, University of Vienna, Austria

Stefan Klink, Karlsruhe Institute of Technology, Germany

Agnes Koschmider, Karlsruhe Institute of Technology, Germany

Horst Kremers, CODATA-Germany

Susanne Leist, University of Regensburg, Germany

Peter Loos, Saarland University, Germany

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria

Daniel Moldt, University of Hamburg, Germany

Bernd Mueller, Fachhochschule Braunschweig/Wolfenbüttel, Germany

Bela Mutschler, University of Applied Sciences Ravensburg-Weingarten, Germany

Markus Nüttgens, Universität Hamburg, Germany

Andreas Oberweis, Karlsruhe Institute of Technology, Germany

Sven Overhage, University of Bamberg, Germany

Hansjuergen Paul, Institut Arbeit und Technik, Germany

Henderik Proper, Public Research Centre Henri Tudor, Luxembourg

Manfred Reichert, University of Ulm, Germany

Ulrich Reimer, University of Applied Sciences St. Gallen, Switzerland

Stefanie Rinderle-Ma, University of Vienna, Austria

Peter Rittgen, University of Borås, Sweden

Andreas Schoknecht, Karlsruhe Institute of Technology, Germany

Stefan Strecker, Universität Duisburg Essen, Germany

Oliver Thomas, Universität Osnabrück, Germany

Klaus Turowski, Otto-von-Guericke-University Magdeburg, Germany

Gottfried Vossen, European Research Center for Information Systems, Germany

Barbara Weber, University of Innsbruck, Austria

Mathias Weske, University of Potsdam, Germany

Additional Reviewers

Uwe Lienert, FernUniversität in Hagen, Germany

Marcel Rosenberger, University of St. Gallen, Switzerland

Tim Niesen, DFKI, Germany

Dominik Bork, University of Vienna, Austria

Process Matching Contest Organizers

Henrik Leopold, VU Amsterdam, The Netherlands

Heiner Stuckenschmidt, University of Mannheim, Germany

Matthias Weidlich, HU Berlin, Germany

Christian Meilicke, University of Mannheim, Germany

Elena Kuss, University of Mannheim, Germany

Local Organization Chairs

Cornelia Haisjackl, University of Innsbruck, Austria

Ilona Zaremba, University of Innsbruck, Austria

Directory

Keynote

Agnes Koschmider

Quality of Process Element Labels – Where are we now, where should we go from here?13

Enterprise Modelling

Arian Storch, Ralf Laue, Volker Gruhn

Flexible Evaluation of Textual Labels in Conceptual Models17

**Agnes Koschmider, Timm Caporale, Michael Fellmann, Jonas Lehner,
Andreas Oberweis**

Business Process Modeling Support by Depictive and Descriptive Diagrams31

Michael Radloff, Martin Schultz, Markus Nüttgens

*Extending different Business Process Modeling Languages with Domain Specific
Concepts: The Case of Internal Controls in EPC and BPMN45*

Kathrin Figl, Mark Strembeck

Findings from an Experiment on Flow Direction of Business Process Models59

Information Systems Architecture

Felix Kossak, Verena Geist

An Enhanced Communication Concept for Business Processes.....77

Stefanie Rinderle-Ma, Zhendong Ma, Bernhard Madlmayr

*Using Content Analysis for Privacy Requirement Extraction and
Policy Formalization93*

Anne Baumgrass, Cristina Cabanillas, Claudio Di Ciccio

*A Conceptual Architecture for an Event-based Information Aggregation Engine
in Smart Logistics109*

Process Model Matching Contest

Goncalo Antunes, Marzieh Bakhshandeh, Jose Borbinha, Joao Cardoso, Sharam Dadashnia, Chiara Di Francescomarino, Mauro Dragoni, Peter Fettke, Avigdor Gal, Chiara Ghidini, Philip Hake, Abderrahmane Khiat, Christopher Klinkmüller, Elena Kuss, Henrik Leopold, Peter Loos, Christian Meilicke, Tim Niesen, Catia Pesquita, Timo Péus, Andreas Schoknecht, Eitam Sheetrit, Andreas Sonntag, Heiner Stuckenschmidt, Tom Thaler, Ingo Weber, Matthias Weidlich <i>The Process Model Matching Contest 2015</i>	127
---	-----

Keynote

Quality of Process Model Element Labels - Where are we now, where should we go from here?

Agnes Koschmider¹

Abstract: The redesign of business process models is up to now mainly limited to the improvement of their semantic quality. Conformance is checked between statements that are used in the model and the domain to be modeled. However, to ensure the semantic quality of a process model it is crucial to consider its intended purpose (e.g., as a communication foundation). Also the empirical and pragmatic quality, which improves readability and understandability, respectively, must be addressed. Awareness should be raised about the fact that the improvement of both quality dimensions is a critical success factor. In this talk, I will argue that the curriculum of BPM must be extended by teaching concepts and guidelines towards making process models readable and understandable. Also, the improvement of process model element labels in particular and process models in general must be tackled interdisciplinary. I will show that the improvement of both quality dimensions is a hard mathematical problem. An “optimal” design of process element labels and process models must therefore be considered as a trade-off between empirical and pragmatic quality.

Keywords: process model redesign, visualization, quality, semantics.

¹ Institute AIFB, Karlsruhe Institute of Technology, agnes.koschmider@kit.edu

Enterprise Modelling

Flexible Evaluation of Textual Labels in Conceptual Models

Arian Storch¹, Ralf Laue², Volker Gruhn³

Abstract:

This paper introduces a flexible and generic approach to define customised style rules for labels in conceptual models. A rule-based language is presented which can express style rules in a flexible and context-specific way. The formalised style rules are used to analyse and evaluate textual labels of model elements. To analyse a textual label, a combination of standardised natural language processing tools such as a part-of-speech tagger or a named entity recogniser are used. With the help of these techniques, custom-defined information entities can be extracted from the model.

Keywords: Conceptual Modelling, Natural Language Processing, Configuration, Customisation, Information Entity Extraction

1 Introduction

Conceptual models are typically used to document and communicate the architecture, environment and processes of an organisation [HWPZ03]. Once created, they can be analysed, discussed and gradually changed or improved to fit the constantly changing needs. Furthermore, they can be used as a source for designing and implementing software applications. In order to serve as a means for communication, the models have to be not only correct, but also easy to understand. Therefore, domain experts, companies and researchers have defined modelling guidelines with the aim to improve the quality and understandability of models [Si11].

[LSS94] elaborated three quality characteristics of conceptual models: *syntactic quality*, *semantic quality* and *pragmatic quality*. One aspect of *pragmatic quality* is *label quality*, and one measurable aspect of *label quality* is the adherence to naming conventions [SLG13]. Enforcing naming conventions can help to avoid misunderstandings. Mendling et al. [MRR10] give several examples of labels that can raise understanding problems when no naming convention is in use. For example, for “measure processing” it is unclear whether “to measure” or “to process” is the verb describing the action. Other authors emphasise the importance of text labels for the understanding and comparison of process models [MRR10, Be09b, NH15]. When models have to be compared (for purposes such as benchmarking, compliance analysis or model merging) it is necessary to know which nodes correspond to each other – even if a node is labeled “test software” in one model and

¹ it factum GmbH, Hainstr. 6, 04109 Leipzig, Germany, arian.storch@it-factum.de

² University of Applied Sciences of Zwickau, Department of Computer Science, Dr.-Friedrichs-Ring 2a, 08056 Zwickau, Germany, ralf.laue@fh-zwickau.de

³ Paluno - The Ruhr Institute for Software Technology, University of Duisburg-Essen, Gerlingstr. 16, 45127 Essen, Germany, volker.gruhn@paluno.uni-due.de

“software testing” in another one. The same is true for approaches such as [HD15] where process errors are found based on patterns which depend on the actions expressed by the element labels. For such purposes, it is necessary to extract information entities from a label. In the given example, one wants to know that “to test” describes the activity and “software” is the object.

Recommendations for naming conventions are typically grounded on empirical studies and related to a modelling language, but not to a business or organisation. However, in certain domains, there can be special demands for label styles which are not covered by existing naming conventions. For example, Combi et al. [Co12] suggest the modelling of medical processes with activity labels that can contain time information such as “Patient Evaluation [5,20] min”. Obviously, such a label would not fulfill any of the commonly suggested style rules. Therefore, tools which can check only whether an activity label follows some “standard” style, would recognise this label as *irregular*. Therefore, what is needed is an approach that allows organisations to define own naming conventions.

In this paper, we introduce a flexible and generic approach that allows users to define style rules for model element labels according to their specific needs. These style definitions are expressed using a proprietary part-of-speech (POS) pattern description language. This language also supports the extraction of specific semantic word groups of a label text (such as a business object from an activity label). Based on these formalised label style rules, we build up an algorithm to analyse and evaluate labels of model elements. We utilise a combination of natural language processing (NLP) tools such as a POS tagger or named entity recognition (NER) to determine the POS of words and validate them against the formalised label style rule. Making use of the fact that there are mature NLP tools for the English language, we implemented an algorithm for analysing English model element labels.

This paper is structured as follows. First, we describe the idea of part-of-speech tagging, the NLP technique on which our approach is grounded. Next, we analyse the related work about label analysis and discuss the research gap that resulted in the approach presented in this paper. In section 3 we explain our approach and discuss its advantages and shortcomings. Finally, we summarise the work in section 4 and motivate further research.

2 Background

2.1 Part-of-Speech Tagging

NLP refers to a theoretically motivated range of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing [Li01]. In the context of conceptual models, one field of application of NLP is to recognise the part-of-speech (POS) of words and phrases in a label of a model element. Based on this information, the word sequence can be decomposed into information entities, e.g. the activity (verb) or the affected subject (object).

The process of determining the POS of words and phrases is called POS tagging. POS taggers typically combine lexical databases with statistical algorithms to determine the kind of a word or phrase within a sentence [Me02]. One of the most powerful and popular POS taggers is the Stanford Tagger⁴ which is part of a toolset developed by the Stanford Natural Language Processing Group⁵. Its tagging methods are based on *statistical parsing*, where the frequency of grammar rules is exploited for finding the most probable POS of a word. These frequencies were obtained from a manually preprocessed (hand-parsed) collection of texts, called *text corpus*.

However, even though NLP tools provide reliable results when being applied to natural language texts [To03] they have to deal with difficult issues when processing labels in conceptual models. In most cases, such labels consist of one up to three words and do not fulfill a complete sentence structure [LSM12]. This makes it difficult to find the POS of a word, given the fact that in English derivation (such as from a noun to a verb) can occur without any change of form (this phenomenon is called *conversion* or *zero derivation*) [Di08]. For instance, the English word “log” can be a verb or a noun. Additionally, the accuracy of NLP tools decreases if the label contains special characters, for instance “Read/save/print notification [A5 page size]”.

2.2 Related work

Combining techniques of NLP tools with conceptional models has been applied to many fields. On the one hand, there are several approaches to create conceptional models from natural language text or vice versa. Montes et al. [Mo08] use a POS tagger and parser to automatically generate a conceptual model from the textual descriptions of use case scenarios. Other authors are using parsers and WordNet⁶ (a lexical database which provides information about semantic and lexical relations between words [Fe98]) to create BPMN Process Models, ER-Models or UML-Models from text [FMP11, GSD99, BC12]. Approaches to generate text from conceptual models are described in [Da92, MAA08, LRR96]. All these approaches have in common that they use a conceptual model either as source or as target of a transformation. On the other hand, there are algorithms for inspecting a single model. Some authors developed approaches to increase the understandability and consistency by reducing linguistic variations and enforcing naming conventions of model element labels. This is done by relying on WordNet or domain-specific terminology databases [vdVGvdR97, KHO11, Be09b]. Other authors are using labels for measuring the quality or similarity [EKO07, NH15] or for detecting semantic errors [GL11, HD15]. A detailed overview of existing approaches can be found in [Le13].

Reducing naming variations and enforcing style guidelines can help to avoid errors and misunderstandings. Therefore, modelling and label style guidelines have been developed. Algorithm and tools can be designed to determine and enforce the compliance to such style rules. In the case of business process models, Leopold et al. [LSM11] introduced

⁴<http://nlp.stanford.edu/software/tagger.shtml>

⁵<http://www-nlp.stanford.edu/>

⁶<http://wordnet.princeton.edu/>

an algorithm to recognise the style of labels for describing activities. Assuming that there are seven labeling styles used in business process models [LSM12], they designed an algorithm to recognise the label style by comparing words, their order within the word sequence and their POS to grammatical phrase structures that have been derived from the style rules. The POS determination is done with the help of WordNet and the Stanford Tagger. If a label can not be recognised clearly, it is examined whether this word can be found and assigned to its POS in other labels of the same model or in other models in a repository. However, Leopold et al. do not recommend the application of the Stanford Tagger (cf. [LSM12, LSM09]) because labels often do not meet the requirement of proper sentences. Anyhow, they observed a considerable increase of accuracy when using complements to extend the label text to a full sentence [LSM09]. It has to be noted that in the tools described [LSM12, LSM09, LSM11] the allowed style guidelines are hard-coded in the software. This does not allow easy customisation to specific needs or adaption to new modelling languages whose model elements express other concepts.

In [Le13], Leopold et al. describe impressive results for validating labels in business process models. They used linguistic patterns built from sequences of POS to operationalize style rules. A desirable style for activity labels of business process models (called *verb-object* style) was defined as “*Verb(Imperative) + Noun [+ Preposition + Noun]*” (square brackets denote optional elements). When analysing whether activities in the SAP reference model adhere to this style, they achieved an F-measure (the harmonic mean of precision and recall) of 96.7%. The algorithm presented in [Le13] relies on manually tagged corpora and can therefore be used even if no other NLP tools for a language exist. It first determines the POS of each word and then checks whether this sequence of POS corresponds to the defined style. As an example, the label “Provide service” would be classified as correct (a verb followed by a noun), but “Project planning” won’t. Additionally, Leopold et al. [Le13] exploit information about labels of other model elements and from other models in a model repository. On the downside, such information is not available when a new model is created and no model repository exists so far.

Becker et al. [Be09a] introduced an approach to define naming conventions based on a linguistic grammar which describes phrase structures. This approach is applicable for any modelling language. With the help of the grammar, particular style rules can be expressed. As an example, the expression $\langle \textit{verb, imperative} \rangle \langle \textit{noun, singular} \rangle$ can be used to define a label style for a process activity. The POS determination and expression matching is realised by utilising a domain-specific terminology database and word relations (for instance, synonyms) queried from WordNet. By restricting the set of available words by the terminology database, there is no need to use additional NLP tools. If a word and its POS can not be analysed automatically because of unknown words or missing relations, the modeller has to interact with the modelling system to resolve the problem [Be09b].

Flexible style rules have been used to define linguistic patterns for describing natural language requirements by de Almeida Ferreira and da Silva [dd13]. However, their approach for checking the style cannot directly be transferred to conceptual models, because de Almeida Ferreira and da Silva make use of a deterministic set of keywords frequently occurring in requirements (for instance, “x IS y” or “x HAS y”).

Object	POS tags (see Tab. 2)
Cheque	NN
Salary Cheque	NN NN
Valid cheque	JJ NN
A complete list of overdue salary cheques	DT JJ NN IN JJ NN NNS

Tab. 1: Various Word Sequences Forming an Object

2.3 Motivation

The related work discussed in section 2.2 shows that there is quite some research in the field of analysing labels of conceptual models. However, we still observe a notable gap when adapting common style guidelines to concrete domains, for example in order to allow labels such as “Patient Evaluation [5,20] min” mentioned in Sect. 1.

By studying labels of different conceptual model collections we observed that even without such specific style guidelines, the pattern “*Verb(Imperative) + Noun [+ Preposition + Noun]*” used in [Le13] for describing an activity would be far too strict. As an example, the label “Choose non-available item” would be classified as being wrong because adjectives are not permitted. But in our opinion, this label should be regarded as having the *verb-object* style as well. Moreover, we noted that from a linguistic point of view an object can consist of a wide range of words with different POS. For instance, the label “print list of overdue cheques” contains the object “list of overdue cheques” which is built from multiple nouns, a preposition and an adjective. Some kinds of objects and their POS are shown in Tab. 1. The tags are explained in Tab. 2. Considering the variety of conceptual models and domain specific needs, we believe that there is a need of a flexible linguistic pattern expression language.

As already mentioned by other researches, one main problem of POS determination is the fact that words sharing the same form belong to different word classes. For instance, the word “test” can be a noun or a verb. To obtain a correct decision, more context information is needed. One type of context can be the surrounding words. Using a word sequence such as “test the software” allows classifying the word correctly. Another type of context is the type of the model element carrying the label. For instance, the label “Review” gets a different understanding when being used in an activity or a resource model element respectively.

Becker et al. [Be09a] avoid the problem of words sharing the same form by requiring that all words used in a label must be taken from a set of words defined by a domain-specific terminology database. We acknowledge the advantages of using such a terminology database. However, in many situations maintaining such a database will not be feasible.

In order to offer the possibility for checking labels styles in as many situations as possible, it was our aim to create an approach that...

- allows to define the style rules in a flexible manner,

- does not restrict the vocabulary by requiring that all words belong to a terminology database, and
- does not require the presence of a model repository.

3 Customised Pattern-Based Label Evaluation

In this section, we present our approach to increase the reliability and flexibility of a textual label analysis. First, we introduce a pattern description language that is used to define the expected grammatical phrase structure and POS of word sequences in a model element label. Then, we describe the three stages of our algorithm. Figure 2 illustrates these stages; the steps shown in this figure will be explained in the following subsections.

3.1 Pattern Expression Language

Tag	Part-of-Speech
CC	coordinating conjunction
DT	Determiner
EX	Existential there
IN	preposition
JJ	Adjective
NN	Noun
VB	Verb
VRN	Verb, past participle
VBZ	Verb, 3rd person singular present

Tab. 2: Short Extract of the Penn Treebank Tag Set

In our previous work [SLG14], we gave examples how patterns for label styles can be expressed using a combination of tags from the Penn Treebank Tag Set (PTTS) and the Extended Backus-Naur Form (EBNF, ISO standard 14977). The PTTS defines a standardised set of tags denoting the POS of a word that has been processed by a POS tagger [Sa90]. Table 2 shows a subset of the tags defined in the PTTS. With the help of this notation, we defined style rules for the goal-oriented modelling language i^* . In subsequent work, some elements of the style rule specification language have been redefined in order to improve the readability, modularity and flexibility.

Listing 1 and 2 show the style rule specifications for a task and a goal in the language i^* . For example, the rules for task labels define that the text must start with a verb in base form and may be followed by a conjunction and another verb in base form⁷. Then, there must be a word sequence optionally headed by a preposition that matches the object rule. Optionally, an additional conjunction followed by an object may complete the label text.

⁷One might argue that it is not a good idea to allow two verbs in a label [BK04], but our point is that such a rule should not be generally defined but decided by the organisation according to their specific needs.

For example, a task label could be: “**debit** (VB) **the credit card** (object, DT NN NN)” while a goal label according to Listing 2 could look like: “**transfer** (object, NN) **is completed** (VBN)”.

```
task {
  nnseq          = (NN|NNS)+;
  attrNoun       = DT? JJ* VBN? nnseq;
  object         = attrNoun ((IN|TO) attrNoun)?;
  verbImp        = VB (RP|IN)?;
  task           = verbImp (CC verbImp)? (IN|TO)? object
                  (CC object)?;
  []             = "You have to $. ";
}
```

List. 1: Task Style Rule

```
goal {
  nnseq          = (NN|NNS)+;
  attrNoun       = DT? JJ* VBN? nnseq;
  object         = attrNoun ((IN|TO) attrNoun)?;
  goal           = object (CC object)? ("is"|"are") VBN;
  []             = "The $. ";
}
```

List. 2: Goal Style Rule

A rule definition starts by naming the process model element to which the rule shall be applied to. Then, within the curling brackets, sub-rules are defined. Though, different rules can share the same sub-rules without re-defining them, the declaration here has been made redundant to clarify the intention. Table 3 lists the meaning of the special characters.

+	At least one occurrence
*	Zero or more occurrences
?	Optional occurrence
(...)	Groups occurrences or expressions
“...”	Exact match
... ...	Alternative match; At least one expression must match (within a group)
\$	Origin label text

Tab. 3: Overview of the Special Characters to express Operators

As already mentioned, POS taggers can produce more reliable results when operating on full sentences. Therefore, our style rules define a prefix so that the label text can be completed to full sentences (in the expression after the “[]” symbol). The special character “\$” defines the location to insert the original label text. For instance, the activity label “Complete first test” will be complemented to “You have to complete first test.”. A text matches the style rule if and only if it becomes a full sentence when completed using the given complement.

When analysing real-world model element labels, we realised that the use of some words or phrases can lead to problems. For example, in an activity label “perform investigation”, it would be wrong to conclude that “perform” is the verb and “investigation” is the business object. Instead, verbs such as “perform”, “execute” etc. should be avoided, and the text should be “investigate [object]” instead. Similarly, in a model in the language i^* , we do not want to have labels such as “achieve [something]” (which would rather be a goal than an activity); or in VDMML models (describing the process of value delivering), an activity label should not be “provide [something]” (which would rather be a value proposition element). For this reason, our rule definitions allow to exclude certain words or word groups, using constructs such as

```
verb = VB;  
verb != "perform" | "execute";
```

In this way, our language can use both a whitelist (listing all words or word groups that are allowed at a certain position) and a blacklist (listing all words that are not allowed, despite the fact that their use would be correct from a grammatical point of view).

There are further notation elements to express prefix and postfix of words, words surrounded by special characters, etc. Due to space limits, we won't give a full overview here.

For applying the style rules, we transform them into regular expressions. This allows us to use standard library functions for checking whether the output of a POS tagger (and hence the label) conforms to the style rule. For example, the rules of Listing 3 are transformed into the regular expression `^(DT[/A-Z]*)?(JJ[/A-Z]*)?(NN[/A-Z]*)+$`.

```
nnseq      = NN+;
object     = DT? JJ? nnseq;
```

List. 3: Object style rule excerpt

All existing style rules will be aggregated and analysed. For each model element type, the result will be written in a label metadata repository. Figure 1 illustrates this process.

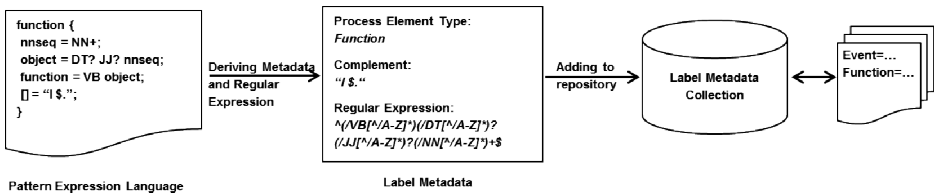


Fig. 1: Creating Label Metadata

3.2 Evaluation of Phrase Structure and POS of a Model Element Label Text

This section outlines our approach to evaluate the phrase structure and POS of model element labels. The designed algorithm consists of three stages as illustrated in Figure 2.

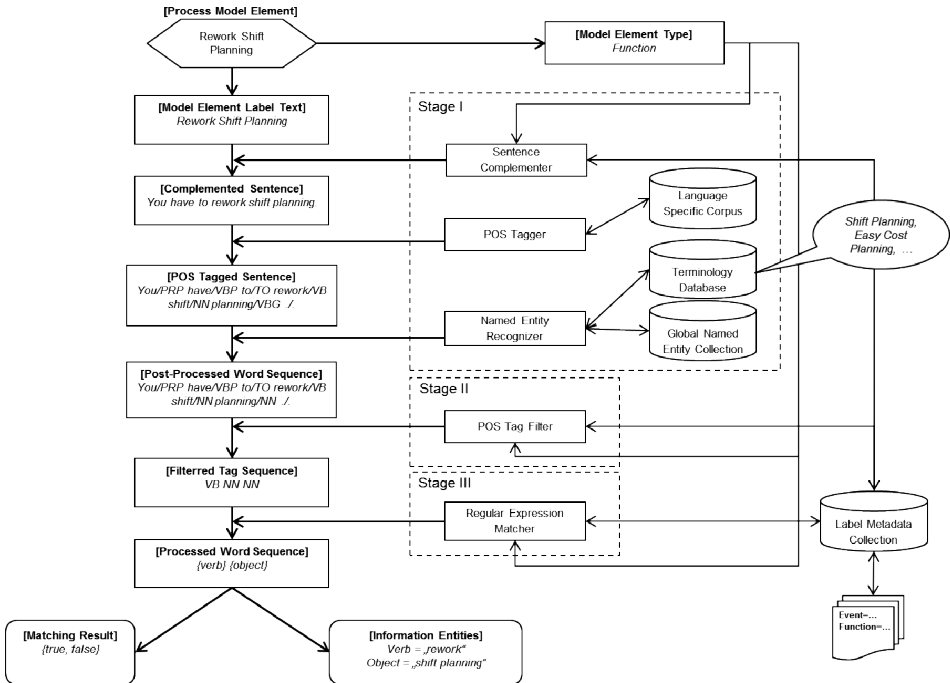


Fig. 2: Process Stage Model

Stage I: Sentence Completion / POS Determination / Named Entity-Recognition (optional) When analysing a label, we start by building a sentence using the sentence complement defined for the model element type. Then, the POS tagger is applied to determine the POS of each word of the text. Afterwards, a *named entity recogniser* (NER) can be used to improve the sequence of POS tags. It looks up words and word groups in a domain-specific terminology database which might be created by domain experts beforehand and re-tags it if necessary. Such a terminology database may also include domain-specific abbreviations. Additionally, the recogniser uses a collection of globally known named entities to regard non-domain-specific named entities (e.g., names of locations, organisations, titles of laws, etc.), too.

Stage II: POS Tag Filtering After each word has been annotated with its most likely POS, we have to apply a *tag filter*. It removes irrelevant POS tags for those words which have been added as a complement to the original label in stage I. At the end of stage II we have a tag sequence that represents the POS tags of the label.

Stage III: Regular Expression Matcher A *regular expression matcher* is finally used to compare the actual POS tag sequence with the expected one. Only if the sequence matches the regular expression, we conclude the label to be valid. In this case, we can additionally

extract the phrases associated with a matching sub-rule, for example we can conclude that in “approve all valid insurance claims” the object is denoted by the phrase “all valid insurance claims”.

3.3 Analysis Results

To validate our approach at an early development stage, we analysed two collections of labels taken from different model collections. The first is a collection of 100 task labels taken from an *i** model repository. The second is a collection of 100 EPC function labels taken from the SAP reference model. We analysed these collections twice. The first time, we used a derived definition of the *verb-object* style as given in [Le13]: “*Verb(Imperative) + Noun [+ Preposition + Noun]*”. Its formalisation in our rule language is shown in List. 4. Though Leopold et al. mentioned composite nouns (e.g., service order), it is not fully described how they recognise them. Therefore, we decided to use a pattern which conforms to the given formal definition. The second time, we used the definition according to List. 1. Following these definitions, we manually classified each label within these collections. We then compared the manual results with those of our algorithm. Tab. 4 gives an overview of the collections and the running time of our algorithm on a Lenovo X1 Carbon 2014 with a 1.50 GHz Intel Core i7-4550U processor, 8 GB RAM and a SSD device, running on Windows 7 and a JVM 1.8. The initialisation time of the Stanford POS Tagger and WordNet has not been measured.

```
activity {
  noun      = (NN|NNS);
  prep      = (IN|TO);
  activity  = VB noun (prep noun)?;
}
```

List. 4: Strict Verb-object Style according to Leopold et al.

As shown in Tab. 4, both collections have very different numbers of labels written in *verb-object* style. Due to the strict definition of the style rule following [Le13], many labels do not match. E.g., “Get relevant items” does not match because it contains an adjective. By contrast, the more tolerant style rule according to List. 1 allows amongst others the existence of adjectives before a noun. Therefore, the number of labels adhering to this style rule is much higher.

As part of our evaluation, we run our algorithm four times for each style rule definition. We used the Stanford Tagger as POS tagger. For correctly classifying named entities and business terms, we made use of the glossary in the terminology base sapterm.com which lists various common business terms. Additionally, we utilised WordNet to correct POS tags of words that may have been tagged wrongly by the Stanford Tagger (due to the used complement) but are actually unambiguous.

We started without a prefix or NER. Then, we successively added each feature, using “You have to \$.” as prefix. The results of each evaluation are shown in Tab. 5 and 6.

Model collection	i^*	SAP
	100 i^* Tasks	100 EPC Functions
Average no. of words per label	2.98	3.23
Minimum no. of words per label	1	1
Maximum no. of words per label	7	9
No. of labels in verb-object style		
As defined in Listing 4	38	5
As defined in Listing 1	80	28
Performance results applying Listing 1		
Avg. running time per label (ms)	1.38	2.02
Max. running time per label (ms)	2.0	52.0

Tab. 4: Model Collections Details and Performance Results

Model Collection	i^*			SAP		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
No prefix, no NER	100.0 %	63.2 %	77.4 %	- ⁸	0.0 %	-
No prefix, NER	100.0 %	71.1 %	83.1 %	- ⁸	0.0 %	-
Prefix, no NER	100.0 %	86.8 %	93.0 %	33.3 %	100.0 %	50.0 %
Prefix, NER	100.0 %	97.4 %	98.7 %	83.3 %	100.0 %	90.9 %

Tab. 5: Evaluation Results of Verb-Object Style according to List. 4

Model Collection	i^*			SAP		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
No prefix, no NER	100.0 %	67.5 %	80.6 %	100.0 %	28.6 %	44.4 %
No prefix, NER	100.0 %	73.8 %	84.9 %	100.0 %	32.1 %	48.6 %
Prefix, no NER	100.0 %	90.0 %	94.7 %	65.8 %	89.3 %	75.8 %
Prefix, NER	100.0 %	98.8 %	99.4 %	100.0 %	96.4 %	98.2 %

Tab. 6: Evaluation Results of Verb-Object Style according to List. 1

We assessed the accuracy of our algorithm using the metrics precision, recall and F-measure (harmonic mean of precision and recall). According to the results, the combination of using a prefix and NER gains best results. In this case, the lowest F-measure is 90.9 % when applying the strict *verb-object* style rule to the SAP label collection. The best result is gained when applying the more tolerant *verb-object* style rule to the i^* label collection.

We identified two issues that may significantly reduce the accuracy of the analysis. First, if a word is misspelled, it is not possible to correctly determine its POS by the POS tagger because it can not be found in the corpus. For instance, if a word sequence is given as “check payment” instead of the correct form “check payment”, the resulting POS tag sequence is “VB VBD” instead of the correct one “VB NN”. We can observe another aspect of misspelling when the change of just one character leads to another POS or even word

⁸Neither True nor False Positives have been measured

meaning. E.g., “write advice” gets the tag result “*VB NN*” while “write advise” is tagged as “*NNP VBP*”.

Another issue is the text corpus that has been used to build/train the POS tagger. POS tagging works such that a POS with a high occurrence rate within the text corpus is preferred. But this occurrence rate is mainly influenced by the manually annotated texts in the corpus. Therefore, different corpora may lead to different tagging results. In general, we assume that corpora built on texts out of business domains will lead to more reliable results.

4 Conclusion and Future Work

In this paper, we present an approach to define highly customisable style rules for model element labels. These rules are expressed using a pattern-based language. It allows domain experts or modellers to define style rules according to their specific needs. We believe that our pattern expression language is very flexible, well to read and to understand. Therefore, it may be helpful in the context of end-user-programming.

One main benefit of our approach is that it can be used without referring to a model repository or a lexical database. Furthermore, it is fully automated and needs no user input once the style rules have been defined. Complements added to the labels provide the possibility to build proper sentences that can be processed by the POS tagger. In addition, the accuracy can be improved by using NER and a terminology database.

By applying standard NLP tools to the label, the POS of each word can be determined, and it can be checked whether a label conforms to a style rule. With the help of custom definable sub-rules, information entities can be extracted from analysed labels. This offers the possibility to transform a model (e.g., into another conceptual model type or natural text) or to analyse the label parts in another way.

The accuracy of this approach is mainly influenced by the spelling of the textual labels. Therefore, we plan to add additional preprocessing stages to our algorithm in order to deal with misspelled words. In addition, we plan to add the possibility to transform a label from a given custom style into another style to support the maintenance of model repositories. This will be important for the purpose of merging models that have been created in different organisations.

References

- [BC12] Bajwa, Imran Sarwar; Choudhary, M Abbas: From natural language software specifications to UML class models. In: Enterprise Information Systems, pp. 224–237. Springer, 2012.
- [Be09a] Becker, Jörg; Delfmann, Patrick; Herwig, Sebastian; Lis, Łukasz; Stein, Armin: Formalizing linguistic conventions for conceptual models. In: Conceptual Modeling-ER 2009, pp. 70–83. Springer, 2009.

- [Be09b] Becker, Jörg; Delfmann, Patrick; Herwig, Sebastian; Lis, Lukasz; Stein, Armin: Towards increased comparability of conceptual models-enforcing naming conventions through domain thesauri and linguistic grammars. pp. 2231–2242, 2009.
- [BK04] Berry, Daniel M.; Kamsties, Erik: Ambiguity in Requirements Specification. In: Perspectives on Software Requirements, volume 753 of The Springer International Series in Engineering and Computer Science, pp. 7–44. Springer, 2004.
- [Co12] Combi, Carlo; Gambini, Mauro; Migliorini, Sara; Posenato, Roberto: Modelling temporal, data-centric medical processes. In: ACM International Health Informatics Symposium. pp. 141–150, 2012.
- [Da92] Dalianis, Hercules: A method for validating a conceptual model by natural language discourse generation. In: Advanced Information Systems Engineering. Springer, pp. 425–444, 1992.
- [dd13] de Almeida Ferreira, David; da Silva, Alberto Rodrigues: RSL-PL: A linguistic pattern language for documenting software requirements. In: Third IEEE International Workshop on Requirements Patterns. pp. 17–24, 2013.
- [Di08] Dixon, Robert MW: Deriving verbs in English. *Language Sciences*, 30(1):31–52, 2008.
- [EKO07] Ehrig, Marc; Koschmider, Agnes; Oberweis, Andreas: Measuring similarity between semantic business process models. In: Proceedings of the fourth Asia-Pacific conference on Conceptual modelling-Volume 67. pp. 71–80, 2007.
- [Fe98] Fellbaum, Christiane, ed. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [FMP11] Friedrich, Fabian; Mendling, Jan; Puhlmann, Frank: Process model generation from natural language text. In: Advanced Information Systems Engineering. Springer, pp. 482–496, 2011.
- [GL11] Gruhn, Volker; Laue, Ralf: Detecting Common Errors in Event-Driven Process Chains by Label Analysis. *Enterprise Modelling and Information Systems Architectures*, 6(1):3–15, 2011.
- [GSD99] Gomez, Fernando; Segami, Carlos; Delaune, Carl: A system for the semiautomatic generation of ER models from natural language specifications. *Data & Knowledge Engineering*, 29(1):57–81, 1999.
- [HD15] Höhenberger, Steffen; Delfmann, Patrick: Supporting Business Process Improvement through Business Process Weakness Pattern Collections. In: 12. Internationale Tagung Wirtschaftsinformatik. pp. 378–392, 2015.
- [HWPZ03] Heemskerk, Marieke; Wilson, Karen; Pavao-Zuckerman, Mitchell: Conceptual models as tools for communication across disciplines. *Conservation Ecology*, 7(3):8, 2003.
- [KHO11] Koschmider, Agnes; Hornung, Thomas; Oberweis, Andreas: Recommendation-based editor for business process modeling. *Data & Knowledge Engineering*, 70(6):483–503, 2011.
- [Le13] Leopold, Henrik; Eid-Sabbagh, Rami-Habib; Mendling, Jan; Azevedo, Leonardo Guerreiro; Baião, Fernanda Araujo: Detection of naming convention violations in process models for different languages. *Decision Support Systems*, 56:310–325, 2013.

- [Li01] Liddy, Elizabeth D: Natural language processing. 2001.
- [LRR96] Lavoie, Benoit; Rambow, Owen; Reiter, Ehud: The modelexplainer. In: Proceedings of the 8th international workshop on natural language generation. pp. 9–12, 1996.
- [LSM09] Leopold, Henrik; Smirnov, Sergey; Mendling, Jan: On labeling quality in business process models. Nüttgens M et al.(Hrsg), 8:42–57, 2009.
- [LSM11] Leopold, Henrik; Smirnov, Sergey; Mendling, Jan: Recognising Activity Labeling Styles in Business Process Models. Enterprise Modelling and Information Systems Architectures, 6(1):16–29, 2011.
- [LSM12] Leopold, Henrik; Smirnov, Sergey; Mendling, Jan: On the refactoring of activity labels in business process models. Information Systems, 37(5):443–459, 2012.
- [LSS94] Lindland, Odd Ivar; Sindre, Guttorm; Solvberg, Arne: Understanding quality in conceptual modeling. Software, IEEE, 11(2):42–49, 1994.
- [MAA08] Meziane, Farid; Athanasakis, Nikos; Ananiadou, Sophia: Generating Natural Language specifications from UML class diagrams. Requirements Engineering, 13(1):1–18, 2008.
- [Me02] Megyesi, Beáta: Shallow parsing with PoS taggers and linguistic features. The Journal of Machine Learning Research, 2:639–668, 2002.
- [Mo08] Montes, Azucena; Pacheco, Hasdai; Estrada, Hugo; Pastor, Oscar: Conceptual model generation from requirements model: A natural language processing approach. In: Natural Language and Information Systems, pp. 325–326. Springer, 2008.
- [MRR10] Mendling, Jan; Reijers, Hajo A; Recker, Jan: Activity labeling in process modeling: Empirical insights and recommendations. Information Systems, 35(4):467–482, 2010.
- [NH15] Niesen, Tim; Houy, Constantin: Zur Nutzung von Techniken der Natürlichen Sprachverarbeitung für die Bestimmung von Prozessmodellähnlichkeiten–Review und Konzeptentwicklung. In: 12. Internationale Tagung Wirtschaftsinformatik, WI 2015, Osnabrück, Germany, March 4-6, 2015. pp. 913–924, 2015.
- [Sa90] Santorini, Beatrice: Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision). 1990.
- [Si11] Silver, Bruce: BPMN method and style. Cody-Cassidy Press, US, 2nd ed. edition, 2011.
- [SLG13] Storch, Arian; Laue, Ralf; Gruhn, Volker: Measuring and visualising the quality of models. In: IEEE International Workshop on Communicating Business Process and Software Models Quality. IEEE, pp. 1–8, 2013.
- [SLG14] Storch, Arian; Laue, Ralf; Gruhn, Volker: Analysing the Style of Textual Labels in i* Models. In: International i* Workshop co-located with the 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014). 2014.
- [To03] Toutanova, Kristina; Klein, Dan; Manning, Christopher D; Singer, Yoram: Feature-rich part-of-speech tagging with a cyclic dependency network. In: 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. pp. 173–180, 2003.
- [vdVGvdR97] van der Vos, Bram; Gulla, Jon Atle; van de Riet, Reind: Verification of conceptual models based on linguistic knowledge. Data & knowledge engineering, 21(2):147–163, 1997.

Business Process Modeling Support by Depictive and Descriptive Diagrams

Agnes Koschmider¹, Timm Caporale¹, Michael Fellmann², Jonas Lehner¹,
Andreas Oberweis¹

Abstract: The design of a “good” business process model is a time-consuming and error-prone task and requests high training effort from the process modeler. These barriers might be a reason why business processes are often designed with software tools, which were not intentionally developed for this purpose, but are highly familiar for the process modeler (e.g., add-ins for MS Office family) and thus a process model can be quickly designed. As consequence of such a tool choice for process modeling the variety of techniques available for Business Process Management cannot be exploited. To mitigate this situation, we first examine approaches aiming to support business process modeling more intuitively. We then suggest the introduction of an additional layer to business process models with depictive diagrams that are not bounded to a concrete process modeling language or descriptive diagrams using natural language text. We then show how such a layer can be aligned with common process modeling languages and thus provides a seamless integration with more advanced Business Process Management languages and tools. We expect that our approach will fertilize techniques facilitating business process modeling for all types of process modelers including business experts with limited experience of process modeling.

Keywords: Process Modeling, Business Process Model, Natural Language Processing, Visual Variables

1 Introduction

In the literature it is known that unexperienced modelers (e.g. novice modelers or business experts with little training) do not share the same expertise as professional modelers (e.g., business analysts) in terms of applying modeling guidelines and the correct use of the modeling language [KW10]. In more detail, unexperienced modelers tend to forget model elements according to a study from Nielen et al.: “*Concerning error frequencies, activity omissions were considerably higher for novices than for experienced modelers*” [NKM11]. Moreover, according to another study from Wilmont et al., unexperienced modelers have problems in finding the right level of details [WBv10]. From these empirical findings it can be concluded that applying a fully-fledged process modeling language supported by a sophisticated tool is too challenging for unexperienced modelers such as business or domain experts. Although they might have a profound knowledge of the domain that is to be modeled, modeling itself presents

¹ Institute of Applied Informatics and Formal Description Methods, KIT, Karlsruhe, Germany,
<firstname.lastname@kit.edu>

² Business Information Systems, University of Rostock, Rostock, Germany, michael.fellmann@uni-rostock.de

a barrier for them. It is thus important to reduce this barrier by providing alternative ways of participation while at the same time retaining the richness of fully-fledged process modeling languages for experts. This is still an open issue despite a large body of work suggesting assistance functions for business process modeling [FZM15]. Such approaches surely might help to decrease the effort of process modeling. These assistance tools suggest (similar to an auto-completion function) suitable fragments to complete a currently edited business process model. Definitely, such assistance functions increase the process model quality [KHO11]. However, the assistance is not process modeling agnostic. This means that process modelers still have to be familiar with the process modeling language and technique in order to fully exploit the modeling assistance.

In our research, we hence try to lower the entry barrier to process modeling in a different way. We suggest a lightweight approach to modeling via an on top layer to process models. This layer contains abstract models (“Layer 0”) that can be represented both depictive (iconic) or descriptive (symbolic) with the possibility to seamlessly switch between them. This layer should enable a quick and comprehensive view of the underlying process model and in addition should expose basic modelling capabilities. With this layer, we aim to make modelling accessible for a larger audience.

To identify relevant influence factors for the design of such a layer, first, a solid revision of related disciplines emphasizing different modalities of visualizing diagrams is required. This revision is presented Section 2. This section also discusses the range of variables in order to appropriately visualize the diagrams. Based on this discussion, Section 3 suggests two approaches for a graphical and textual visualization. Related approaches are compared in Section 4. The paper ends with a summary in Section 5.

2 Variables to Design Abstract Models

Generally, information can be presented either descriptive or depictive [SB03]. Depictive is related to an *iconic* representation of information where, for instance, graphics are used to describe the context. Descriptive is related to a symbolic representation of information, where natural language text describes the information. While some process model readers prefer textual information, others prefer two-dimensional representations such as graphics [Mo09]. Both modalities are processed differently, which means that different concepts are required for depictive and descriptive representations. According to the dual channel theory [MM03], visual representations are processed in parallel by the human visual system, while textual representations are processed serially by the auditory system [Be83].

When examining the strength of depictive representations, the argumentation of [Ai06] stands out that depictive representation “*can more easily express abstract information and more general negations and disjunctions*”. Another strength was observed by [SO95] who argues that “*text permits expression of ambiguity in a way that graphics*

cannot easily accommodate. It is this lack of expressiveness that makes diagrams more effective for solving determinate problems". It seems to be an agreement that "visual displays are often said to enhance or "augment" cognition" [He11].

Despite these strengths of depictive diagrams, descriptive representations are also advantageous. Particularly, descriptive (textual) diagrams have the advantage that no prior training effort is required in order to understand and use the symbols (letters, words). Furthermore, in new directions (e.g., mobile process modeling) textual diagrams, which suit to be created on the go, can be used as an intermediate from which the graphical diagram are generated automatically [Ke14].

Obviously, information can be presented in both modalities combining graphics with textual description. Presenting information in multiple modalities is regarded as being useful to learners who actively process such information [Ai06]. On the other hand, Kalyuga [Ka11] observed that the human working memory is very limited when handling new information because initially no mechanism is available that coordinates novel information. Due to the restricted capability of the working memory Kalyuga advocates to separate channels for dealing with auditory (descriptive) and visual material (depictive).

To sum up the discussion, both modalities of representation have their strengths. To exploit them, the abstract layer suggested in this paper, offers both modalities in order to support particularly business experts with limited experience of process modeling. The diagrams created on this abstract layer can be considered as abstraction of the underlying process models, meaning that fine-grained concepts are abstracted to related concepts on a higher level. Both representations are aligned in order to allow a seamlessly switch between them and they allow a navigation to their subsequent layers in order to support a seamless integration with techniques and tools of Business Process Management. Fig. 1 shows the placement of the new layer. We call it "Layer 0" since it precedes the current starting point of process modeling on e.g., "Layer 1".

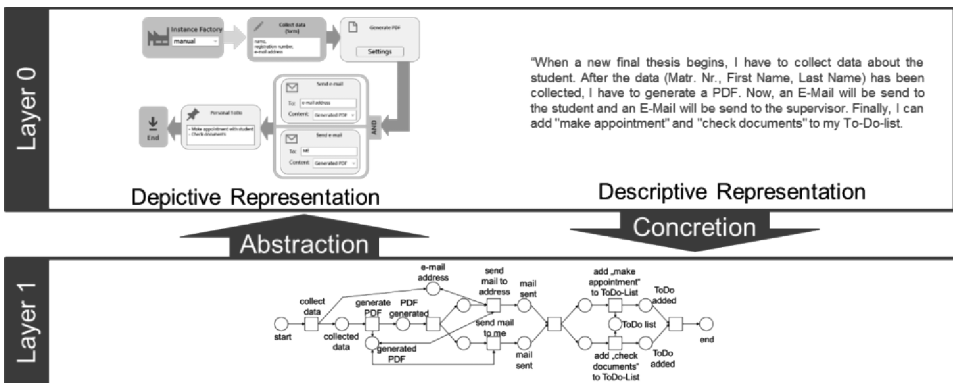


Fig. 1: Two modalities of the abstraction layer ("Layer 0")

The new layer, which allows “textual abstraction” and “graphical abstraction”, has several advantages:

- From the viewer’s perspective: Both representations should allow the process model’s viewer to get a quick and comprehensive view of the underlying process model. If the viewer is further interested in the fine-grained view of the process model he/she can navigate through the process model hierarchy.
- From the creator’s (i.e., process modeler’s) perspective: the concepts of this layer abstract from common process modeling languages, and thus, we expect, that the creation of process models even for inexperienced persons is easier.

In the following we discuss variables how to best design both modalities of representation.

2.1 Designing Depictive Diagrams

The design of a depictive diagram can be described based on the visual variables by J. Bertin [Be83]. These visual variables have been applied to process models by [Ko15] and are summarized in this section. Visual (or graphical) presentation is categorized in planar variables (addresses the X, Y location) and retinal variables (shape, size, color, brightness, orientation, texture). Some of these variables are detected in parallel (color and texture). Shape, for instance, is detected in a less efficient scanning [TS86]. Thus, scanning of shape is affected when combining it with color. This means that business process models have to be designed in a way that users can recognize the fundamental elements of the model with minimal cognitive effort. Each of these variables can be used singularly or in combination.

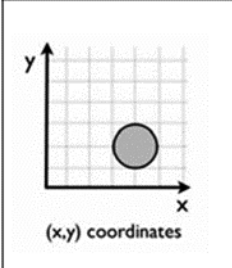




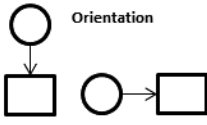

Planar Variables	Retinal Variables		
	Shape	Size	Colour
			
	Brightness	Orientation	Texture
			

Fig. 2: Visual variables

Shape. Different information can be expressed by different shapes. A varying number of visual variables of graphical elements makes the elements easier to identify [Mo09]. To avoid confusion, it is recommended to use common (prominent) and particularly distinct

geometrical shapes (circle, triangle, square, diamond). Following such recommendations of symbol choice increases memory of visual aspects [Fi12]. Shapes are appropriate for the representation of information and the shape choice should be well considered.

Size. Size of a process element must be in relationship to the total number of elements of the process model, the length of the element label and the space of the modeling workbench. In this context it should be considered that [KRD12] found out, when using the preferred style of granularity (flattened process models with no refinements versus modularly built process models) then no negative effects were identified on the performance in making sense of such a process model. This means that “large” process models might be understandable if such a process model corresponds to the preferred style of granularity of the user.

Color. Color is a powerful and effective visual variable because it is detected in parallel [TS86]. Differences in color are perceived faster than differences in shape. Generally, color facilitates information processing [Lo93], when being used effectively. Too many different colors however can impair communication [LGH14] and do not act as effective cognitive aids in problem solving. When using color as visual variable in order to represent the context on layer 0, it might be rational to limit the number of colors to the use of six colors for symbols since it is found most efficient with respect to readability [Pi08].

Brightness³. Few empirical studies exist, which show that these two visual variables improve the readability of graphs or business process models respectively. Identical assumptions are applied for hue and texture as for color (minimize the number of used colors; consider color usability). The empirical study of [KKR11] that subsumes hue under the color aesthetics indicates a stronger preference for color (hue) over brightness for the purpose to visualize changes in business process models.

Orientation. The constructs should be shown in a way that an orientation of the diagram is evolved by the user (mixing of orientation should be avoided). An initial investigation on process model orientation indicates a benefit with respect to readability for a left-to-right flow direction [FS14].

2.2 Designing Descriptive Diagrams

Since no significant training effort is required to “create” a descriptive representation, this representation can easily be used. Moreover, the creation of descriptive diagrams should be a common feature of BPM systems (which is mostly not the case). The creation of diagrams based on this kind of representation would allow process modelers to create “good” business processes in an appropriate way on their own, without having

³ Texture and brightness are not elaborated separately in our context. Brightness and texture (hue) are considered as components of color aesthetic and thus identical assumptions can be applied for hue and texture as for color.

deep knowledge about process modeling languages. For instance, an autocompletion function could be integrated in order to provide lexical templates to be selected for the creation of textual process descriptions. Subsequently, a grammar or syntax has to be defined for the diagram. However, a disadvantage of natural language usage for communication (as process models intent to) is ambiguity. Descriptive or textual diagrams are described using natural language, which is called to cause ambiguity.

When using natural language expressions for diagrams an efficient parsing (decomposition of sentences) should be supported. Generally, a sentence can be decomposed according to the phrase structure grammar [LC57]) or dependency grammar. Dependency refers to the notion that relationships between linguistic units (e.g., words) are directly linked to each other. Grammatical relationships are preserves between linguistic units. The phrase structure grammar also decomposes sentences to linguistic units using a phrase structure tree, which is a recursive decomposition of the whole word sentence into smaller sentences, down to one-word unit without preserving the dependents between the linguistic units [Lo98]. Comparisons between both types of grammar confirm an efficient parsing for dependency grammars.

To sum up our considerations, both types of diagrams have advantages and allow business experts with a limited experience of modeling to create diagrams. Generally, a switch between both modalities should be offered in order to lower the entry barrier to modeling for unexperienced modelers. Based on these considerations, the next section presents two work in progress approaches for the design of depictive and descriptive diagrams for an on- top layer (“Layer 0”).

3 Approaches for Descriptive and Depictive Diagrams

3.1 Generation of depictive diagrams

The design of a depictive diagram for a business process is based on the guidelines discussed in Section 2.1. For this purpose, we use a technique stemming from design thinking (see e.g., [LW11]). In so-called “Tangible Business Process Modeling” plastic elements, which correspond to BPMN iconography, are used to model business processes through play. Particularly, this approach is suitable for process modelers with limited modeling experiences. After the creation of the tangible process model, the process model has to be enriched with additional information in order to make it automatically processible and executable.

As an example we describe the registration of a thesis at a university from the supervisor’s perspective, who first collects data from the student, then generates several documents (e.g., registration form), sends them to the student’s and his own mailbox, and adds two tasks to the supervisor’s personal task list. The process model is shown in Fig. 3. The user selects the needed process activities from the palette on the right hand

side of the tool. The look and feel of the interface is designed closely to the Microsoft Office products family. For instance, the ribbon bar, the fonts and names of the menu items are imitated from MS Office product family. Thus, the user should feel familiar when working with the tool. Also his/her familiarity with the MS Office product family should increase the tool acceptance.

This visualization approach uses the visual variables discussed in Section 2.1 particularly (1) color, (2) shape and (3) symbols are combined allowing an efficient scanning of the process model also particularly by unexperienced users.

Color: The starting point is dyed green (“Instance factory”) and the endpoint is red, which allows the process modeler to quickly identify the starting and end point and thus the range of the diagram. Nodes, which require the user’s interaction, have blue color. All other nodes (workflow activities) are grey. The contrast between grey and the three used colors is high. Altogether, the number of used colors is well-balanced.

Symbols: Each node has a little symbol in the upper left corner, which represents the activity which is performed in this step. The usage of symbol and text allows the user to quickly select the needed construct.

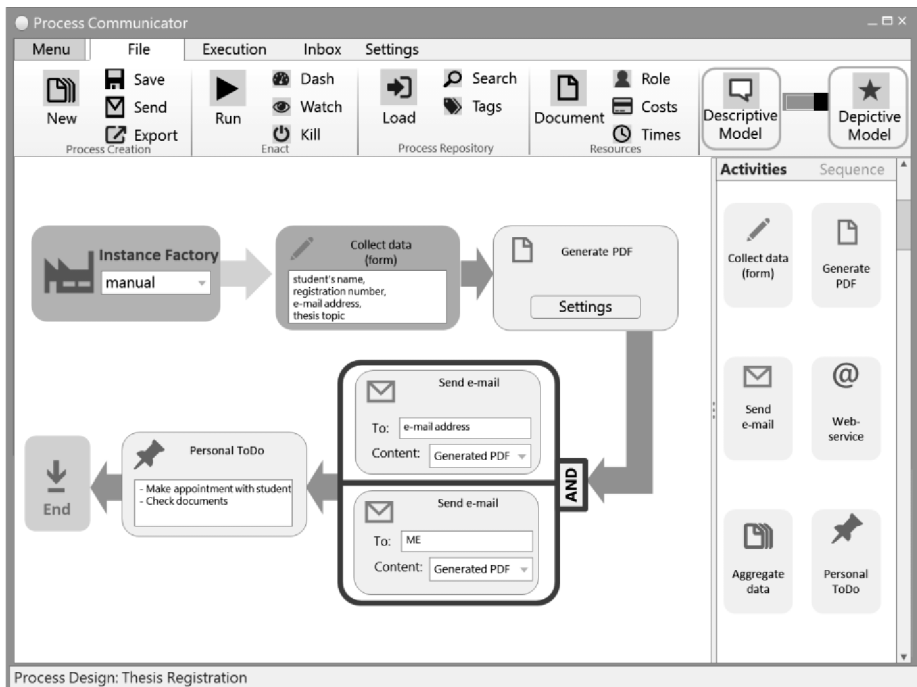


Fig. 3: Approach for a depictive representation (“Layer 0”) in a workflow management system [Le15]

The process model's orientation is evolving as the model grows. That is, in the beginning new elements are inserted from left to right. If one line is full, a line wrap is inserted automatically and new elements are added from right to left. The size of the symbols is changed automatically related to the number of nodes used within the workspace. Brightness is used to highlight special characteristics ("AND" node).

To allow navigation to the subsequent process model a transformation from the depictive diagram to a Petri net is supported as shown in Fig. 4.

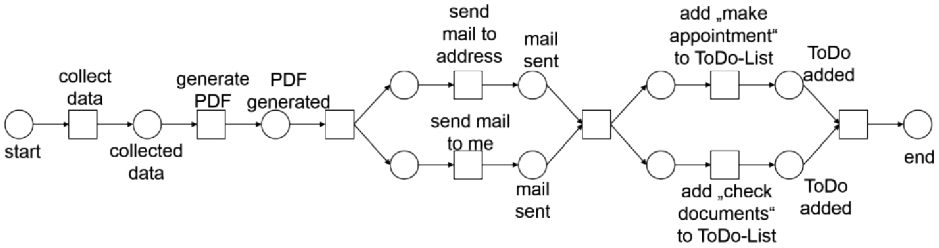


Fig. 4: Petri net resulting from a transformation of the depictive diagram showing the registration process from the supervisor's point of view.

A sequence of activities on the Layer 0 is also translated in a sequence according to the workflow control-flow patterns. The "AND" node is translated into a Parallel Split and Synchronization pattern. Analogously a "XOR" node used in the depictive diagram on Layer 0 would result in an Alternative and Simple Merge pattern.

This automatically generated process model can be further enriched by advanced process modeling by adding additional information, e.g. data objects, which could result in a more detailed Petri net as shown in Fig. 5.

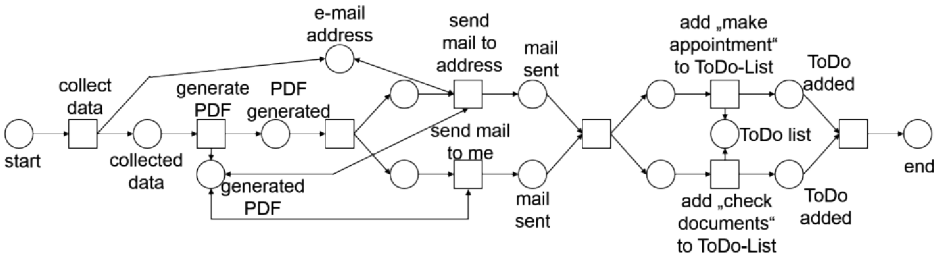


Fig. 5: Process model enriched with additional information by a modeling expert

The suggested visualization approach has been developed according to the design options and guidelines introduced in the Section 2.1. The next section summarizes an approach how to generate graphics from text within the same tool.

3.2 Generation of descriptive diagrams

Descriptive approaches aim to process natural language text. The approach presented in this section is based on a formal model allowing a bidirectional transformation from text to graphics by [Kel14]. This model has been further refined by the introduction of *sentence templates* supporting an efficient decomposition of sentences from natural language text [Ca15]. Assume that Petri nets should be generated from natural language text, the underlying concept can be translated into extended Backus-Naur Form as follows:

```

1: start: sentences*;
2: sentences: placeStart | transitionStart;
3: placeStart: prefix-pl placesList postfix-pl
  prefix-tr transitionList '. ';
4: prefix-pl: 'If ' | 'After ' | 'When ' |
  'As soon as ' | 'In case of ';
5: postfix-pl: ' happened, ' | ' was typed in, ' |
  ' came in, ' | ' is valid, ' | ' is invalid, ' | ', ';
6: prefix-tr: ' I can ' | ' I have to ' |
  ' the system must ' | ' the activity ' | ' then ';
7: placesList: place | ' either ' place ' or '
  furtherplaces | place ' and ' furtherPlaces;
8: furtherPlaces: place | place ' or ' furtherPlaces |
  place ' and ' furtherPlaces;
9: transitionList: transition | ' either ' transition
  ' or ' furtherTransitions | transition ' and '
  furtherTransitions;
10: furtherTransitions: transition | transition
  ' or ' furtherTransitions | transition ' and '
  furtherTransitions;
11: transitionStart: 'Now, ' transitionList '. ' |
  prefix-tr transitionList '. ';
12: place: content;
13: transition: content;
14: content: STRING+ ( ' ' | STRING )*;
15: STRING: (~(' '|'.')+);

```

Fig. 6 shows an exemplary User Interface for a descriptive approach. To use this kind of “modeling” does not require any knowledge of process modeling. Instead the natural language techniques are used to transform the natural language to graphical elements. The sentences can be either typed in manually or they can be recorded and processed by a voice-to-text recognition tool. After the insertion of text, process pattern recognition takes place and the recognized patterns are visualized and displayed immediately. The natural language is exploited in two ways. A bidirectional link between the graphical process model and the textual process description allows checking the correspondence

between the spoken or typed text and the graphical process model at any time. Additionally, we have developed a modeling assistant. Support is available through an automatic selection of natural language templates, which assist in the formulation of sentences of the underlying grammar. The syntax of the templates depends on the modeling language syntax.

The natural language text is inserted on the left hand side, while the corresponding patterns are visualized on the right hand side.

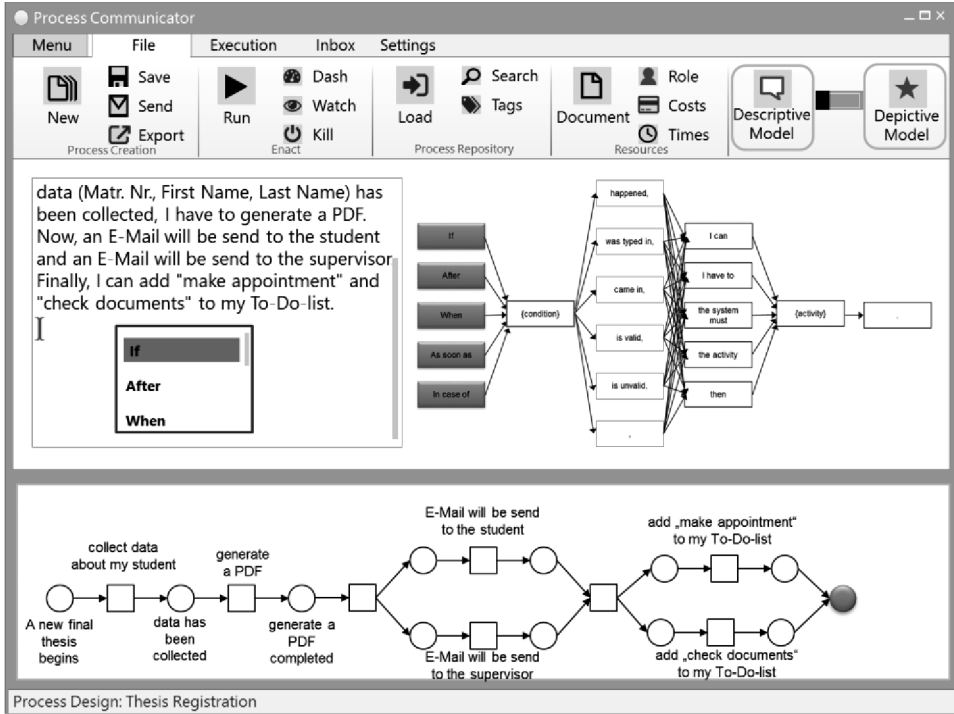


Fig. 6: Approach for a textual abstraction

The templates on the right are connected to a specific modeling pattern and automatically show alternative formulation variants. A sentence is composed by lining up the possible elements, which is illustrated by the connecting arrows. Yellow elements mark placeholders for conditions and blue elements placeholders for activities. The user will automatically be shown up examples when the text input comes either to the text relevant placeholder or the user clicks on the placeholder by mouse. Using our tool, it is possible to create traditional models (Level 1) and display an abstract depictive or descriptive model based on the model. Moreover, it is possible to switch between the depictive and descriptive view in a fast and seamless way.

4 Related Work

Several related scientific works as well as developments from the software industry address the barrier for knowledge externalization.

In regard to modeling methods, much research has been conducted on how novice modelers are constructing process models. In this area, it is known from empirical studies that novice modelers struggle to create “good” process models since they tend to forget important model elements [NKM11] or have problems in finding the right level of abstraction [WBv10]. These observations support our goal of creating a layer for simplified modeling. Also, empirical insights suggest that the combination of abstract graphical symbols (depictive) in conjunction with describing text (descriptive) improves model comprehension for unexperienced modelers [RSR12] which is in line with our ambitions to combine both.

In order to provide abstract graphical models, research has devised techniques to automate process model abstraction [PSW15]. Moreover, in order to switch between models and text, research in the intersection of linguistics and BPM has put forth techniques to generate process models from text [FMP] and vice versa [LMP12]. These approaches are designed in order to transform texts or models, i.e. to be applied *before* or *after* modeling, while our approach is intended to support modeling itself and hence to be applied *during* modeling. In regard to similar modeling approaches, Process Chain Diagrams (PCD, in German “Vorgangskettendiagramm”) [Sc13] already intended to provide a high-level overview layer over a set of more detailed process models that may be linked to the PCD. However, in contrast to our approach, this layer has to be created and updated manually. Another approach hence is to omit the detailed layer and exclusively focus on models that are somewhere in the middle of the granularity continuum ranging from detailed task-oriented models to coarse-grained PCDs. An approach in this direction PICTURE [BPR07]. It offers a lightweight domain-specific language providing a vocabulary and set of symbols to efficiently capture the processes of public administrations. In contrast to this approach, our aim is a generic approach to facilitate the access to Business Process Modeling that is not bound to a specific domain or modeling language. Another approach is the Guarded Process Spaces (GPS) approach [RDR12]. It is applied in the domain of hospitals where process management is important. With GPS, business users can model executable process templates and moreover flexibly adapt running process instances. Both are accomplished using a “navigation paradigm”. This means that the end user is guided in modeling as well as in performing ad-hoc deviations during runtime. In contrast to our approach, this approach is also domain specific. It moreover mixes modeling with execution which is beyond the scope of our work.

In respect to modeling tools, related approaches focus on alternative process model presentations that are easier to understand than e.g. fully-fledged BPMN process models. For example, the *Signavio Process Editor* (cf. www.signavio.com/products/process-editor) provides a mechanism “Quick Model” that allows basic process modeling based

on a spreadsheet-like working platform. With this feature, the product aims to involve all participants in the process design, even those that are not capable of process modeling. The mechanism is based on filling out tables with start- and end-events. In addition, incoming and outgoing documents as well as different roles are assigned to the respective process steps. The tool then generates a process model in BPMN 2.0 notation. Another tool *Blueworks Live* from IBM (cf. www.blueworkslive.com) provides a similar feature. It moreover allows switching forth and back between the lightweight table-based process presentation and the more traditional BPMN-based process model representation.

5 Conclusion and Future Work

This paper first elicits and discusses variables of how to best design diagrams consisting of graphical or textual elements on top of business process models. It then suggests concrete approaches to the design and implementation of such a layer in terms of the necessary functionality and required user interface. These approaches may pave the way for the detailed specification of requirements and elicitation of further design options and choices. These, in turn, can ultimately result in the development of an explanatory design theory [BP10] for on the top layer modeling support systems.

One direction for the future is the complete implementation and user evaluation of both approaches for (abstract) descriptive and depictive design of process models.

References

- [Ai06] Ainsworth, S.: DeFT: A conceptual framework for considering learning with multiple representations. In: *Learning and Instruction*, 2006, 16; p. 183–198.
- [Be83] Bertin, J.: *Semiology of graphics: diagrams, networks, maps*, 1983.
- [BP10] Baskerville, R.; Pries-Heje, J.: Explanatory design theory. In: *Business & Information Systems Engineering*, 2010, 2; p. 271–282.
- [BPR07] Becker, J.; Pfeiffer, D.; Räckers, M.: Domain specific process modelling in public administrations-the PICTURE-approach. In: *Electronic Government*. Springer, 2007; p. 68–79.
- [Ca15] Caporale, T.: A Method for Modeling and Analyzing Business Processes for Knowledge Carriers. In: *Proceedings of the 7th Central European Workshop on Services and their Composition*, Jena, 2015; p. 18–21.
- [Fi12] Figl, K.: Symbol choice and memory of visual models. In: *Visual Languages and Human-Centric Computing (VL/HCC)*, 2012 IEEE Symposium on, 2012; p. 97–100.
- [FMP] Friedrich, F.; Mendling, J.; Puhlmann, F.: Process model generation from natural language text. In: *Advanced Information Systems Engineering*, 2011; p. 482–496.

- [FS14] Figl, K.; Strembeck, M.: On the Importance of Flow Direction in Business Process Models, 2014.
- [FZM15] Fellmann, M. et al.: Requirements Catalog for Business Process Modeling Recommender Systems, 2015.
- [He11] Hegarty, M.: The cognitive science of Visual-Spatial displays: Implications for design. In: Topics in cognitive science, 2011, 3; p. 446–474.
- [Ka11] Kalyuga, S.: Cognitive load theory: How many types of load does it really need? In: Educational Psychology Review, 2011, 23; p. 1–19.
- [Ke14] Keuter, B.: Bidirektionale Abbildung zwischen Geschäftsprozessmodellen und IT-Kommunikationssystemen. KIT Scientific Publishing, 2014.
- [KHO11] Koschmider, A.; Hornung, T.; Oberweis, A.: Recommendation-based editor for business process modeling. In: Data & Knowledge Engineering, 2011, 70; p. 483–503.
- [KKR11] Kabicher, S.; Kriglstein, S.; Rinderle-Ma, S.: Visual change tracking for business process models. In: Conceptual Modeling-ER 2011. Springer, 2011; p. 504–513.
- [Ko15] Koschmider, A.: Identifying Impacts on the Quality of Business Process Models: A Bottom-Up Approach. Technical Report. KIT, 2015.
- [KRD12] Koschmider, A.; Reijers, H. A.; Dijman, R.: Empirical Support for the Usefulness of Personalized Process Model Views. In: Multikonferenz Wirtschaftsinformatik, 2012.
- [KW10] Karagiannis, D.; Woitsch, R.: Knowledge engineering in business process management. In: Handbook on Business Process Management 2. Springer, 2010; p. 463–485.
- [LC57] Lees, R. B.; Chomsky, N.: Syntactic Structures. In: Language, 1957, 33; p. 375–408.
- [Le15] Lehner, J.: Personal BPM—Bringing the Power of Business Process Management to the User. In: Proceedings of the 7th Central European Workshop on Services and their Composition, Jena, 2015; p. 22–25.
- [LGH14] Li, L.; Grundy, J.; Hosking, J.: A visual language and environment for enterprise system modelling and automation. In: Journal of Visual Languages & Computing, 2014, 25; p. 253–277.
- [LMP12] Leopold, H.; Mendling, J.; Polyvyanyy, A.: Generating natural language texts from business process models. In: Advanced Information Systems Engineering, 2012; p. 64–79.
- [Lo93] Lohse, G. L.: A cognitive model for understanding graphical perception. In: Human-Computer Interaction, 1993, 8; p. 353–388.
- [Lo98] Lombardo, V.: A computational model of recovery. In: Reanalysis in Sentence Processing. Springer, 1998; p. 287–325.
- [LW11] Luebbe, A.; Weske, M.: Bringing design thinking to business process modeling. In: Design Thinking. Springer, 2011; p. 181–195.
- [MM03] Mayer, R. E.; Moreno, R.: Nine ways to reduce cognitive load in multimedia learning. In: Educational psychologist, 2003, 38; p. 43–52.

- [Mo09] Moody, D. L.: The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. In: *Software Engineering, IEEE Transactions on*, 2009, 35; p. 756–779.
- [NKM11] Nielen, A. et al.: An empirical analysis of human performance and error in process model development. In: *Conceptual Modeling-ER 2011*. Springer, 2011; p. 514–523.
- [Pi08] Pietersma, D.: Symbol Type and Colour in Graphs. In: *Pharmaceutical Users Software Exchange*, 2008; p. 1–7.
- [PSW15] Polyvyanyy, A.; Smirnov, S.; Weske, M.: Business process model abstraction. In: *Handbook on Business Process Management 1*. Springer, 2015; p. 147–165.
- [RDR12] Reuter, C. et al.: Guarded process spaces (GPS): A navigation system towards creation and dynamic change of healthcare processes from the end-user’s perspective. In: *Business Process Management Workshops*, 2012; p. 237–248.
- [RSR12] Recker, J.; Safrudin, N.; Rosemann, M.: How novices design business processes. In: *Information Systems*, 2012, 37; p. 557–573.
- [SB03] Schnotz, W.; Bannert, M.: Construction and interference in learning from multiple representation. In: *Learning and Instruction*, 2003, 13; p. 141–156.
- [Sc13] Scheer, A.-W.: *ARIS—Modellierungsmethoden, Metamodelle, Anwendungen*. Springer-Verlag, 2013.
- [SO95] Stenning, K.; Oberlander, J.: A cognitive theory of graphical and linguistic reasoning: logic and implementation. In: *Cognitive science*, 1995, 19; p. 97–140.
- [TS86] Treisman, A.; Souther, J.: Illusory words: The roles of attention and of top-down constraints in conjoining letters to form words. In: *Journal of Experimental Psychology: Human Perception and Performance*, 1986, 12; p. 3.
- [WBv10] Wilmont, I. et al.: Exploring intuitive modelling behaviour. In: *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2010; p. 301–313.

Extending different Business Process Modeling Languages with Domain Specific Concepts: The Case of Internal Controls in EPC and BPMN

Michael Radloff¹, Martin Schultz², Markus Nüttgens¹

Abstract: Conceptual models of business processes and related business process modeling languages play a crucial role in today's information systems research and practice. Common BPMLs such as BPMN 2.0 or EPC are widely accepted and applied in various domains. However, such BPMLs provide a set of generic process modeling elements but do not allow for modeling domain specific concepts. This also holds true for control means as one of the key concepts for process audits. To address this gap, this paper presents an empirically grounded extension of the EPC with modeling concepts for process-integrated control means. The results of a laboratory experiment with 58 participants demonstrate that the extension facilitates a comprehensive enactment of process audits. In conclusion, the results of this research project are contrasted with a previously designed BPMN 2.0 extension in order to present insights we gain from extending two different BPMLs with the same domain concept.

Keywords: Process Modeling Language Extension, EPC Extension, Process Audits

1 Introduction

Business process management (BPM) with related business process modeling languages (BPML³) is a well-established research area with high relevance for practical applications [Da06]. Existing BPML like the Business Process Model and Notation (BPMN) and the Event-driven Process Chain (EPC) are accepted in academia and practice. Such BPMLs provide a set of generic modeling constructs for typical elements of a business process (e.g. control flow, data, resource) which makes them applicable in various domains [RRF08]. However, with this broad focus these languages do not provide appropriate modeling elements for domain specific concepts. For comprehensively covering a particular domain these general-purpose BPMLs need to be extended with additional modeling elements. Doing so, specific requirements of a domain and stakeholder perspectives can be addressed more precisely. Such language extensions facilitate model understanding and foster communication among experts of a particular domain.

This also applies to the audit domain. Regulatory requirements directly impact the design and enactment of business processes in and across today's organizations. Accordingly, BPM practitioners and researchers have paid greater attention to compliance and

¹ University of Hamburg, Faculty of Economics - Information Systems

² University of Applied Sciences Wedel

³ In this paper we use the acronym "BPML" to refer to common business process modeling languages.

audit aspects in recent years [LMX07]. In current audit practice the main focus is set on business processes. This approach is based on the assumption that well controlled processes lead to a compliant state of an organization e.g. in terms of fairly presented financial statements [Ru06]. When it comes to auditing a business process, auditors mainly review the design and enactment of control means that are embedded in the process flow. Empirical research results indicate that auditors benefit from an integrated representation of business process models and embedded control means [BHT09]. However, surveys among auditors show that BPMLs are not widely used in current process audit practice [BJJ07, SM14]. This signifies that common BPMLs do not sufficiently meet auditors' requirements for presenting audit-relevant concepts in process models [Ca06, Sa11]. This appraisal complies to the results of previous literature reviews which identify methods for annotating, and enhancing business process models with compliance/audit modeling elements as one of the main open issues on the research agenda for the compliance and audit domain [ASI10, Sa11]. To address this gap, this paper presents an approach for extending the EPC as a wide-spread BPML with modeling elements for control means. An existing EPC meta-model is extended and notation elements are introduced to provide appropriate concepts for enriching process models with control means. A laboratory experiment with 58 participants is used to evaluate the utility of the designed extension. This research project marks a further step in a larger research endeavor. The proposed extension is based on thorough empirical research work in the audit domain, especially focusing on auditors' conceptualization and representation of control means in the context of process audits [SR14]. In previous research work we have extended the BPMN 2.0 to provide modeling elements for process-integrated control means. Against this background, the contribution of this paper is twofold: 1) we propose an extension of the EPC for control means; and 2) discuss insights that we gained from extending two separate BPMLs with the same domain concept.

The remainder of this paper is structured as follows. The next section elaborates on related research regarding the EPC, the extension of BPMLs and relevant concepts in the audit domain. Section 3 outlines the applied research approach and presents the proposed extension for the abstract and concrete syntax of the EPC and a corresponding XML-based interchange format. An example demonstrates the applicability of the extension. The results of the evaluation are summarized in section 4. In section 5 the insights we gain from extending two different BPMLs with the same domain specific concept are comprehensively discussed. The paper closes with a conclusion along with implications for future research work in section 6.

2 Related Research

BPMLs as basis for conceptual models of business processes are a research topic with a long tradition and ongoing attention in information systems research and practice [LS07]. Most common example for BPML are Petri Nets [Pe62], IDEF [MM98], Unified Modeling Language (UML) Activity Diagrams [OMG11], BPMN 2.0 [ISO13] and

EPC [KNS92]. The latter two are well-established semi-formal BPMLs which find widespread use in the BPM domain. The EPC was introduced by Keller et al. as a modeling language to represent temporal and logical dependencies in business processes [KNS92]. It became popular as a BPML in the context of reference modeling (e.g. the SAP reference model [CK97]) and is used in common business software (i.e. Microsoft Visio) as well as open-source tools (e.g. bflow* tool box [Bo10]. The ‘Architecture of Integrated Information Systems’ (ARIS) utilizes the EPC as a central method to conceptually integrate functional, organizational, data, and output perspectives in process modeling and information systems design [STA05]. To enable interchangeability of EPC models, Mendling and Nuettgens [MN06] complement the EPC with an XML-based interchange format termed as EPC markup language (EPML). Several approaches added domain specific concepts to EPC, e.g. for performance measures [Ko08], risk-oriented concepts [RW08], inter-organizational process modeling [SV05], and financial statements [MN14]. BPMN 2.0 is a broadly accepted BPML with standardized meta-model, notation elements, and XML-based interchange format. It is –in contrast to the EPC– defined as ISO standard and provides an extensibility mechanism that enables the integration of new concepts while ensuring validity of BPMN 2.0 core elements [ISO13]. A recent literature review lists not less than 30 domain specific BPMN 2.0 extensions [BE14].

In this context, in a previous research project we proposed a BPMN 2.0 extension that provides modeling concepts for process-integrated control means [SR14]. The design of this extension is based on empirical research results that we acquire with a multi-method research approach (expert interviews, online survey). Aim was to rigorously derive all relevant concepts and their attributes as well as modeling requirements for process audits [SMN12]. In total, 12 modeling concepts and their relations were identified. These results were transformed to an empirically grounded conceptual model that describes all relevant concepts of the process audit domain and their relations to business processes [Sc13]. This conceptual model was used for designing the BPMN 2.0 extension and also lays the basis for the EPC extension outlined in this paper.

One key concept is “control means” which constitute recommended courses of action to ensure that a desired state of a process (control objective) is achieved [SHF11]. They are either directly integrated in a process (e.g. invoice approval) or are independently performed from a particular process (e.g. internal audit) [ISA12]. Our empirical analyses reveal that auditors conceptualize control means mainly as process-integrated measures respectively as ‘special’ process activity [SMN12]. Essential attributes for control means are timing (preventive or detective), nature (manual or automated), and frequency (time period a control means has to occur, e.g. daily, monthly).

On a general level, there is an ongoing debate on appropriate methods for supporting domain aspects in conceptual modeling. There are two options: 1) developing a new domain specific modeling language (DSML) [Fr10]; or 2) extending existing general-purpose modeling languages. We opt for the later approach as a large number of concepts that have been identified as relevant for process audits are already well-considered in existing BPMLs [RRF08]. The approach also enables reuse of well-known modeling

concepts, benefits from advantages of established BPMLs (standardization, tool support, practical relevance), and avoids costly development of a new DSML [BE14]. Against this background, several researchers recently paid increased intention to the extensibility of BPMLs in general. Atkinson et al. [AGF13] identify three different strategies for supporting the extension of modeling languages: 1) *In-built*: Mechanisms enable changes to the meta-model without changing the core elements; 2) *Meta-model Customization*: The meta-model is directly changed as the language does not provide appropriate mechanisms for extensions; and 3) *Model annotation*: An extension is defined in a separate language and instances of the extension are linked to instances of the language core elements via model weaving. As stated earlier, BPMN 2.0 provides an extensibility mechanism (In-built) whereas EPC lacks of such mechanisms and requires direct changes of the meta-model to enable domain specific extensions (Meta-Model Customization). Braun and Esswein [BE15] propose a generic framework for meta-model modifications which systematizes operations (add, delete, alter) for changing components of a modeling language (abstract syntax, concrete syntax, semantics). With this paper we want to contribute to this research stream on domain specific BPML extensions.

3 Extension for Internal Controls – EPC+C

3.1 Research Approach

The research presented in this paper follows the design science research approach [He04]. The designed artifact is an extension to the EPC and the corresponding XML-based interchange format EPML. The relevance of the artifact stems from the fact that methods for annotating process models with compliance modeling elements are still lacking [Sa11]. The applied research method is conceptual modeling. A BPML consists of an abstract syntax (meta-model), concrete syntax (notation), and semantics [HR04, Pa06]. Accordingly, the proposed artifact extends the EPC meta-model and notation to provide a complete language extension for control means.

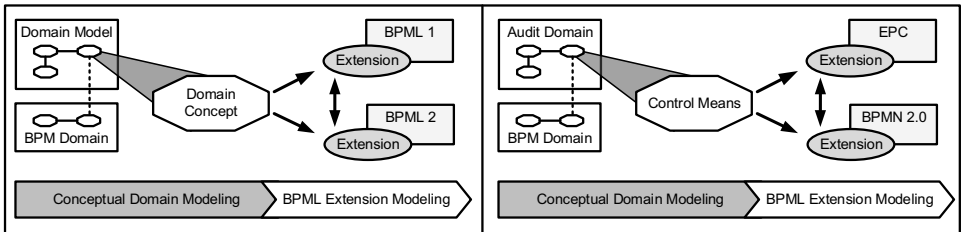


Fig. 1: General concept of Inter-BPML extensions and an instantiation

For the evaluation of the EPC extension we choose a 1 x 2 between-group laboratory experiment with participants from the audit domain. With this experiment design we focus on the stakeholders' perception of the EPC extension regarding understandability

Our extension comprises the classes *ProceduralControlMeans*, *AuditResult*, *Risk*, and *ControlObjective* which represent all relevant control means-related domain concepts. The core element of this extension is the class *ProceduralControlMeans*. It provides a set of attributes that represent relevant characteristics of control means. These attributes are *frequency*, *timing* (preventive, detective), and *nature* (manual, automated) which are further specified by corresponding enumerations. The attribute *recommendedAction* defines an action that should be performed to enact the control means. The class *ProceduralControlMeans* is linked to the EPC core element *Function* which represents a process activity in an EPC model. This linkage is in line with auditors' conceptualization of

control means as ‘special’ process activity (cf. section 2). By means of this composition, an EPC *Function* inherits attributes of *ProceduralControlMeans* and is thereby extended. The classes *AuditResult*, *Risk*, and *ControlObjective* are further elements of the audit domain. They are solely considered from the control means perspective (e.g. risk is also associated to other BPM concepts). The classes *Risk* and *ControlObjective* are linked to the EPC element *DataFlowConnector* and inherit from the *AdditionalProcessObject* class. Two constraints have to be defined for these elements: 1) A *Risk* may only be connected to a *ControlObjective* 2) A control objective may only be attached to a function, which is extended by the *ProceduralControlMeans* class.

The interchange format for the BPMN 2.0 extension was defined with the help of a dedicated method and model transformations [SCV11]. In the case of EPC and EPML such a method is not available. However, extensibility was one design principle for constructing the EPML XML schema [MN06]. EPML provides several possibilities for domain specific extensions. We use the extension point in the XML complex type *tEpcElement*. Our EPML extension leverages the results of the BPMN 2.0 extension as the generated XML complex types are reused for constructing an extended EPML XML schema.

For visualizing the EPC+C elements, we propose an extension of the EPC notation. The notation extension should not alter core EPC notation elements and should be as close as possible to it (look and feel). Accordingly, our design considers the EPC notation elements and existing approaches for EPC extensions. The prior described *ProceduralControlMeans* enhances the EPC function. For the attribute *timing* a marker is required to distinguish detective and preventive control means. The marker concept is inspired by the EPC extension of Rieke and Winkelmann [RW08]. A single lens denotes detective control means whereas a lens encircled by a shield indicates a preventive control. A similar design for both icons facilitates the perceptibility of control means in a process model. Likewise, a marker is used to separate risks and control objectives from EPC information objects. A checkbox icon denotes a *ControlObjective* and an exclamation mark a *Risk*. Fig. 3 depicts all notation elements of our EPC extension.

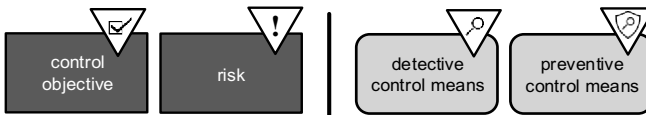


Fig. 3: Notation Elements for the EPC+C Extension

3.3 Application Example for the EPC extension

As an application example, Fig. 4 portrays an EPC model for a simple purchase-to-pay process fragment (left part) and the corresponding section of the EPML document (right part). Such an EPC process model is also used in the laboratory experiment to introduce the extension of the EPC notation to the participants. The process starts with the event ‘Purchase Requisition created’. The purchase requisition is processed by the function

‘Order Goods’. After the order goods are received (event ‘Goods received’) the goods receipt is reconciled with the corresponding purchase order. The EPC function ‘Reconcile received Goods with Order’ is extended with all attributes from class *Procedural-ControlMeans* (frequency, nature, timing, and recommendedAction) which transforms a common EPC function to an detective control means. The EPC process model also includes the EPC+C notation elements for the concepts *ControlObjective* and *Risk*. The EPML document in Fig. 4 shows that the control means is related to an audit result and refers to a control objective which is linked to a specific risk. Details for these concepts are given as additional XML elements of the EPC node (not the function node). In Fig. 4 the extended parts of the EPML document are highlighted in grey boxes.

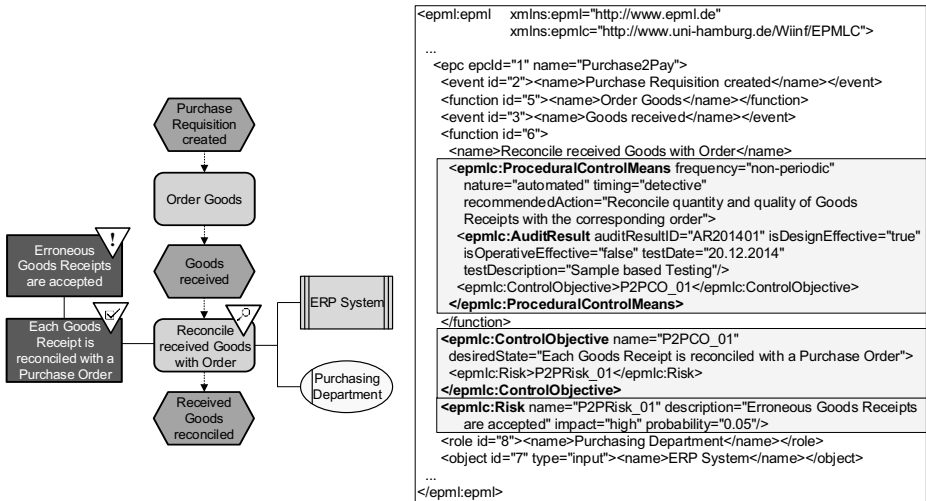


Fig. 4: Sample Process as EPC Model (left) and EMPL description (right)

4 Evaluation

4.1 Experimental Design⁴

The evaluation in a design science research project tries to measure how well the designed artifact supports a solution for the addressed problem [Pe07]. In this project, a 1x2 between-group experiment is used for evaluating the concrete syntax of the EPC extension. The two groups receive the same information on a purchase-to-pay process (process model) and embedded control means (controls matrix). For one group the process model is extended with EPC+C. However, a transformation of one presentation into

⁴ This section is based on the published results of the BPMN extension as the experiment for both extensions are equivalently designed [SR14].

the other is possible without loss of information [He14]. In process audits, the main task for process auditors is to interpret existing models. For evaluating the quality of model interpretation two perspectives are discussed in academia: *interpretational fidelity* (how faithfully does the interpretation of the model supports the reader to comprehend the domain semantics included in the model?) and *interpretational efficiency* (what resources are required for interpreting the model?) [BM06, Re13]. In similar studies interpretational fidelity is measured by using comprehension tasks to test how well a model user understands the content of a given model (comprehension task performance) [MSR12, Re13]. To operationalize interpretational efficiency, usually the time is measured a model user needs to complete these comprehension tasks (comprehension task efficiency) [He14, MSR12, Re13]. In our experiment following hypotheses are tested: Comprehension task performance (H1) and Comprehension task efficiency (H2) are positively affected by using the EPC extension. The experiment is implemented as an online accessible test using Qualtrics research suite [Qu13]. As participants internal and external auditors as well as process analysts are recruited by utilizing business networks (e.g. XING) and large audit associations (e.g. DIIR, ISACA). After answering questions about demographic characteristics, process and audit knowledge, the participants are randomly assigned to one of the groups (EPC, EPC+C). The experiment comprises three process models with increasing complexity. As first task for each model, the participants are asked to identify control means by clicking them in the model. As second task, multiple choice questions have to be answered, which refer to the process control flow and embedded control means (model 1: 4 questions, model 2: 7, model 3: 15).

4.2 Results of the Experiment

In total 58 participants passed the experiment (EPC = 28 and EPC+C = 30). Chi-square tests confirm that in terms of demographic characteristics (age, education, employment status, working experience, self-reported process modeling knowledge and self-reported audit knowledge) the participants are equally represented in both groups. Only for the characteristic “gender” there is an unequal distribution (in total: 45 male/13 female, EPC: 19/9, EPC+C: 26/4). However, we noticed no significant differences for the measured variables for male and female participants. In order to test our hypothesis we conduct a Mann-Whitney-U-Test (MWU) [BD95] as non-parametric test for small sample sizes ($n < 30$) to compare the means of both groups (EPC, EPC+C) regarding five variables. For comprehension task efficiency the time is measured (in seconds) the participants need to conduct the controls identification task for the first control means (*Duration – First Identification*), the identification of all control means (*Duration - Controls Identification*), and to answer the comprehension questions (*Duration – Comprehension Questions*). Comprehension task performance is operationalized by the number of correctly identified control means (*Correctly identified Controls*) and correctly answered questions (*Correct Answers*). All variables are added up for all three models.

Tab. 1 summarizes the results of the MWU tests. The results demonstrate that both groups significantly differ in the variables for duration of answering the comprehension

questions, duration of full control means identification and the number correctly identified control means. The mean rank of correctly identified control means for EPC+C (34.15) in comparison to EPC (24.52) indicates significantly more correct identified control means in the EPC+C group which supports the hypothesis H1. For comprehension task efficiency, the results indicate contradictory findings. The mean rank of variable Duration – Controls Identification for the EPC+C group (17.12) is significantly lower than for the EPC group (33.21) which indicates a significantly faster perception of the control means in the EPC+C group. However, regarding the duration for the comprehension questions, the mean ranks for EPC group (18.81) and EPC+C group (27.44) indicate, that the participants in the EPC group answered the questions much faster. Accordingly, regarding H2 (comprehension task efficiency) the experimental results are inconclusive. It can be assumed that the extension supports perception of control means, but does not significantly facilitate the understanding of the whole process model.

	Mean		Mean Rank		MWU ^[1]	Z ^[2]	AS ^[3]
	<i>EPC</i>	<i>EPC+C</i>	<i>EPC</i>	<i>EPC+C</i>			
<i>Comprehension Task Performance</i>							
Correctly identified Controls (0-7)	6.54	6.90	24.52	34.15	280.50	-2.854	0.004
Correct Answers (0-26)	20.57	20.80	28.13	30.78	381.50	-0.603	0.547
<i>Comprehension Task Efficiency</i>							
Duration – Controls Identification (sec)	91.98	50.99	33.21	17.12	103.00	-3.940	0.000
Duration – First Identification (sec)	40.77	31.71	29.33	19.69	161.00	-2.397	0.170
Duration – Comprehension Questions (sec)	449.43	537.58	18.81	27.44	164.00	-2.172	0.030

^[1] Mann-Whitney-U Value ^[2] Empirical Z Value (asymptotic probability of error for n<30), ^[3] Asymptotic significance

Tab. 1: Means & M-W-U Results for the Groups EPC (n=28) and EPC+C (n=30)

These results are in line with the comparable experiment for our BPMN 2.0 extension. In the BPMN experiment, also the duration of control means identification was significantly improved by the extension. Accordingly, both experiments indicate a positive effect on *interpretational efficiency*. One possible interpretation is that the integrated documentation of control means and process control flow reduces the cognitive load for process model and control matrix interpretation. However, the EPC experiment reveals a negative effect on the duration for answering the comprehension questions. This might indicate that the additional information in the process model hampers model understanding or forces the participants to investigate the process model more deeply. In contrast to the BPMN extension, the results of the EPC+C experiment show a positive effect on *interpretational fidelity*. These results are potentially caused by the different extension designs in both BPMLs. For BPMN only markers were used whereas for EPC also the additional elements (e.g. risk) are represented as separate notational elements.

5 Discussion

With accumulated experiences from the design of two BPML extensions, in the following we discuss general aspects of the design process, domain modeling, reusability of

BPML core elements, and BPML extensibility mechanisms in general.

BPML-independent Domain Modeling: The first step for the design of a domain specific extension should be a comprehensive domain analysis that results in a complete conceptual model of the application domain. This modeling step should be conducted independently from a specific BPML in order to avoid adverse effects to clarity and accuracy of the domain model. This recommendation complies with related methods for domain specific BPML extensions [Br14, SCV11]. Such a profound domain model facilitates a consistent design of extensions for different BPMLs. It ensures that the domain specific concepts are consistently interpreted for each BPML when considering BPML specific capabilities and restrictions. The results of our two BPML extensions demonstrate that in regard to the domain specific concepts the designed extensions for both BPMLs are quite similar. For instance, the extension of the XML-based interchange format of BPMN could be reused for EPML with only slight changes.

Conceptual Link of Domain Concept & BPM Domain: As a second step, we recommend relating the constructed domain model with a generic, BPML independent conceptual model for business processes to explicitly outline all relevant relations between a process model and the domain specific concepts. In our research projects the established conceptual link between BPM and the application domain facilitates identifying BPML concepts that lent themselves for reuse in the design of the BPML extension [Sc13]. This recommendation supports the idea of an equivalence check to evaluate conceptual and semantic links in an early design stage [Br14]. In addition, this link should guide the subsequent development of BPML specific extensions to ensure consistent semantics and conceptualizations of domain concepts across several BPMLs. For instance, control means should be consistently conceptualized as an activity in different BPMLs.

BPML Extension Mechanisms: In our research projects we propose an extension for a BPML with a built-in extension mechanism (BPMN 2.0) and a BPML without specifications for extensibility (EPC). The advantage of the latter is obviously the design freedom for the construction of the BPML extension. Drawbacks are uncontrolled changes to the meta-model and violations of the separation of concerns design principle which may adversely affect the understandability of the BPML extension [AGF13]. It also permits different extension approaches which lead to different non-comparable and non-interoperable BPML extensions. Furthermore, tool support for such extensions is not ensured which restricts its practical relevance. However, an in-built extension mechanism does not solve all these problems. The extension mechanism first of all ensures the consistency of the BPML meta-model. For instance, the BPMN 2.0 extension mechanism does not support a semantic link for extended elements. In particular, the added elements and attributes can be attached to any BPMN 2.0 core element and not only to an activity as it is intended in our case. This restriction can only be accomplished by adding further rules to an extension in textual language or by conceptual domain specific extension models including the link of domain concepts to BPML elements. Furthermore, such an extension mechanism does not provide methodological support for the design of an extension. Designed extensions that comply with the extension mechanism can indeed

be interchanged between modeling tools but not all tools necessarily support such extensions since it is not required as the standard allows different levels of compliance [ISO13]. Hence, an interchange of a process model can lead to a loss of extended concepts. In summary, we conclude that the extension mechanism is only a protection for the abstract syntax but does not appropriately guide the design of an extension.

Reuse of Existing BPML Concepts: For our BPML extensions we focus on the reuse of core BPML modeling elements. For example, we use the BPMN marker concept to represent the control means attribute *timing* and the task concept for specifying the attribute *nature*. In contrast, the EPC extension uses the *additional process element* concept to represent *Risk* and *ControlObjective* as own notation elements. This approach complies with ARIS which integrates elements from different modeling perspectives i.e. organizational units from an organizational model. Our extension may refer to an ‘audit perspective’ in which separate domain models can be defined. However, the native modeling concepts of the EPC do not allow representing all domain specific concepts. In fact, we apply a marker concept which was proposed by another EPC extension [RW08] to represent the control means attribute *timing*. In case of BPMN, the risk and control objective are not integrated in the concrete syntax. This is a design decision we made, as the focus of the BPMN extension is set to control means. To cover the complete semantics of e.g. the concept risk a separate extension is required to include all relevant risk related aspects. Such a comprehensive augmentation of a BPML can lead to a language ‘defacement’ as stated by Braun and Esswein [BE15]. Alternatively, the concepts *ControlObjective* and *Risk* could be added by using BPMN 2.0 annotations. However, annotations are not directly connected to the process flow and do not meet the complex semantic of these two concepts. In conclusion, it can be stated that BPML core elements may support domain concepts differently. Measures and methods to evaluate the degree of fitting between domain concepts and BPML core elements would facilitate the design of effective BPML extensions.

6 Conclusion and Further Research

The design of dedicated modeling languages for an application domain is an important research topic for conceptual modelers. This also applies to the audit domain. Accordingly, in recent years researchers have been paying more attention to a comprehensive modeling support for audits. Nevertheless, recent research results show that methods for annotating, and enhancing business process models with audit modeling elements are still lacking [Sa11]. To address this gap, the paper presented an extension to the EPC. This extension enables an integrated representation of business processes and control means to support the enactment of process audits. A laboratory experiment with 58 participants demonstrated that the extension increases auditors’ interpretational efficiency compared to a separated documentation of process models and control means. A comparison with a similar extension for BPMN 2.0 revealed essential design aspects when it comes to extending general-purpose BPMLs with domain specific concepts. We identi-

fied a BPML-independent domain modeling and the construction of a conceptual link between the domain model and a generic conceptual model of a process as crucial steps in the design process for a BPML extension. Furthermore, we discussed the reuse of existing modeling concepts as well as implications and shortcomings of BPML extension mechanisms and a methodology for the design of a BPML extension.

The results of the evaluation and especially the discussion on BPML extensions point to several opportunities for fruitful research directions. The case of EPC demonstrates that the design of an extension requires a standardized meta-model to facilitate the design of additional domain specific extensions. Such a commonly accepted meta-model for EPC should be established in the EPC research community to increase its dissemination. On a more abstract level, our discussion reveals several shortcomings regarding extension mechanisms and methodical support for extending BPMLs in general. Both aspects should be considered more intensely by the BPM research community. In this regard, general principles for designing effective BPML extensions that foster interoperability between various extensions from potentially different domains and the design of an extension repository would be further valuable research contributions. These artifacts would facilitate reuse and combination of existing research results in terms of cumulative design science research and would lay a basis for methods to evaluate the completeness and effectiveness of BPML extensions. These topics remain on our research agenda.

References

- [AGF13] Atkinson, C.; Gerbig, R.; Fritzsche, M.: Modeling Language Extension in the Enterprise Systems Domain. IEEE, Vancouver, 2013; pp. 49–58.
- [ASI10] Abdullah, N.S.; Sadiq, S.; Indulska, M.: Information Systems Research: Aligning to Industry Challenges in Management of Regulatory Compliance. PACIS 2010 Proceedings, 2010.
- [BD95] Bortz, J.; Döring, N.: Forschungsmethoden und Evaluation. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.
- [BE14] Braun, R.; Esswein, W.: Classification of Domain-Specific BPMN Extensions. In (Frank, U.; Loucopoulos, P.; Pastor, Ó.; Petrounias, I. eds): The Practice of Enterprise Modeling. Springer Berlin Heidelberg, 2014; pp. 42–57.
- [BE15] Braun, R.; Esswein, W.: A Generic Framework for Modifying and Extending Enterprise Modeling Languages. 2015.
- [BHT09] Bierstaker, J.L.; Hunton, J.E.; Thibodeau, J.C.: Do Client-Prepared Internal Control Documentation and Business Process Flowcharts Help or Hinder an Auditor's Ability to Identify Missing Controls? In: AUDITING: A Journal of Practice & Theory Vol. 28, No. 1, 2009; pp. 79–94.
- [BJJ07] Bierstaker, J.; Janvrin, D.; Jordan Lowe, D.: An Examination of Factors Associated with the Type and Number of Internal Control Documentation Formats. In: Advances in Accounting Vol. 23, No. 2007; pp. 31–48.
- [BM06] Burton-Jones, A.; Meso, P.: The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model. In: Journal of the Association for Information Systems Vol. 9, No. 12, 2008;

- pp. 748–802.
- [Bo10] Böhme, C.; Hartmann, J.; Kern, H.; et al.: bflow* Toolbox - an Open-Source Modeling Tool. Business Process Management Demonstration Track 2010.
 - [Br14] Braun, R.; Schlieter, H.; Burwitz, M.; Esswein, W.: BPMN4CP: Design and implementation of a BPMN extension for clinical pathways. 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2014; pp. 9–16.
 - [Ca06] Carnaghan, C.: Business process modeling approaches in the context of process level audit risk assessment: An analysis and comparison. In: *International Journal of Accounting Information Systems* Vol. 7, No. 2, 2006; pp. 170–204.
 - [CK97] Curran, T.; Keller, G.: SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. 1997.
 - [Da06] Davies, I.; Green, P.; Rosemann, M.; et al.: How do practitioners use conceptual modeling in practice? In: *Data & Knowledge Engineering* Vol. 58, No. 3, 2006; pp. 358–380.
 - [Fr07] Frank, U.: Evaluation of Reference Models. In (Fettke, P.; Loos, P. eds): *Reference Modeling for Business Systems Analysis*. IGI Global, Hershey, USA, 2007; pp. 118–140.
 - [Fr10] Frank, U.: Outline of a method for designing domain-specific modelling languages. ICB-Research Report, 2010.
 - [He04] Hevner, A.R.; March, S.T.; Park, J.; Ram, S.: Design science in information systems research. In: *MIS Quarterly* Vol. 28, No. 1, 2004; pp. 75–105.
 - [He14] van der Heijden, H.: Effects of diagram format and user numeracy on understanding cash flow data. 37th Annual Congress of the European Accounting Association. Tallinn, 2014.
 - [HR04] Harel, D.; Rumpe, B.: Meaningful modeling: what’s the semantics of “semantics”? In: *Computer* Vol. 37, No. 10, 2004; pp. 64–72.
 - [ISA12] International Federation of Accountants: ISA 315 (Revised), Identifying and Assessing the Risks of Material Misstatement through Understanding the Entity and Its Environment. 2012.
 - [ISO13] ISO: ISO/IEC 19510:2013 - Information technology – Object Management Group - Business Process Model and Notation 2.0. 2013.
 - [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: *Semantische Prozeßmodellierung auf der Grundlage “Ereignisgesteuerter Prozessketten (EPK).” Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes*, 1992.
 - [Ko08] Korherr, B.: *Business Process Modelling - Languages, Goals and Variabilities*. Vienna University of Technology
 - [LMX07] Liu, Y.; Muller, S.; Xu, K.: A static compliance-checking framework for business process models. In: *IBM Systems Journal* Vol. 46, No. 2, 2007; pp. 335–361.
 - [LS07] Lu, R.; Sadiq, S.: A Survey of Comparative Business Process Modeling Approaches. In *Business Information Systems*. Springer Berlin Heidelberg, 2007; pp. 82–94.
 - [MM98] Menzel, C.; Mayer, R.J.: The IDEF Family of Languages. In (Bernus, A.P.D.P.; Mertins, P.D.K.; Schmidt, P.D.G. eds): *Handbook on Architectures of Information Systems*. Springer Berlin Heidelberg, 1998; pp. 215–249.
 - [MN06] Mendling, J.; Nüttgens, M.: EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). In: *Information Systems and e-Business Management* Vol. 4, No. 3, 2006; pp. 245–263.
 - [MN14] Müller-Wickop, N.; Nüttgens, M.: Conceptual Model of Accounts Closing the Gap between Financial Statements and Business Process Modeling. *Proceedings of the Modellierung 2014*. Wien, Austria, 2014.

- [MSR12] Mendling, J.; Strembeck, M.; Recker, J.: Factors of process model comprehension—Findings from a series of experiments. In: *Decision Support Systems* Vol. 53, No. 1, 2012; pp. 195–206.
- [OMG11] OMG: Unified Modeling Language Infrastructure 2.4.1. 2011.
- [Pa06] Patig, S.: *Die Evolution von Modellierungssprachen*. Frank u. Timme, 2006.
- [Pe07] Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. In: *Journal of Management Information Systems* Vol. 24, No. 3, 2007; pp. 45–77.
- [Pe62] Petri, C.A.: *Kommunikation mit Automaten*. Rhein.-Westfäl. Inst. f. Instrumentelle Mathematik an der Univ. Bonn, 1962.
- [Qu13] Qualtrics: Qualtrics Research Suite. Provo, Utah, USA, 2013.
- [Re13] Recker, J.: Empirical investigation of the usefulness of Gateway constructs in process models. In: *Eur J Inf Syst* Vol. 22, No. 6, 2013; pp. 673–689.
- [RRF08] Rosemann, M.; Recker, J.; Flender, C.: Contextualisation of business processes. In: *International Journal of Business Process Integration and Management* Vol. 3, No. 1, 2008; pp. 47–60.
- [Ru06] Ruhnke, K.: Business Risk Audits: State of the Art und Entwicklungsperspektiven. In: *Journal für Betriebswirtschaft* Vol. 56, No. 4, 2006; pp. 189–218.
- [RW08] Rieke, T.; Winkelmann, A.: Modellierung und Management von Risiken. Ein prozessorientierter Risikomanagement-Ansatz zur Identifikation und Behandlung von Risiken in Geschäftsprozessen. In: *Wirtschaftsinformatik* Vol. 50, No. 5, 2008; pp. 346–356.
- [Sa11] Sadiq, S.: A Roadmap for Research in Business Process Compliance. In (Abramowicz, W.; Maciaszek, L.; Węcel, K. eds): *Business Information Systems Workshops*. Springer Berlin Heidelberg, 2011; pp. 1–4.
- [Sc13] Schultz, M.: Towards an Empirically Grounded Conceptual Model for Business Process Compliance. In (Ng, W.; Storey, V.C.; Trujillo, J.C. eds): *Conceptual Modeling*. Springer Berlin Heidelberg, 2013; pp. 138–145.
- [SCV11] Stroppi, L.J.R.; Chiotti, O.; Villarreal, P.D.: Extending BPMN 2.0 : Method and Tool Support. Third International Workshop, BPMN 2011, Lucerne, Switzerland, November 21–22, 2011. 2011; pp. 59–73.
- [SHF11] Strecker, S.; Heise, D.; Frank, U.: Prolegomena of a modelling method in support of audit risk assessment. In: *Enterprise Modelling and Information Systems Architectures* Vol. 6, No. 3, 2011; pp. 5–24.
- [SM14] Schultz, M.; Mueller-Wickop, N.: Towards Auditors’ Preferences on Documentation Formats in Business Process Audits. *Modellierung 2014*. Vienna, Austria, 2014.
- [SMN12] Schultz, M.; Mueller-Wickop, N.; Nuettgens, M.: Key Information Requirements for Process Audits – an Expert Perspective. *Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*. Vienna, Austria, 2012; pp. 137–150.
- [SR14] Schultz, M.; Radloff, M.: Modeling Concepts for Internal Controls in Business Processes – An Empirically Grounded Extension of BPMN. In (Sadiq, S.; Soffer, P.; Völzer, H. eds): *Business Process Management*. Springer International Publishing, Cham, 2014; pp. 184–199.
- [STA05] Scheer, A.-W.; Thomas, O.; Adam, O.: *Process modeling using event-driven process chains. Process-Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley Publishing, 2005; pp. 119–146.
- [SV05] Seel, C.; Vanderhaeghen, D.: Meta-Model based Extensions of the EPC for Interorganisational Process Modelling. In (Nüttgens, M.; Rump, F.J. eds): *Proceedings 4th GI-Workshop EPK 2005 - Geschäftsprozessmanagement*. 2005; pp. 117–136.

Findings from an Experiment on Flow Direction of Business Process Models

Kathrin Figl¹, Mark Strembeck²

Abstract: A core aspect of diagrammatic process modeling is the visualization of the logical and temporal order in which tasks are to be performed in a process. While conventions and guidelines exist that promote modeling processes from left-to-right or from top-to-bottom, no empirically validated design rationale can be provided for this choice so far. Therefore, this paper seeks to determine whether some flow directions are better than others from a cognitive point of view. We present the results of a controlled pilot experiment comparing the effects of four flow directions (left-to-right, right-to-left, top-to-bottom, bottom-to-top) on process model comprehension with a small sample size of 44 participants. Although there is a variety of theoretical arguments which support the use of a left-to-right flow direction as convention for process models, the preliminary empirical results of the pilot experiment were less clear-cut and showed that model readers also adapted well to uncommon reading directions.

Keywords: Model Layout, Reading Direction, Flow Direction, Business Process Models.

1 INTRODUCTION

Business processes describe which tasks need to be performed to reach certain business goals. Visual modeling of business processes is associated with several benefits such as a better understanding of the respective processes, improved communication between stakeholders, and easier identification of possible improvements. In general, diagrammatic process models are created using process modeling notations — i.e. sets of graphical symbols and rules for combining them — with the Business Process Model and Notation (BPMN) [BU13] being a de-facto-standard in that area. While such modeling notations also provide means to model actors or data involved in the execution of the process, in this paper we focus on the control flow logic describing the logical and temporal order in which tasks are performed. In particular, we are interested in different options to visualize the pre-defined order of process tasks. In essence, process modeling notations use node-link diagrams, a specific type of directed graphs to depict the process flow, viz. the execution order of tasks in a process. Thus, the position of the start and the end nodes as well as the arrowheads of the edges show the precedence relations between the model elements. From a cognitive point of view, such “arrows” are understood intuitively with respect to their causal and time-related meaning [TV00]. Still, there are

¹ WU - Vienna University of Economics and Business, Institute for Information Systems and New Media, Welthandelsplatz 1, 1020 Vienna, kathrin.figl@wu.ac.at

² WU - Vienna University of Economics and Business, Institute for Information Systems and New Media, Welthandelsplatz 1, 1020 Vienna, mark.strembeck@wu.ac.at

various design options in which direction to “draw” the arrows and how to position the task symbols during modeling. Basically, there are four main options for the overall direction: left-to-right, top-to-bottom, bottom-to-top, right-to-left. While the modeling symbols are usually provided through the respective modeling tool and thus standardized via the corresponding notation, modeling direction is not predefined and users usually start modeling on a blank canvas [EJS11]. In this paper, our objective is to provide insights on how the choice of modeling direction will influence the readability of a model.

A considerable amount of literature has been published on cognitive effectiveness of modeling notations [e.g. MO09]. Several attempts have been made to transfer such insights to the area of business process modeling [GHA10], for instance with respect to different symbol sets including routing symbols of languages [see, e.g., FMS13, FRM13]. Moreover, layout factors such as modularization or line crossings and their impact on process model comprehension have been given considerable attention [EJS11, FKK13, RM08].

However, research has not yet sufficiently addressed the issue of modeling direction. [LA11] mentions the issue of direction in their layout guideline for BPMN diagrams and [FS14] makes a first effort to provide an overview of theories to predict which modeling direction should be optimal from a cognitive point of view favoring left-to-right orientation. However, empirical evidence for the superiority of a left-to-right orientation for process models is still missing, and to the best of our knowledge no empirical evaluation of flow direction has so far been undertaken. To close this gap, this paper reports on an pilot experiment in which we examined the influence of different flow directions on process model comprehension (with a focus on BPMN process models). This research question is important, because the “lack of commonly agreed publicly available guidelines” for style and layout of diagrams may impede quality of modeling tools and of resulting models [ES09]. Empirical foundations will enable the modeling community to establish sound guidelines concerning preferred modeling directions.

The remainder of this paper is structured as follows. The first part provides the theoretical background for our research. The next section describes the experiment we used to test our propositions. Subsequently, we present our data analysis and an examination of the results. Finally, the results are discussed from both theoretical and practical perspectives and we outline future research directions.

2 Background

While the *primary* (modeling) notation defines the concrete syntax of a language (the symbols and the rules for combining them), the *secondary* notation relates to “things which are not formally part of a notation which are nevertheless used to interpret it, such as conventions (e.g., reading a circuit diagram left-to-right and top-to-bottom)” [PE06, p. 293]. Thus, advice and recommendations concerning flow directions in process models

can not only be found in standard documents, but also in layout guidelines or research articles. In contrast to other modeling languages, the BPMN standard document also mentions the flow direction aspect as a recommendation. In particular, the BPMN standard document [BU13, p. 40] gives the advice to either use a left-to right or top-to-bottom flow direction for modeling the sequence flow of a process model (“we also RECOMMEND that modelers use judgment or best practices in how Flow Objects should be connected so that readers of the Diagrams will find the behavior clear and easy to follow. This is even more important when a Diagram contains Sequence Flows and Message Flows. In these situations it is best to pick a direction of Sequence Flows, either left to right or top to bottom, and then direct the Message Flows at a 90° angle to the Sequence Flows. The resulting Diagrams will be much easier to understand.”). However, since a recommendation is not compulsory, it is also important to take into account other literature on the use of flow direction for BPMN diagrams. The recommendation from the BPMN standard we quoted above is also picked up by one of the few available guidelines for laying out BPM diagrams on canvas [LA11]. Moreover, accompanying materials of the OMG standardization organization show that the BPMN example models are almost exclusively modeled left-to-right [BU13]. The convention to model from left-to-right is also reflected by different model layout algorithms. Such algorithms can be included in modeling tools to offer different layout options for orientation, alignment or spacing of elements. Therefore, information on modeling direction can also be found in research papers on layout algorithms for BPMN diagrams. For example, Effinger et al. [EF11] move the start symbol of a process model to the left-hand side and end events to the right-hand side in their layout algorithm. Likewise, [KI09] uses a left-to-right orientation in their layout algorithm for BPMN diagrams, and even gives a specific rationale for this choice: the match with “the horizontal progression of text in western handwriting”.

Top-to-bottom direction seems to be less common than left-to-right, although some authors reported that the flow direction of BPMN diagrams is “usually top-to-bottom or left-to-right” [see, e.g., ESK09].

From a broader perspective, we also discuss how flow direction can be positioned in the overall context of laying out diagrams. Layout of diagrams can be applied on different design levels [ST12]: (1) there are layout principles relevant to all kinds of diagrams (e.g. Gestalt laws, minimizing number of overlapping objects), (2) principles relevant to graphs (e.g. minimizing line crossings, maximizing number of objects in flow direction, keeping uniform flow and edge direction in diagrams [ES09]) and (3) principles relevant to the specific type of diagram (e.g. aligning similar edges or consequences of a decision in a process diagram, placing task symbols right (and not under/above) a split gateway [KI09]). Flow direction as investigated in our study can predominantly be classified as belonging to the 3rd level (specific type of diagram in a specific notation), but also to the 2nd level (graphs in general). To a certain degree, our results might be generalizable to other kinds of directed graphs, since they face the same challenge to visually support the “inherent ordering of elements” by their visual flow [ST12].

As mentioned above, the BPMN standard and other guidelines do not clarify why left-to-right or top-to-bottom should be superior to other directions. In the following, we will draw on related disciplines such as cognitive research on diagram and graph perception to discuss potential effects of using different orientations.

Prior expectations and experience influence how people read diagrams and search for information in diagrams. Winn [WI82, p. 80] mentions that “diagrams convey information about sequences in two ways. First, English-speakers will tend to ‘read’ diagrams in the same way that they read language, from left to right and top to bottom. Diagrams not arranged in this logical sequence would lead to difficulty in information processing and to less learning. Second, lines and arrows can be used to suggest direction”. There is a strong cultural influence of the direction of written language for reading and drawing diagrams. In the area of data models, a diagram type that does not have a predefined reading direction indicated by visual hints as arrows, Nordbotten and Crosby [NC99] showed via eye tracking experiments that users follow these “natural” reading strategies. On average, 60% of their participants followed a text-like reading strategy from left-to-right and top-to-bottom. (The other 40% followed an image-like reading strategy starting in the center followed by scanning in different directions.)

Understanding is easier if diagrams match user expectations and if they are consistent with previously learned diagram schemas [WI83]. Indeed, Winn [WI82] was able to demonstrate that for native English speakers it is more difficult to learn sequences in reversed-order (right-to-left) than in normal-order (left-to-right) diagrams. Similarly, research on flowcharts has shown, that directional orientation influences problem solution quality, time taken to view the charts and time taken to solve the problems [KR83]. Participants performed best when the orientation of flowcharts was consistent with the corresponding reading direction (best results for left-to-right, second-best results for top-to-bottom and worst results for right-to-left flowcharts). In those cases the participants made fewer errors and needed less time.

However, test subjects can develop “reversed diagram” schemas when working with reversed diagrams [WI83]. Winn found evidence for this phenomenon by investigating eye-movements in a study with right-to-left reversed diagrams. At first, participants performed worse in information searching tasks than participants with left-to-right diagrams. However, after four trials the participants adapted their perceptual strategy and no longer started looking at the upper left quadrant which contained little useful information. Winn concluded that if diagrams contradict usual schemas, they are more difficult to understand and provoke more errors in information search tasks at first, but an appropriate strategy can be obtained after time.

Studies in the field of cognitive science have further revealed that humans associate abstract semantic concepts with specific orientations (left, right, top, bottom). With respect to concepts that are relevant in the context of process modeling, the scientific literature shows that a clear preference exists to assign “earlier-later” to left-to-right followed by top-to-bottom and to assign “cause-effect” to top-to-bottom and left-to-right

[HD68, p. 354]. Based on these results it would be most naturally to design process models from left-to-right, and top-to-bottom is likely to be the second best option.

While it is not clear from the literature whether these internal associations between semantic concepts and spatial orientations are actually caused by conventions in visual representations (as diagrams, tables, or text) or vice versa, humans have chosen to use these conventions, because they seem more natural. A variety of examples demonstrate that specific semantic concepts are used predominantly with specific orientations. For instance, when looking at how temporal relations are represented in every-day life it is interesting to note that often top-to-bottom orientation is used (e.g. calendars, school schedules, programs, public transport schedules). Furthermore, in graphs time is often expressed from left-to-right on the horizontal axis [TKW91].

3 Hypotheses

Following from the theoretical discussion above, we will now advance propositions regarding the superiority of specific flow directions in regard to process model understandability. One of the essential arguments is that understanding a process model will be easier if its flow direction matches users' expectations [KR83, WI82]. Such expectations are formed by the direction of written language and typical conventions used in visual representations [TKW91, WI83]. Furthermore, humans associate specific semantic concepts with spatial orientations. Therefore, we suggest that flow direction will influence objective comprehension performance, as well as subjective experience of the comprehension task and the ease of use of the models. As the goal in our study is set at determining the optimal flow direction to contribute to a validation or challenge of existing conventions, we additionally want to address specific hypotheses on an optimal flow direction. In light of the above arguments, we specifically expect that left-to-right flow direction in a model is superior to other flow directions (top-to-bottom, bottom-to-top, right-to-left) with respect to process model comprehension. This is because it is consistent with text reading direction and the existing association between semantic concepts as "earlier-later" and left-to-right [HD68]. Therefore, we hypothesize:

- H1: Flow direction has an influence on process model comprehension accuracy.
 - H1a: Left-to-right flow direction in a model is superior to other flow directions concerning process model comprehension accuracy.
- H2: Flow direction has an influence on process model comprehension efficiency.
 - H2a: Left-to-right flow direction in a model is superior to other flow directions concerning process model comprehension efficiency.
- H3: Flow direction has an influence on the perceived ease of use of the model.
 - H3a: Left-to-right flow direction in a model is superior to other flow directions concerning the perceived ease of use of the model.

4 Research Method

4.1 Experimental Design

We conducted an experiment with model flow direction (with four levels: left-to-right, right-to-left, top-to-bottom, bottom-to-top) and label semantics (with two levels: abstract—text label, concrete—single letter) as two between-groups factors. The label semantics factor was added because for every language text has an inherent reading direction which might interact with the flow direction of the model. In addition, a text label adds additional cognitive load and increases the reading time and effort to assemble information in comparison to a label consisting of a single letter only [MSR12]. Therefore, we considered it important to use experimental groups with and without textual labels. As the approximate sample size requirement for analyzing this research design with an ANCOVA (and expecting medium effect sizes of $f(U) > 0.25$ with type-1 error probability of $\alpha < 0.05$ and sufficient statistical power > 0.80) would be 270 participants (calculated with G*Power 3 software [FA07]), we decided to first run a pilot study with a lower number of participants. Main advantages of pilot experiments are the possibility to evaluate the feasibility of the experimental design and to estimate the variability of differences between experimental groups prior to carrying out a full-scale experiment.

The pilot experiment took place in the context of information systems courses at a European university. In the following, we describe the paper-based questionnaire we used in our study. In particular, it was based on the questionnaire previously described in [FRM13].

4.2 Materials

The questionnaire included four main sections. The first section comprised questions about the participants' demographic data and prior knowledge on process modeling. In the second section we used the set of process modeling questions developed by Mendling and Strembeck [MSR12] to measure prior knowledge. The third section contained a tutorial on BPMN to inform participants about the meaning of the symbols and provided the participants with everything they needed to know to perform the subsequent comprehension tasks. The fourth section of the questionnaire displayed two different process models with eight corresponding comprehension tasks for each model. The models were drawn using basic symbols of the BPMN standard [FRM13, BU13].

In the concrete labels condition, we used actual labels stemming from different domains (an emergency process plan for drinking water pollution with tasks such as 'control drinking water quality', or 'prepare information brochure' and a model on the marketing process in a company with tasks such as 'revise current marketing plan', or 'define quality criteria'). The reading direction for all labels was set horizontal left-to-right for

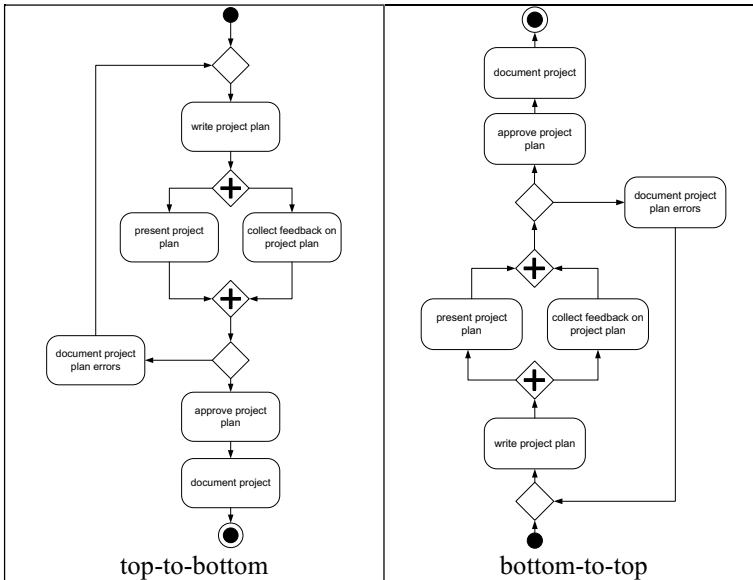
all four experimental groups of differing flow directions, because reading speed for horizontal text is higher than for marquee or rotated text [YU10].

In the abstract label condition we used labels with uppercase alphabetic letters (e.g. ‘A’, ‘B’, ‘C’, ‘D’, etc.) in random order.

The comprehension tasks included questions on the control flow logic between pairs of tasks. In particular, the questionnaire included questions on concurrency (e.g. “[Task A] and [Task B] can be executed in parallel”), exclusiveness, order and repetition. ‘Task A’ and ‘Task B’ were substituted either by the concrete or the abstract label of the corresponding model. The comprehension questions had already been validated in a larger study on notional design and process model comprehension [FRM13].

Participants could answer the respective questions with ‘right’, ‘wrong’ or ‘I don’t know’. After each model we included a scale in which participants could rate the perceived ease of use of the models. The participants were allowed to spend as much time as desired for the completion of the experimental tasks and we asked them to write down the point of time at the beginning and the end of the comprehension questions.

To manipulate the “flow direction” factor in our experiment, we transposed the models to different directions and each experimental group was provided with one of the four flow directions — both models were modelled in the same flow direction. Fig. 1 shows an excerpt of four process models, which are structurally and informationally equivalent, but use different flow directions.



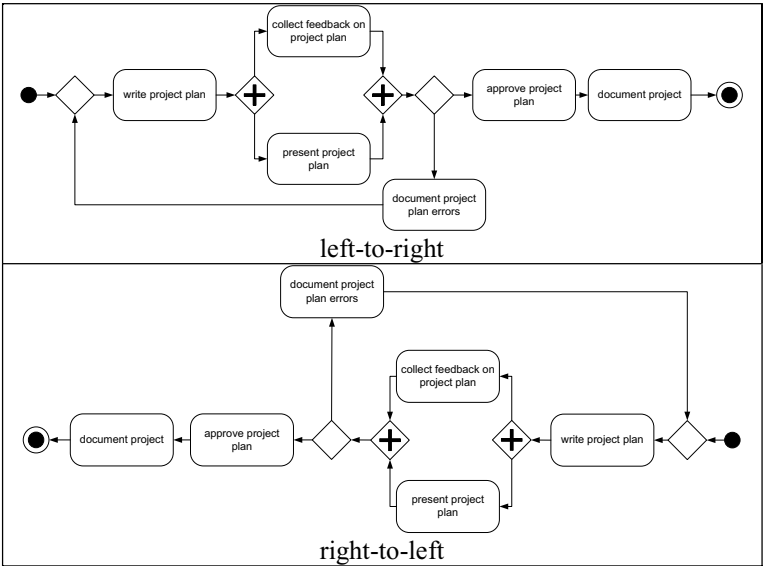


Fig. 1: Detail of a BPMN process model in different flow directions with concrete labels

4.3 Measures

Tab. 1 gives an overview on variables used in the experiment and their measurement.

Variable	Measurement
Comprehension accuracy (dependent variable)	Number of correct answers in the model comprehension tasks (8 comprehension tasks per model)
Comprehension efficiency (dependent variable)	Self-report completion time for the comprehension questions
Perceived ease of use of model (dependent variable)	4 items with a 7-point Likert scale (anchored between “strongly disagree” and “strongly agree”) from Maes and Poels [MP07]
Process Modeling Knowledge (Covariate)	Process modeling test score: 8 items derived from Mendling and Strembeck [MSR12]

Tab. 1: Measurement of variables in the experiment

4.4 Participants and Data Screening

A total of 44 information systems students participated in this study. Half of participants (22) received the abstract label version, the other half (22) the concrete label version. There were 4-6 participants in each cell of the experimental plan (label semantics x flow

direction). Of all respondents, 12 were female (27%) and 32 male (73%). The participants were on average 25 years old. 80% of respondents already had training in process modeling. To screen for possible differences between the experimental groups' demographics, we calculated variance tests, which yielded no problematic differences.

5 Results of the Pilot Experiment

In order to examine the data we collected on the hypotheses, we conducted four univariate analyses of covariance (ANCOVAs). We ran one ANCOVA each for the dependent factors comprehension accuracy (total score), comprehension efficiency (time) and perceived ease of use of the respective model. Flow direction and label semantics were used as independent factors and process modeling test score as covariate.

As can be seen from Tab. 2, no statistically significant differences were found between the investigated flow directions for any of the dependent variables. Thus, our hypotheses suggesting an influence of flow direction on process modeling comprehension accuracy (H1), efficiency (H2) and perceived ease of use of the model (H3) cannot be accepted. In addition, our analyses did not reveal interaction effects between flow direction and label semantics. Fig. 2 depicts comprehension accuracy for different flow directions.

Turning to the experimental evidence on process modeling knowledge, we observe from Tab. 2 that individual knowledge is a relevant influence factor for comprehension accuracy of the comprehension task. Higher individual process modeling knowledge is related to better performance in the comprehension task.

Label semantics did have a significant effect on the variable comprehension efficiency. On average, participants took over 1 minute longer to answer 8 questions on a model with concrete labels (5:36) than with abstract labels (4:02).

	Effect	F (df _{Hypothesis} ; df _{Error})	p	Partial eta squared
Comprehension accuracy (Total score)	Flow direction	1.77 df=3; 36	0.17	0.13
	Label semantics	0.28 df=1; 36	0.60	0.008
	Process modeling knowledge	27.64 df=1; 36	0.000	0.43
Comprehension efficiency (Time)	Flow direction	2.18 df=3; 29	0.11	0.18
	Label semantics	6.39 df=1; 29	0.02	0.18
	Process modeling knowledge	0.00 df=1; 29	0.97	0.00
Perceived ease of use of model	Flow direction	1.66 df=3; 37	0.19	0.12

	Label semantics	0.49 _{df=1; 37}	0.49	0.01
	Process modeling knowledge	2.61 _{df=1; 37}	0.12	0.07

Tab. 2: Experimental results: influence of flow direction

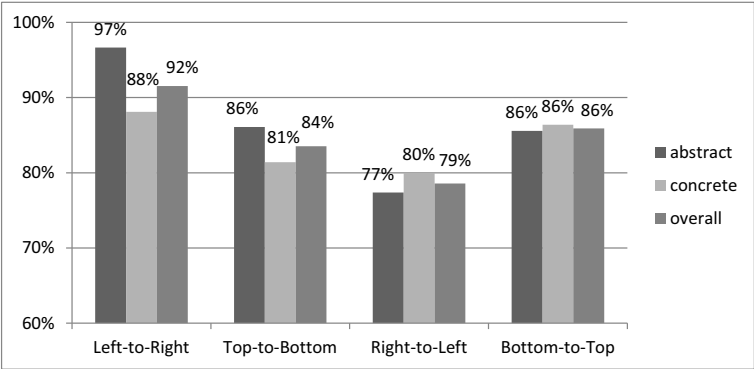


Fig. 2: Flow direction and comprehension accuracy

6 Discussion

The study presented in this paper set out with the aim of assessing the importance of flow direction in process model comprehension. We hypothesized that the use of the flow direction left-to-right would ease comprehension compared to unconventional flow directions, because of a cultural background of written language direction [WI83] and conventional use of left-to-right in diagrams from other areas [TKW91].

Our analyses revealed a number of interesting results. In contrast to our expectations, the experiment did not detect statistically significant evidence for a superiority of the left-to-right flow direction, although absolute comprehension values were highest. One other unanticipated finding was that the top-to-bottom flow direction did not outperform the bottom-to-top flow direction – absolute comprehension values were even slightly lower, although it is mentioned to be a second-best option in standard documents [BU13]. These results also differ from experimental results on flowcharts which indicate that top-to-bottom is the second best option after left-to-right [KR83]. Furthermore, our study found that uncommon flow directions such as bottom-to-top and right-to-left were not more difficult to understand than the conventional left-to-right direction. Right-to-left which is the sharpest contrast to the regular left-to-right reading direction did yield the lowest absolute comprehension values, although this difference was not statistically significant. It is possible though that this difference might be statistically significant with a larger sample size (92% overall comprehension accuracy vs. 79% in right-to-left) in the current sample).

These rather contradictory results concerning obviously uncommon flow directions (in specific, bottom-to-top) may be explained by the fact that when confronted with models, participants might have been especially cautious and also motivated to answer comprehension questions correctly as they perceived the task as a special challenge they wanted to solve. However, the models with the top-to-bottom flow direction lacked the aspect of an unusual challenge that would heighten participants' motivation, thus the cognitive disadvantage of being inconsistent with reading direction weighted stronger and could explain the lower performance of the top-to-bottom group. While any explanation of these unexpected results can only be speculative, it is worth noting that other researchers have found that people adapt surprisingly fast to uncommon reading directions in diagrams [W183]. This is consistent with our results because a fast adaption of the participants to the uncommon reading direction might have resulted in the fact that we could not measure any performance loss for the corresponding flow directions. Further work on this topic could address the extent to which further model complexity of a process model would make the adaption to an uncommon reading direction more difficult. As the models used in the experiments had only included basic symbols to represent the sequence flow, they lacked complexity of models which model additional aspects such as message flows.

Moreover, other explanations for the result that left-to-right did not statistically outperform all other reading directions are possible. Empirical evidence has demonstrated that for reading tasks a left-to-right and a top-to-bottom bias exists in human attention [SH05]. The focus of attention is constantly shifted to the right/bottom while reading and the probability to search for information is higher for the direction of reading than to return to a previously scanned part. This "inhibition of return" bias is larger if the starting point for reading is presented on the left-hand side rather than on the right-hand side [SH05]. Thus, in the context of modeling this could mean that, compared to other directions, in the left-to-right flow direction, with a starting point on the left, people are less likely to move their attention backwards even in the case of a loop. This might lead to lower performance in understanding loops in models drawn from left-to-right and outweigh positive effects of familiar flow direction. Further research would be needed to validate if this explanation holds true though.

Because our experiment investigated BPMN models we also like to discuss an aspect concerning the generalizability for other process modeling notations. While we do believe that BPMN models are representative in terms of general visual characteristics of process models, a specific limitation to generalizability needs to be noted: BPMN XOR and AND routing symbols are constructed symmetrically. Results might differ if routing symbols are sensitive to rotation (as for instance in the UML, where AND is represented by a narrow rectangle (bar)) and would be presented from another angle when changing flow direction.

7 Limitations

As this paper presented a pilot experiment, a main limitation regarding statistical conclusion validity is the low sample size. We did not collect the suggested 20 observations per cell [SNS11] and also could not verify whether distribution assumptions of ANCOVA were met because of the low cell sizes. Therefore, the reported results must be interpreted with caution and it is too early to provide proof to contradict prior research.

In our data we noticed a ceiling effect as the comprehension scores piled up in the end of the scale. Such a restriction of range is a common threat to statistical conclusion validity.

One further source of weakness of this study is the selection of subjects. We recognize that the fact that our sample was drawn from information systems students with basic modeling experience might limit external validity. We do not know whether results can be generalized to experts in process modeling. In particular, it might be easier or harder for experts to adapt to uncommon flow directions. However, we believe that choosing a sample of students who were not biased by a high amount of prior exposure to a specific flow direction was consistent with the goals of the study to investigate the basic usefulness of different flow directions for modeling beginners.

8 Directions for Future Research

Further investigation and experimentation into flow direction of process models is strongly recommended. First, the presented pilot experiment needs to be replicated in form of a large-scale experiment with a higher sample size before the association between reading direction and process model comprehension is more clearly understood.

Second, it would be interesting to investigate not only consistent flow directions as done in this experiment, but also mixtures and changes of flow directions in the same process model. In practice, it can sometimes be noticed that people create “zigzag models” for instance in order to avoid the need for scrolling in a modeling editor or to fit a model to a specific paper format without having to reduce the overall size of model elements and labels. Moreover, right-to-left direction is often used in the context of loops; top-to-bottom and bottom-to-top are used when connecting tasks from different (swim)lanes. Thus, uncommon flow directions as right-to-left or bottom-to-top are in general not primarily used for a process model, but occur in practice in the context of directional changes in a model. We encourage future research to explore various forms of combinations of flow directions in models.

Third, further research might explore flow direction in the context of cultural differences. As reading directions differ across written languages, results might be different in other cultural areas.

Fourth, future research could address whether long experience with a modeling notation would lead to problems if a diagram is presented in an uncommon flow direction. Different notations often are connected to preferred flow directions. For example, UML activity diagrams and BPMN models are often seen with a left-to-right flow direction, while Event-driven Process Chains [SE00] are seen more often modeled from top-to-bottom.

9 Conclusion

To the best of our knowledge, the experiment reported in this paper is the first to investigate the effects of flow direction on process model comprehension. The findings from this pilot study serve as a valuable, first contribution to existing findings on process model layout and have implications for both process modeling practice and research. Moreover, the results have implications on secondary notation research in general. Our pilot study has been unable to empirically confirm a superiority of the left-to-right flow direction to other flow directions with respect to model comprehension, but we also found no negative effects of the left-to-right flow direction. Concerning the top-to-bottom flow direction, our preliminary results do not support a strong recommendation. However, a follow-up experiment with a larger sample size is needed to provide more definitive evidence.

Our findings support retaining existing modeling conventions suggesting left-to-right flow direction. From a theoretical perspective, we believe that advising left-to-right flow direction is beneficial.

References

- [EF11] Effinger, P.: Layout Patterns with Bpmn Semantics. In (Dijkman, R., Hofstetter, J., Koehler, J.): Business Process Model and Notation, Lecture Notes in Business Information, Springer Berlin, p. 130-135, 2011.
- [EJS11] Effinger, P., Jogsch, N., Seiz, S.: On a Study of Layout Aesthetics for Business Process Models Using Bpmn. In (Mendling, J., et al.): Business Process Modeling Notation. vol. 67, Springer Berlin Heidelberg, pp. 31-45, 2011.
- [ESK09] Effinger, P., Siebenhaller, M., Kaufmann, M., An Interactive Layout Tool for Bpmn. In: Proc. Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing, IEEE Computer Society, pp. 399-406, 2009.
- [ES09] Eichelberger, H., Schmid, K.: Guidelines on the Aesthetic Quality of Uml Class Diagrams. Information and Software Technology, 51, pp. 1686-1698, 2009.
- [FA07] Faul, F., Erdfelder, E., Lang, A.-G., Axel, B.: G*Power 3: A Flexible Statistical Power Analysis for the Social, Behavioral, and Biomedical Sciences. Behavior Research Methods 39, pp. 175-191, 2007.
- [FKK13] Figl, K., Koschmider, A., Kriglstein, S.: Visualising Process Model Hierarchies. In:

- Proc. ECIS (European Conference on Information Systems), 2013.
- [FMS13] Figl, K., Mendling, J., Strembeck, M.: The Influence of Notational Deficiencies on Process Model Comprehension. *Journal of the Association for Information Systems*, 14, pp. 312-338, 2013.
- [FRM13] Figl, K., Recker, J., Mendling, J.: A Study on the Effects of Routing Symbol Design on Process Model Comprehension. *Decision Support Systems*, 54, pp. 1104-1118, 2013.
- [FS14] Figl, K., Strembeck, M.: On the Importance of Flow Direction in Business Process Models. In: Proc. 9th International Conference on Software Engineering and Applications (ICSOFT-EA), SCITEPRESS, Vienna, Austria, 2014.
- [GHA10] Genon, N., Heymans, P., Amyot, D.: Analysing the Cognitive Effectiveness of the Bpmn 2.0 Visual Syntax. In: Proc. Software Language Engineering, Lecture Notes in Computer Science, pp. 377-396, 2010.
- [HD68] Handel, S., Desoto, C. B., London, M.: Reasoning and Spatial Representations. *Journal of Verbal Learning and Verbal Behavior*, 7, pp. 351-357, 1968.
- [KI09] Kitzmann, I., et al.: A Simple Algorithm for Automatic Layout of Bpmn Processes. In: Proc. Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on, IEEE, pp. 391-398, 2009.
- [KR83] Krohn, G. S.: Flowcharts Used for Procedural Instructions. *Human Factors*, 25, pp. 573-581, 1983.
- [LA11] La Rosa, M., et al.: Managing Process Model Complexity Via Concrete Syntax Modifications. *Industrial Informatics, IEEE Transactions on*, 7, pp. 255-265, 2011.
- [MP07] Maes, A., Poels, G.: Evaluating Quality of Conceptual Modelling Scripts Based on User Perceptions. *Data & Knowledge Engineering*, 63, pp. 701-724, 2007.
- [MSR12] Mendling, J., Strembeck, M., Recker, J.: Factors of Process Model Comprehension — Findings from a Series of Experiments. *Decision Support Systems*, 53, pp. 195-206, 2012.
- [MO09] Moody, D. L.: The “Physics” of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35, pp. 756-779, 2009.
- [NC99] Nordbotten, J. C., Crosby, M. E.: The Effect of Graphic Style on Data Model Interpretation. *Information Systems Journal*, 9, pp. 139-155, 1999.
- [BU13] Object Management Group, Business Process Model and Notation (Bpmn) Version 2.0.2, ed, 2013.
- [PE06] Petre, M.: Cognitive Dimensions 'Beyond the Notation'. *Journal of Visual Languages & Computing*, 17, pp. 292-301, 2006.
- [RM08] Reijers, H. A., Mendling, J.: Modularity in Process Models: Review and Effects. In (Dumas, M., et al.): *Business Process Management - Bpm 2008*. vol. 5240, ed: Springer, pp. 20-35, 2008.
- [SE00] Scheer, A. W.: *Aris - Business Process Modeling*. 3rd Edition: Springer Verlag, 2000.
- [SH05] Spalek, T. M., Hammad, S.: The Left-to-Right Bias in Inhibition of Return Is Due to the Direction of Reading. *Psychological Science*, 16, Wiley-Blackwell, pp. 15-18, 2005.

-
- [ST12] Storrie, H.: On the Impact of Layout Quality to Understanding Uml Diagrams: Diagram Type and Expertise. In: Proc. Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on, pp. 49-56, 2012.
- [SNS11] Simmons, J.P., Nelson, L.D., Simonsohn, U.: False-positive psychology undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11), Wiley-Blackwell, pp.1359-1366, 2011.
- [TKW91] Tversky, B., Kugelmass, S., Winter, A.: Cross-Cultural and Developmental Trends in Graphic Productions. *Cognitive Psychology*, 23, pp. 515-557, 1991.
- [TV00] Tversky, B., et al., Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. In: Proc. Proceedings of the First International Conference on Theory and Application of Diagrams, Springer-Verlag, pp. 221-230, 2000.
- [WI82] Winn, W.: The Role of Diagrammatic Representation in Learning Sequences, Identification and Classification as a Function of Verbal and Spatial Ability. *Journal of Research in Science Teaching*, 19, pp. 79-89, 1982.
- [WI83] Winn, W.: Perceptual Strategies Used with Flow Diagrams Having Normal and Unanticipated Formats. *Perceptual and Motor Skills*, 57, pp. 751-762, 1983.
- [YU10] Yu, W., et al.: Comparing reading speed for horizontal and vertical English text. *Journal of vision*, 10/2, 2010.

Information System Architecture

An Enhanced Communication Concept for Business Processes¹

Felix Kossak², Verena Geist²

Abstract: Simple communication patterns often do not suffice for modelling the interplay between different business processes. In this paper, we introduce and formally specify an event-based communication concept for business process modelling, constituted by event trigger properties and event pools. We claim that this concept provides a much bigger scope for modelling communication than currently available concepts, particularly when actor and user interaction modelling are included.

1 Introduction

As long as business processes are modelled individually, simple communication patterns typically suffice for communication with an abstractly modelled environment and within the process. However, we often need to model the interplay between different processes as well, often between very heterogeneous systems. Due to growing demand to integrate different processes of different organisations – e.g. in the context of the European “Industry 4.0” initiative –, simple patterns like “Messages” or “Signals” do not always suffice.

Simple communication patterns, such as those provided by the BPMN 2.0 standard [Ob11], have the advantage of being relatively simple and easy to depict in diagrams with relatively intuitive symbols. But integration of automated processes, human actors (see e.g. [NG13, NC12]), and user interaction (see e.g. [KG12, ADG10]) demands more flexibility and customisation. In this paper, we propose a very general event concept for business process modelling, based on a set of *event properties* as well as *event pools*.

In general, a communication concept is supposed to serve different purposes in the context of business process modelling:

- The environment demands a new process instance to be started.
- A process instance notifies its environment that it has finished, that it needs to abort, and/or that an error has occurred.
- The default workflow may be left as in case of an error or of special circumstances, where e.g. compensation is required.

¹ The research reported in this paper supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

² Software Competence Center Hagenberg GmbH (SCCH), Softwarepark 21, 4232 Hagenberg im Mühlkreis, Austria, {felix.kossak,verena.geist}@scch.at

- Different processes need to synchronise, or a process with its environment.
- Data need to be exchanged.

We claim that the proposed communication concept provides a much bigger scope for serving all these purposes than currently available concepts, which we review in Section 5. This review shows that what is obviously missing is a *general* concept for covering different ways of communication in BPM which is *concrete* enough to be integrated in tools in a straightforward way. This is what our paper aims to contribute.

The notion of an “event” has been used ambiguously in the literature, including the BPMN 2.0 standard, where it typically (though not always) denotes a flow node rather than something which happens at a certain point in time. Therefore we need to introduce a clear notation. We try to stay close to the well-established BPMN 2.0 standard, whose basic knowledge we assume. To summarise our notation:

An *event node* either throws or catches *triggers* at certain points in time. Every trigger is of one particular *trigger type* (e.g. “Message” or “Error”), and an event node can throw or catch triggers of one or several different trigger types, which are defined in its *event definitions*.

In the following, we will use Abstract State Machine (ASM) notation [BS03] to formalise our concepts. We expect that the notation provides an intuitive understanding without familiarity with the formal semantics of ASMs. The reader may look at it as pseudocode, with special attention to understandability. Yet this notation makes the semantics of the concepts unambiguous and the complete ASM-based specification, of which we show the important parts, can be easily and provably correct refined towards software code.

The specification is embedded in a comprehensive business process modelling concept, the *Hagenberg Business Process Modelling Method (H-BPM)*, whose core is largely based on BPMN 2.0. A formal specification of the core model has been published in [Ko14], which also demonstrates the requirements for the communication concept. The whole method, including actor, user interaction, and data modelling is presented in [Sc15].

This paper is structured as follows. In Section 2, we introduce a set of event trigger properties to generalise trigger types, opening a wide scope of customised communication. In Section 3, an event pool concept is presented for flexible distribution of triggers, enhancing also flexibility and comfort for human participants. We evaluate the combination of both concepts in Section 4, where we also look at standard trigger types again. Section 5 reviews related work and Section 6 gives a summary.

2 Trigger Properties

Trigger types, such as those provided by BPMN, can be distinguished by different purposes, as their usual names (such as “Message”) suggest. However, more generally, most

of them can also be distinguished by different properties, mostly properties concerning the distribution of triggers, answering questions like the following (cf. [A110]):

- Is the trigger intended for a particular process or (potentially) for different processes (like a fire alert for employees of different companies in a single office building)?
- Is the trigger intended for a particular process *instance*, e.g. for a particular business case, or may it be caught by any instance of a given process (like a call to a help desk may be answered by *any* employee concerned)?
- If there is no particular recipient (no particular process instance) addressed, is it sufficient that *just one* process and process instance deals with the trigger or should more or even all of the potential recipients react (as in the case of a fire alert)?
- Is it obligatory that someone deals with the trigger (as in the case of a help desk call), or is it optional (as in “There is a special offer today in the canteen”)?
- Can the trigger only be caught instantly, or is it valid for some time (or indefinitely)?

When we define respective properties for triggers, we can use them to identify e.g. a signal as a trigger which should be broadcast to all processes (of a given set) and to all process instances and which should be sustained even when some actor has already reacted to it.

However, all the possible combinations of possible values for different properties (reduced by possible constraints) offer more scope than a small set of standard trigger types. We now introduce a set of key properties for triggers and necessary constraints on them.

We need to consider a system in which several communicating processes are running concurrently. Thus we need to identify, for any trigger, one or more recipient processes.

```
shared recipientProcesses : triggers → Set
```

recipientProcesses is a function from the set of triggers to the power set of *processes*. It is a **shared** function, which means that both the process considered and its environment can set the value of this function for a particular trigger. The given process needs to set the value for triggers which it throws (sends), while the environment (including other processes) need to set the value for triggers which the process in question shall receive.

We stipulate that if no recipient process is identified, all processes running on or visible to the workflow engine shall receive the trigger, except if a particular public event pool is specified (see below).

Next, we may want to identify a particular event node of the target process. For instance, there may be alternative start nodes and the environment wants to determine where exactly the new process instance shall start. If no *recipientNode* is specified, any suitable event node within the target process may catch the trigger (if no further constraints apply).

```
shared recipientNode : triggers → flowNodes
```

There are dependencies between *recipientProcesses* and *recipientNode*. For a start, if no particular recipient process is identified, or more than one process is identified, then we cannot specify a particular event node:

```
assert
  forall trigger ∈ triggers holds
    if recipientProcesses(trigger) = undef or
       | recipientProcesses(trigger) | ≠ 1 then
       recipientNode(trigger) = undef
```

Above, the keyword **undef** denotes “undefined” and the vertical bars around “recipientProcesses(trigger)” denote the cardinality of this set of processes.

If a *recipientNode* is defined, then the only member of *recipientProcesses* must be the parent of the *recipientNode*. Thereby we also make sure that only event nodes that are direct children of a given process can be addressed (for propagation into sub-processes, see further below).

```
assert
  forall trigger ∈ triggers holds
    if recipientNode(trigger) ≠ undef then
      forall process in recipientProcesses(trigger) holds
        process = parentNode(recipientNode(trigger))
```

Note that in combination with the previous assertion, we can derive that when *recipientNode* is defined, then *recipientProcesses* must be defined as well and the cardinality of *recipientProcesses* must be 1; thus **forall**, above, actually ranges over a single process.

If a recipient process is specified but no particular recipient node, then we shall be able to specify whether the trigger may be *propagated* into sub-processes (recursively). This corresponds to the distinction between the two concepts of *direct resolution* and *propagation* in BPMN ([Ob11, p. 234f]).

```
shared maybePropagated : triggers → Boolean
```

If *recipientNode* is specified, then propagation is obviously not desired:

```
assert
  forall trigger ∈ triggers holds
    if recipientNode(trigger) ≠ undef then
      maybePropagated(trigger) = false
```

We may also want to address a particular process *instance*. E.g. when a customer has placed an order and subsequently asks when they can expect delivery, then this request must be linked with the proper process instance associated with the respective order number. An order number is an example of *correlation information*. In general, this can be any piece of information through which a particular process instance can be identified. To make correlation possible, the same *correlationInfo* must be shared by the respective properties of both process instance and trigger. (The term “correlation information” is also used in BPMN. Also compare with the *correlation sets* of WS-BPEL [OA07, Sect. 9].)

As we do not want to restrict the form of correlation information, we define an own universe (data type), “*correlationInfo*”, whose implementation is left open. We re-use the name for the respective properties of both triggers and process instances. Note that *correlationInfo* of instances is **controlled**, which means that this property can only be set within the process in question, i.e. by the process engine.

```
shared correlationInfo : triggers → correlationInfo
controlled correlationInfo : instances → correlationInfo
```

In the next section, we will introduce event pools (represented by the universe *eventPools*), some of which may be directly addressed by a trigger.

```
shared recipientPool : triggers → eventPools
```

Another important trigger property shall indicate whether it suffices that one actor reacts to it or not. In other words, shall the trigger be deleted once it has been caught by some event node or shall it be sustained so others can catch it as well?

```
shared deleteUponCatch : triggers → Boolean
```

Next, we want to specify whether a trigger is supposed to be caught instantaneously or if it shall be sustained for some time, and if so, for how long. There are at least three possible ways to define a *timeout*:

- in terms of an absolute point in time (“until 1 Feb 2015, 15:00”);
- in terms of a time span from the creation of the trigger; or
- in terms of a particular hour, day of the week, week, etc. after the creation of the trigger (“until the following Friday, 14:00”).

More exotic variants are imaginable, but we think that *at least* those should be supported, requiring the following properties:

- The first variant requires a simple time property, *timeout*.
- The second variant requires a duration, *lifetime*, in combination with a *timestamp* of the time of creation of the trigger.
- The third variant also requires a *timestamp*, along with a “semi-relative” time property, allowing for values like “the 5th of the following month”, “November of the same year”, etc., for which we use an abstract universe, *RelativeTime*; we call the respective trigger property *relativeTimeout*.

```
shared timestamp : triggers → Time
shared timeout : triggers → Time
```

```

shared lifetime : triggers → Time

shared relativeTimeout : triggers → RelativeTime

```

lifetime and *relativeTimeout* require a *timestamp*.

```

assert
  forall trigger ∈ triggers holds
    if lifetime(trigger) ≠ undef or
      relativeTimeout(trigger) ≠ undef then
        timestamp(trigger) ≠ undef

```

Furthermore, at most one of the functions *timeout*, *lifetime*, and *relativeTimeout* may be defined for a particular trigger.

If neither *timeout* nor *lifetime* nor *relativeTimeout* are defined, then either the process engine has defined a default lifetime which will come into effect or the trigger does not expire as long as any potential recipient process is running.

Finally, in many cases, the process that sent a given trigger is of interest. For instance, we would like to know which process sent an “Error” or “Escalation” trigger. Even the throwing event node may be of interest, and as the process can be derived from that, we define the *senderNode* as a trigger property. (Note that the sender instance can be derived from the sender process in combination with *correlationInfo*.)

```

shared senderNode : triggers → flowNodes

```

Additionally, we retain the property *triggerType* (as in BPMN, with values like “Message”, “Signal”, “Error”, etc.) for the following reasons:

- The BPMN trigger types “Signal”, “Error”, and “Escalation” cannot be distinguished by the other properties, yet “Error” and “Escalation” have algorithmic significance for the workflow.
- The relatively small number of trigger types defined by BPMN, reflecting the most common communication needs, can be represented by symbols which are relatively easy to identify and to remember and render a diagram much easier to understand.
- We want to remain compatible with the BPMN standard as far as possible.

```

shared triggerType : triggers → eventTriggerTypes

```

However, there is a certain redundancy of information shared between the *triggerType* and other properties, and we must assure consistency. We will discuss the respective relations further below.

For the following considerations, we further stipulate that triggers must be uniquely identifiable and that duplication always leads to *different* triggers.

3 Event Pools

If we want to enable users to choose in which order to process messages (and possibly other event triggers), we have to give them a kind of “pool” into which event triggers are delivered and from which users can pick. This concept is already well established in the form of the “inbox” of an email client. The pool concept we are going to introduce is also influenced by that proposed for S-BPM (see [F112]); S-BPM lays a special focus on the viewpoint of actors (or “subjects”).

We not only want users to be able to choose the order in which to process triggers but also to be able to opt-in for additional, non-obligatory trigger sources, like certain kinds of news (like RSS feeds). This can be enabled by giving users access to certain additional event trigger pools, i.e. pools not directly associated with a particular process.

Furthermore, there are certain kinds of event triggers, like signal, error, or compensation triggers, which may be supposed to be caught by more than one process or sub-process. One way to handle this is to duplicate such events for every potential recipient. Another possibility, at least for the conceptual level, is to deposit such a trigger in a pool which is not associated with a particular process but which is “public”.

So a process might have access to different event pools, some private, some public. However, a user might want to have a single view on all the relevant pools. To this end, we can define a view on all the triggers from all the pools relevant for a particular process by means of a virtual pool which we call the process’s *inbox*. For the abstraction of the throwing of triggers, we further define an *outbox* for each process.

In summary, the event pool concept we are proposing comprises the following pool types:

- a *private* event pool for each process or sub-process for triggers which are only visible for event nodes that are directly within this (sub-)process (this corresponds to “direct resolution” in BPMN);
- a *group* event pool for each (sub-)process for triggers which are visible also within sub-processes of this (sub-)process, recursively, to enable propagation;
- *public* event pools to which processes can subscribe or to which several processes can be mandatorily subscribed (by the process designer);
- a virtual *inbox* for each (sub-)process to provide a single view on all relevant pools; and
- an abstract *outbox* for each (sub-)process to hide the details of delivering triggers thrown within this (sub-)process in accordance with the triggers’ properties.

Within private and group event pools, triggers for a particular process instance can be identified by correlation information.

We now introduce event pools in more detail. We assume a universe (data type) *eventPools*, on which the rules (algorithmic functions) *AddTrigger* and *RemoveTrigger* as well as a derived function (derived property) *containsTrigger* are defined.

An event pool may or may not be associated with a particular (sub-)process, i.e. an *owner-Process*. A public event pool is associated with the *environment* instead. We also assume that the *environment* has a pool for receiving triggers.

For the sake of simplicity, we assume that there is a fixed number of event pools with fixed associations in a given run of a process engine. Consequently, we model the function *ownerProcess* as *static* (i.e. it cannot change during runtime).

```
static ownerProcess : eventPools → processes ∪ { environment }
```

A derived function can identify all event pools owned by a particular (sub-)process or by the environment:

```
derived eventPools : processes ∪ { environment } → Set
eventPools(process) =
  { pool | pool ∈ eventPools and ownerProcess(pool) = process }
```

An event pool associated with a particular (sub-)process may be *private*; else, it is a group event pool. If an event pool associated with the environment is *private*, it is supposed to receive triggers addressed to the environment. If an event pool associated with the environment is *not private*, it is a public event pool.

```
static private : eventPools → Boolean
```

We can then define:

- a *private event pool* as a pool with *ownerProcess(pool) ∈ processes* and *private(pool) = true*;
- a *group event pool* as a pool with *ownerProcess(pool) ∈ processes* and *private(pool) = false*;
- a *public event pool* as a pool with *ownerProcess(pool) = environment* and *private(pool) = false*; and
- the environment's event pool (for triggers addressed to the environment) as a pool with *ownerProcess(pool) = environment* and *private(pool) = true*.

We define a *default public event pool* which is visible for all processes and to which e.g. signals can be distributed if their destination is not further specified:

```
static defaultPublicEventPool : → eventPools

assert
  ownerProcess(defaultPublicEventPool) = environment and
  private(defaultPublicEventPool) = false
```

Additional public event pools may be defined by the business process designer.

We may want event pools to have further properties such as access rights, but we do not consider more properties in this place.

We assert that every process has exactly one private event pool and one group event pool. The environment has one unique private event pool.

We can now identify the unique event pools of a given process by derived functions:

```
derived privateEventPool : processes → eventPools
privateEventPool(process) =
  choose pool in eventPools(process) with private(pool) = true do
    return pool

derived groupEventPool : processes → eventPools
choose pool in eventPools(process) with private(pool) = false do
  return pool
```

The *visiblePublicEventPools* of a process are all the public event pools to which the process in question has, or has been, subscribed:

```
monitored visiblePublicEventPools : processes → Set
```

The *defaultPublicEventPool* must be visible for all processes:

```
assert
  forall process ∈ processes holds
    defaultPublicEventPool ∈ visiblePublicEventPools(process)
```

The *visibleEventPools* of a process are then the *visiblePublicEventPools* plus the private and group event pools.

```
derived visibleEventPools : processes → Set
visibleEventPools(process) =
  eventPools(process) ∪ visiblePublicEventPools(process)
```

We can now define the *inbox* of a process as a view showing all triggers available in any of the *visibleEventPools*.

```
derived inbox : processes → Set
inbox(process) =
  { trigger | forsome pool ∈ visibleEventPools(process) holds
    containsTrigger(pool, trigger) }
```

From a process's viewpoint, for throwing a trigger it shall suffice to put it into an *outbox*.

```
shared outbox : processes → eventPools
```

We assume that some delivery service will pick triggers up from the *outbox* and distribute them according to their properties. For any *trigger*:

- If *recipientProcess(trigger)* is **undef** or empty, then the *trigger* shall be delivered to a public event pool; if additionally *recipientPool(trigger)* = **undef**, then the *trigger* shall be delivered to the *defaultPublicEventPool*.
- If there is some *process* in *recipientProcesses(trigger)* and *mayBePropagated(trigger)* = **true**, then the *trigger* shall be delivered to the group event pool of each specified process.
- If there is some *process* in *recipientProcesses(trigger)* and *mayBePropagated(trigger)* = **false**, then the *trigger* shall be delivered to the private event pool of each specified process.
- If *environment* \in *recipientProcesses(trigger)*, then the *trigger* shall (also) be delivered to the environment's (private) event pool.

When a particular process instance has reacted to a trigger in a public event pool, we set a controlled function *hasBeenCaughtByInstance* to true so that the instance will not react twice. The function value is **false** by default and set to **true** once the instance in question has reacted. Note that the process in question can always be identified via the instance.

controlled *hasBeenCaughtByInstance* : triggers \times instances \rightarrow Boolean

This concludes an outline of the major features of the proposed enhanced communication concept for business processes.

4 The Scope of Possible Communication and Standard Trigger Types

We now evaluate the scope of communication which the proposed concept enables. We start with a comparison with the BPMN standard, which describes “different strategies to forward the *trigger* to catching **Events**: publication, direct resolution, propagation, *cancellations*, and *compensations*” [Ob11, p. 234]. Cancellation and compensation do not actually constitute different ways of delivering triggers, but the actual delivery strategies can be handled by our proposal:

- **Publication** within a process can be achieved by specifying a recipient process of the trigger and leaving the recipient node undefined; publication across processes can be achieved by specifying a public event pool as the recipient pool.
- **Direct resolution** can be achieved by specifying a recipient node.
- **Propagation** can be achieved by setting *mayBePropagated* to **true**.

Aldred defines “process integration patterns” [Al10], many of which are relevant for our concept. Aldred distinguishes the following “dimensions” of communication:

- **Participants**: 1–1, 1–many, or many–many; the first two can be covered by setting *deleteUponCatch* to **true** for 1–1 and **false** for 1–many, and also by choosing a

suitable event pool, e.g. a public event pool for 1–many. The case of many–many can be handled by allowing different senders to send triggers to a particular public event pool (with *deleteUponCatch* set to **false**).

- **Uni-directional / bi-directional:** this is a matter of process design (though a public event pool could aid in bi-directional communication).
- **Synchronous / asynchronous:** this can be supported via the *timeout / lifetime* properties of triggers.
- **Thread-coupling:** this is a matter of process design.
- **Time** (whether two participants need to both be participating in an interaction at the exact same moment): this can be supported via the *lifetime* property, which is set to zero (or a minimum) for immediate communication.
- **Direct / indirect contact:** indirect contact between communication partners that need not know each other can be supported by public event pools.
- **Duplication:** in our concept, duplication *can* (but need not) be replaced by setting the trigger property *deleteUponCatch* to **false** and possibly using a public event pool.

Patterns of process instantiation, however, as e.g. discussed in [DM09], are a matter of process design and not of trigger design.

So it turns out that our concept covers a wide range of communication patterns.

“Standard” event trigger types as defined by the BPMN standard can be matched to particular settings of trigger properties as proposed here:

- A **Message** trigger has a single recipient process, *deleteUponCatch* is **true**, and there is no timeout.
- A **Signal** trigger has *deleteUponCatch* set to **false** and *maybePropagated* is **true**.
- An **Error** trigger is in effect a special-purpose **Signal** trigger. The same holds for an **Escalation** trigger.
- A **Cancel** trigger has defined *recipientProcesses*, *maybePropagated* is **true**, *deleteUponCatch* is **false**, and timeout is minimal.
- A **Compensation** trigger and a **Terminate** trigger have the same properties as a **Cancel** trigger (except the *triggerType*).

(Note that we do not consider **Link** triggers as they do not actually serve communication.)

5 Related Work

We have already commented on BPMN [Ob11] and on the “process integration patterns” of Aldred [Al10] in the context of YAWL in the previous section. Some of the “dimen-

sions” of communication discussed by Aldred concern process design rather than pure communication mechanisms. Moreover, Aldred’s patterns are not translated into formal mechanisms which can be straightforward integrated in tools.

More generally, the Workflow Patterns of van der Aalst, ter Hofstede, et al. [vdAtH] (on which YAWL is based) address various perspectives relevant for event handling. Regarding the control-flow patterns, events are in particular involved in *implicit termination*, *deferred choice*, and in several *cancellation* patterns. There are also two patterns that explicitly describe the notion of *triggers*. However, support for external data interaction patterns and for triggering work execution (see *auto-start* patterns) is limited.

By adopting the concept of YAWL, Mendling et al. [MNN05] define an extension to EPC to enhance support for workflow patterns. They introduce e.g. cancellation areas to support *cancellation* patterns. However, the focus of EPC is on semi-formal process documentation rather than formal process specification.

The event pools of S-BPM [FI12] provided inspiration for the pool concept introduced here. The pools of S-BPM are tailored for actor comfort, but are not embedded in a wider delivery concept. S-BPM provides some extra “configuration parameters” for input pools, whose most important application appears to be the enforcement of synchronous communication, which is handled differently in our more general concept.

WS-BPEL [OA07] supports correlation, propagation, and definition of timeouts (by *message* and *alarm* events); however, it shows deficiencies regarding the generality of specifying event handlers and event consumption.

Lucchi and Mazzara [LM07] propose a framework for generic event and error handling in business processes by reducing the amount of different mechanisms for exception, event, and compensation handling in WS-BPEL to a single mechanism based on the idea of event notification. The resulting specification helps simplify BPEL models and implementations in the area of Web services orchestration similar to our improvements for BPM.

Common event-driven patterns are presented by Etzion and Niblett in [EN11]. The authors regard BPM as a related technology to event processing and reflect current trends, e.g. event-driven architecture and asynchronous BPM, and future directions. They propose basic and dimensional patterns including common temporal patterns as we do. There are also certain parallels concerning pattern policies, e.g. consumption or cardinality policies.

A set of service interaction patterns is proposed by Weske in [We12]. The patterns primarily apply to the service composition layer; however, an issue common to our proposed concept is the classification according to the number of involved participants.

Herzberg et al. [HMW13] introduce so-called *process events* that enrich events that occur during process execution with context data to create events correlated to the proper process descriptions. They address correlation as a main issue of their event processing platform. However, they concentrate on business process monitoring and analysis rather than modelling.

The WED-flow approach of Ferreira et al. [Fe10] proposes data states to integrate event processing into workflow management systems. Data states store required information for event-handling, thereby increasing backward and forward recovery options. In contrast to our work, the WED-flow approach does not define control flow but triggers over attribute values (wed-states), yielding the flow as a consequence of satisfied conditions.

The Complex Event Processing (CEP) discipline [Lu02], an emerging technology dealing with event-driven behaviour, and its combination with BPM is a main topic of interest [BDG07] and used e.g. for *Event-Driven Business Process Management* [Am09] to detect and react to possible errors within processes and also to support dynamic business process adaptation [HSD10] or business process exception management [Li14].

6 Summary

We introduced a communication concept for advanced business process modelling which enables modelling of a wide range of different communication styles. We showed how different communication patterns can be modelled by a combination of a set of event trigger properties and a few different types of event pools. Event pools make it also possible to model flexibility for human actors, such as the ability to process messages in a custom order or to subscribe to optional communication sources (such as news).

We have compared our communication concept in particular with BPMN as well as with the patterns introduced by Aldred [Al10]. We think it is obvious that our concept is much more general as that of BPMN-style modelling languages and is able to meet all relevant communication needs identified by Aldred.

The communication concept we have proposed is part of an overall BPM method developed at the Software Competence Center Hagenberg, Austria, which we call the *Hagenberg Business Process Modelling Method (H-BPM)*; it is outlined in [Sc15].

Acknowledgement: This publication was supported by the *AdaBPM* project, which is funded by the FFG under the project number 842437.

References

- [ADG10] Atkinson, C.; Draheim, D.; Geist, V.: Typed business process specification. In: EDOC'10. IEEE Computer Society, pp. 69–78, 2010.
- [Al10] Aldred, L.: Process integration. In (ter Hofstede, A. M.; van der Aalst, W. M. P.; Adams, M.; Russell, N., eds): *Modern Business Process Automation: YAWL and its Support Environment*, pp. 489–511. Springer, Heidelberg, 2010.
- [Am09] von Ammon, R.; Emmersberger, C.; Ertlmaier, T.; Etzion, O.; Paulus, T.; Springer, F.: Existing and future standards for event-driven business process management. In: *Proc. of the 3rd ACM Int. Conf. on Distributed Event-Based Systems*. ACM, pp. 24:1–24:5, 2009.

- [BDG07] Barros, A.; Decker, G.; Grosskopf, A.: Complex events in business processes. In: *Business Information Systems*. Springer, pp. 29–40, 2007.
- [BS03] Börger, E.; Stärk, R.: *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer, Berlin, Heidelberg, 2003.
- [DM09] Decker, G.; Mendling, J.: Process instantiation. *Data & Knowledge Engineering*, 68(9):777–792, 2009.
- [EN11] Etzion, O.; Niblett, P.: *Event Processing in Action*. Manning Publications, 2011.
- [Fe10] Ferreira, J.; Wu, Q.; Malkowski, S.; Pu, C.: Towards flexible event-handling in workflows through data states. In: *SERVICES-1*. IEEE, pp. 344–351, 2010.
- [Fl12] Fleischmann, A.; Schmidt, W.; Stary, C.; Obermeier, S.; Börger, E.: *Subject-Oriented Business Process Management*. Springer, Berlin, Heidelberg, 2012.
- [HMW13] Herzberg, N.; Meyer, A.; Weske, M.: An event processing platform for business process management. In: *Proc. of the 2013 17th IEEE Int. Enterprise Distributed Object Computing Conf.* IEEE, pp. 107–116, 2013.
- [HSD10] Hermosillo, G.; Seinturier, L.; Duchien, L.: Using complex event processing for dynamic business process adaptation. In: *Proc. of the 2010 IEEE Int. Conf. on Services Computing*. IEEE, pp. 466–473, 2010.
- [KG12] Kopetzky, T.; Geist, V.: Workflow charts and their precise semantics using abstract state machines. In: *EMISA. LNI. Gesellschaft für Informatik e.V.*, pp. 11–24, 2012.
- [Ko14] Kossak, F.; Illibauer, C.; Geist, V.; Kubovy, J.; Natschläger, C.; Ziebermayr, T.; Kopetzky, T.; Freudenthaler, B.; Schewe, K.-D.: *A Rigorous Semantics for BPMN 2.0 Process Diagrams*. Springer, 2014.
- [Li14] Linden, I.; Derbali, M.; Schwanen, G.; Jacquet, J.-M.; Ramdoyal, R.; Ponsard, C.: Supporting business process exception management by dynamically building processes using the BEM framework. In: *Decision Support Systems III*, volume 184 of LNBIP, pp. 67–78. Springer International Publishing, 2014.
- [LM07] Lucchi, R.; Mazzara, M.: A pi-calculus based semantics for WS-BPEL. *The Journal of Logic and Algebraic Programming*, 70(1):96 – 118, 2007.
- [Lu02] Luckham, D.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional, 2002.
- [MNN05] Mendling, J.; Neumann, G.; Nüttgens, M.: Yet another event-driven process chain. In: *Business Process Management*, pp. 428–433. Springer, 2005.
- [NC12] Natschläger-Carpella, C.: *Extending BPMN with Deontic Logic*. Logos Verlag Berlin, 2012.
- [NG13] Natschläger, C.; Geist, V.: A layered approach for actor modelling in business processes. *Business Process Management Journal*, 19:917–932, 2013.
- [OA07] OASIS: , WS-BPEL 2.0. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>. Accessed 2014-11-03., 2007.
- [Ob11] Object Management Group: , Business Process Model and Notation (BPMN) 2.0. <http://www.omg.org/spec/BPMN/2.0>. Accessed 2014-11-03., 2011.

- [Sc15] Schewe, K.-D.; Geist, V.; Illibauer, C.; Kossak, F.; Natschläger-Carpella, C.; Kopetzky, T.; Kubovy, J.; Freudenthaler, B.; Ziebermayr, T.: Horizontal Business Process Model Integration. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII, pp. 30–52. Springer, 2015.
- [vdAtH] van der Aalst, W.M.P.; ter Hofstede, A.H.M.: , Workflow Patterns Homepage. <http://www.workflowpatterns.com>. Accessed 2015-07-20.
- [We12] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer Science & Business Media, 2012.

Using Content Analysis for Privacy Requirement Extraction and Policy Formalization

Stefanie Rinderle-Ma¹, Zhendong Ma², Bernhard Madlmayr³

Abstract:

Privacy in cyberspace is a major concern nowadays and enterprises are required to comply with existing privacy regulations and ensure a certain level of privacy for societal and user acceptance. Privacy is also a multidisciplinary and mercury concept, which makes it challenging to define clear privacy requirements and policies to facilitate compliance check and enforcement at the technical level. This paper investigates the potential of using knowledge engineering approaches to transform legal documents to actionable business process models through the extraction of privacy requirements and formalization of privacy policies. The paper features two contributions: A literature review of existing privacy engineering approaches shows that semi-automatic support for extracting and modeling privacy policies from textual documents is often missing. A case study applying content analysis to five guideline documents on implementing privacy-preserving video surveillance systems yields promising first results towards a methodology on semi-automatic extraction and formalization of privacy policies using knowledge engineering approaches.

1 Introduction

Privacy in cyberspace has become a major concern nowadays and enterprises are obliged to ensure a certain level of privacy as demanded by law [Bi08] and society [In97]. As a multidisciplinary topic, privacy is influenced by social, legal, and technical factors. The ambiguity of privacy definition, the difference in privacy perception, and the fast changing technological landscape make it very challenging for an enterprise to keep up with the privacy stipulations and expectations.

Since business processes capture activities at both human and system level within an enterprise, they often serve as the basis for privacy checks [AM14], i.e., it can be analyzed how (daily) routines in an enterprise are conducted with respect to privacy requirements and policies. As for security [Le14], business processes can be either checked for their compliance with privacy requirements (privacy ensuring) or they can be used to implement privacy policy (privacy enforcement).

Verification of privacy requirements over a system or business process can be conducted by, for example, model checking. Such approaches require the formal

¹ University of Vienna, Faculty of Computer Science, Austria, stefanie.rinderle-ma@univie.ac.at

² Austrian Institute of Technology, Digital Safety and Security Department, Austria, zhen-dong.ma@ait.ac.at

³ University of Vienna, Faculty of Computer Science, Austria, bernhardmad@gmail.com

representation of privacy requirements as structured privacy policies. However, privacy requirements are often originated from legal documents [BA08], i.e., in natural language and hence in an unstructured way, which is subject to interpretations (e.g. by law professionals) and lacks clarity at the technical level. Often these documents are vague and generic. For example, the clause “to provide adequate privacy protection” might be sufficient for lawyers but way too ambiguous for system designers and engineers to implement. Therefore, engineers have great difficulties to understand and interpret such documents and translate them into practical technical privacy-preserving designs and practices. A recent multi-disciplinary approach to address privacy in surveillance systems found out the main difficulty in designing privacy-preserving systems is the ambiguity from the knowledge gap between technical and non-technical world [Ma14]. Hence, the ability to extract the relevant information from privacy documents and provide the extracted information in a structured (formalized) and unambiguous way (i.e. understandable and actionable technical specifications) can be very beneficiary in designing and developing privacy-preserving ICT systems. As extraction of privacy requirements can be tedious and error prone when done manually, it would be useful to employ techniques to at least derive candidates for privacy policies in a semi-automatic way. Here, we advocate the investigation of knowledge engineering techniques such as content analysis [St06] or text mining [AZ12] for their suitability to extract privacy requirements from legal documents in a semi-automatic way. For clarification of terminology, throughout the paper, we denote as privacy requirements the privacy-related information within the textual documents which are first extracted and then modeled or formalized as privacy policies.

In summary, the paper addresses two questions:

1. How to utilize knowledge engineering techniques for extracting privacy requirements from text in legal documents in a semi-automatic manner?
2. How to model the extracted information as structured privacy policies?

Many approaches have addressed privacy requirement engineering, e.g., [ANM10, BM10, BA08, Ch08, Ch11, Co07, De11, Gr12, Gü05, He03, KBG11, KS85, Le06, LYM03, MdAY14, MPZ05, MMZ11, MMZ08, PDG14, RGK13, Ri14, dRAF05]. However, as it will be shown, most of these approaches are manual or do not consider textual input. In order to underpin this claim and provide an overview of existing privacy requirements engineering approaches, the paper provides a literature review in Sect. 2, guided by the questions: What knowledge engineering technique is used? What are source and target format for privacy requirement engineering? Section 3 presents the results of applying content analysis to five documents for implementing privacy-preserving video surveillance systems. The result is a first suggestion of how knowledge engineering techniques can be utilized for privacy policy extraction and formalization and is presented in Sect. 4. As such, the proposal can be used in almost any of the existing approaches. It also discusses next steps in validation and transferability of the methodology.

2 Literature review

A literature review was conducted in order to obtain an overview of existing approaches for elicitation of privacy requirements. Specifically interesting in the context of this paper are approaches that utilize knowledge engineering techniques. The guidelines for conducting a systematic literature review were taken up in a simplified form from [Ki09].

At first, the following keywords were selected for the horizontal literature search:

`policy engineering, privacy policy engineering, privacy requirements engineering, security policy engineering, security requirements engineering, policy elicitation, privacy policy elicitation, security policy elicitation, privacy requirement elicitation, security requirement elicitation.`

As search method, the keywords were used as title search in google scholar⁴ from 22 – 24 Oct 2014 as well as on 27 Oct 2014, excluding patents and citations. Table ?? shows the results of the horizontal literature search, i.e., the first column contains the keywords and the second column the number of papers found.

<i>Keywords</i>	<i># Hits</i>	<i>Selection words</i>	<i># selections</i>
policy engineering	505	privacy, security	9
privacy policy engineering	3		0 (overlap with policy engineering)
privacy requirement engineering	26	focus: privacy requirements	21 (1 overlap, 1 not available, 1 duplicate)
security policy engineering	11	focus: security policies	0 (overlap with policy engineering and selection criteria)
security requirement engineering	120	64 focus: security requirements	(duplicates, unavailable, journal extension)
policy elicitation	14	privacy	0
privacy / security policy elicitation	0		0
privacy requirement elicitation	1	0	
security requirement elicitation	6		5
overall vertical	686		99

Tab. 1: Results of vertical literature search

Within a primary selection process, each paper title was checked for the covered area. For each keyword, selection words were defined, i.e., those words that specify and restrict the found papers for the specific area of privacy and security policy elicitation. Take, for example, keyword `policy engineering` which results in 505 found papers during the primary search. However, policy engineering might also refer to other policies than privacy and security policies. Hence, the found papers

⁴ scholar.google.com

were scanned through their title and abstract whether or not they refer to the privacy and security area. resulting in 9 papers. On top of these content-related selection criteria, general selection criteria such as availability of paper, written in English, and scientific paper were applied.

The result of the vertical literature search, i.e., a list of the primarily selected number of 99 publications can be found at⁵. The primary literature list was reduced within an expert discussion based on the following criteria: lack of focus on privacy, model-driven approaches, lack of linkage to knowledge / requirements engineering methods. In addition, similar approaches, specifically from the same group of authors on the same topics were aggregated by considering a selection of their papers.

The reduction resulted in 27 papers. Based on these papers, snowballing was conducted, resulting in $27 + 6 + 5 = 38$ papers⁶. In addition, snowballing led to a new keyword, i.e., *extraction* which was combined with keywords *privacy policy* and *privacy requirement* when conducting another round of vertical search. However, the keywords did not yield any results.

These core papers were analyzed along the following research questions:

1. Is a knowledge engineering method suggested / applied? If yes, which ones?
2. Which sources are used?
3. What is the target format?

The first question was used as a reduction criteria, i.e., if an approach was neither proposing nor applying a knowledge engineering method it was excluded from further analysis. Out of the 38 papers, 25 approaches were found during horizontal and vertical search that suggest usage of knowledge engineering method(s): [AM14, AE00, ANM10, BM10, BVA06, BA08, Ch08, Ch11, Co07, De11, Gr12, Gü05, He03, KBG11, KS85, Le06, LYM03, MdAY14, MPZ05, MMZ11, MMZ08, PDG14, RGK13, Ri14, dRAF05]. 4 papers provide an overview of existing security requirements engineering / modeling / elicitation techniques themselves [El11, Fa10, Me10, SK12] and were hence not considered in the further analysis. The remaining 9 papers did not suggest any elicitation method and were hence discarded from further investigation.

With respect to the research questions set out in the introduction, the 25 resulting papers were analyzed whether they (a) employ a manual or (semi-)automatic engineering technique, (b) take text as input format, and (c) produce an output format that can be utilized for business process compliance checking. Results:

1. The only approach (from 1985) that suggests a (semi-)automatic approach is [KS85]. All other approaches propose, extend, or employ manual methods.

⁵ http://cs.univie.ac.at/fileadmin/user_upload/fak_informatik/RG_WST/documents/Rinderle-Ma/PrimarySearch_SEC15_MaRi.pdf

⁶ Again, papers of the same group were considered in an aggregated way, i.e., with the most current or comprehensive paper.

Some of these methods are tool-supported, i.e., PRET [MMZ08] and the method proposed in [Gr12] supported by Objectiver. It is worth taking a look what is exactly supported by tools, the extraction or the modeling or both.

2. Several approaches extract privacy requirements from textual sources, i.e., PRET [MMZ08], [BA08] specifically for HIPAA, [BVA06] in form of Unrestricted Natural Language Statements (UNLR), using Secure Tropos on law by [MPZ05], and specifically analyzing DITSCAP [Le06]. The other approaches range from business process models [AM14] and stakeholder knowledge [Gü05, De11, dRAF05, KBG11, ANM10], to requirements [Ch11, PDG14]. The other approaches remain either unspecific, e.g., by stating “various” information sources or information systems.
3. Regarding the last question of the target format, most approaches provide some structured format, i.e., requirements, policies or rules, patterns, XML, and ontologies. By contrast, [AM14, MdAY14] have text as target format.

Overall, none of the approaches fits the requirements set out in the introduction, i.e., provides a (semi-)automatic methodology for extracting structured privacy requirements from text. Overall, most of the approaches aim at comprehensive methodology for guiding the entire engineering process from identifying relevant documents or other artifacts until privacy policies are specified. In particular, most of the approaches include the users, e.g., domain experts. This is for sure an important issue. This paper does not suggest to replace an overall methodology and inclusion of users, but aims at support of ONE specific step of the overall methodology, i.e., the extraction and formalization step as discussed in the next section.

3 Preliminary study: Content analysis

Methods for the extraction of information from text are proposed and applied in different areas. Knowledge Engineering [SBF98] deals more generally with the construction of Knowledge-based Systems and comprises the extraction of information as one step next to other steps such as modeling and derivation. Information extraction also plays a crucial role in web environments where often (semi-)structured data is the basis to extraction [Sa08]. Specifically geared towards information extraction from text are, for example, text mining [AZ12], qualitative content analysis [St06], and Natural Language Processing (NLP) [Fr11].

The purpose of this preliminary study was to evaluate the suitability of knowledge engineering methods based on the example of content analysis for the extraction of privacy requirements from text or unstructured data such as regulatory documents or laws. Qualitative Content Analysis (QCA) has a manual component as documents must be unitized, categorized, and coded. Support is provided by tools such as QDA Miner⁷ and Atlas.ti⁸. Particular advantages of QCA are reliability and maintain-

⁷ <http://provalisresearch.com/products/qualitative-data-analysis-software/>

⁸ <http://atlasti.com/>

ability. We have gathered positive experience with QCA in deriving the teaching process at the University of Vienna based on interview transcripts [KRM11].

The case study was focused on privacy in video surveillance. As a widely deployed technology for protecting humans and property in public and private spaces, video surveillance has always been a privacy concern and a subject of debate. Moreover, due to technological advancement, video surveillance systems are becoming more powerful and hence more privacy-intrusive, in which multiple information sources can be aggregated and video images can be analyzed automatically in large scales. Due to the privacy concern around video surveillance, a large amount of regulations and guidelines exist. However, similar to many other privacy-related documents, they often lack the clarity and precision that are important for compliance check and system design at the technical level. The case study was based on the following guidelines on implementing privacy-preserving video surveillance systems.

1. *The EDPS Video-Surveillance Guidelines* contains guidelines “for European institutions and bodies on how to design and operate their video-surveillance system”⁹.
2. *OECD Privacy Guidelines* “govern[...] the protection of privacy and transborder flows of personal data”¹⁰.
3. *Guidelines for Public Video Surveillance* provided by an initiative for protecting “civil liberties” in America¹¹.
4. *Data protection and privacy ethical guidelines*¹² address data and privacy issues in the context of EU FP7 projects.
5. *Operational Guidance on taking account of Fundamental Rights in Commission Impact Assessments*¹³ issued by the European Commission.

Due to experience and availability we opted for using QDA Miner. The QCA was conducted by one analyst. In a first round, the analyst read through the above documents and obtained a general overview of the content and the relation between the documents.

As the target format is process-structured, the two basic categories to be extracted from the text are **Actors** and **Activities**. Focusing on **Actors** and **Activities** as a first step corresponds to the idea of analyzing sentences finding verbs and objects as featured in, e.g., Friedrich et al. [Fr11] extracting actors and actions from sentences.

In a second round, the analyst read through the documents again highlighting relevant phrases from the document that fit into those two categories. Examples for

⁹ https://secure.edps.europa.eu/EDPSWEB/webdav/shared/Documents/Supervision/Guidelines/10-03-17_Video-surveillance_Guidelines_EN.pdf

¹⁰ <http://www.oecd.org/sti/ieconomy/oecdguidelinesonthe protectionofprivacyandtransborderflowsofpersonaldata.htm>

¹¹ <http://www.constitutionproject.org/wp-content/uploads/2012/09/54.pdf>

¹² http://ec.europa.eu/research/participants/data/ref/fp7/89827/privacy_en.pdf

¹³ http://ec.europa.eu/justice/fundamental-rights/files/operational-guidance_en.pdf

actors are **Government**, **Child**, and **Employee** and for processes **Impact assessment**, **Monitor Area**, and **Install System**.

Based on evaluating statistics on word frequencies, the documents were coded along the categories **Actors** and **Activities**. The code base for QDA Miner can be found here: http://cs.univie.ac.at/fileadmin/user_upload/fak_informatik/RG_WST/documents/Rinderle-Ma/Privacy.ppj. Figure 1 shows the code book for the five documents. Note that codes abstract from different terms and phrases in the documents. One example is activity **Consultation** which represent, for example, phrase **Consult DPO**. The coding was aggregated and reviewed several times in order to overcome errors and to provide the coding at an adequate abstractions level.

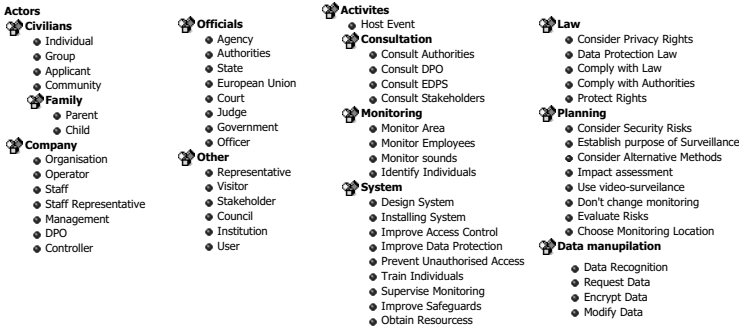


Fig. 1: Coded Actor and Activity Hierarchy (Code Book Produced Using QDA Miner, Optimized Presentation)

Let us first take a look at the **Actors**. Here different categories can be organized into sub-categories, e.g., category **Civilian** has sub-categories **Individuals**, **Group**, **Applicant**, **Community**, and **Family**. The code hierarchies for **Actor** can be transferred and modeled as, for example, organigram in order to connect the organizational information with the processes to be derived. The model shown in Fig. 2 was modeled using Signavio.

Category **Actor** was used during QCA. Organigrams usually offer more meta model elements to capture organizational information such as **Roles**, **Organizational Units**, and **Persons**. Hence, in principle, two design decisions can be made. Either more categories are considered during QCA or the categories that are coded are mapped onto different meta model elements. In this example, the second option was chosen, i.e., category **Actor** was mapped onto **Roles**, **Organizational Units**, and **Persons**. The mapping was done manually.

At the end of this step, an organigram exists that captures the information from all documents and can be directly used in processes that express privacy requirements.

In a second step, the coded activities (cf. Fig. 1) are to be combined into a process model. We gained positive experience with expressing medical guidelines with

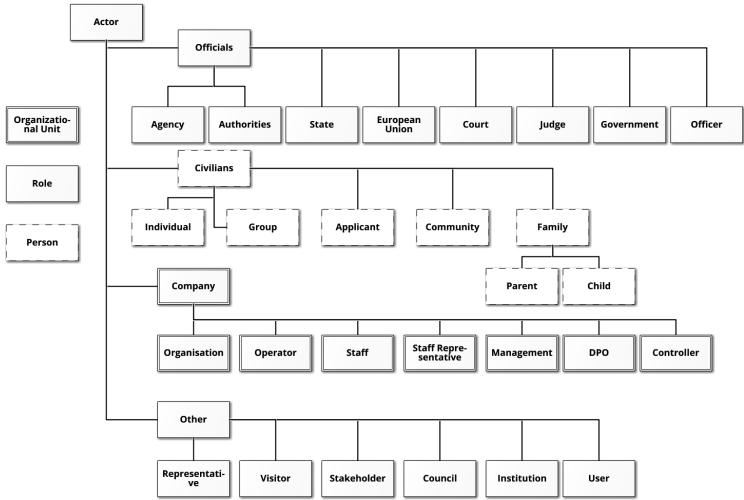


Fig. 2: Transformation into Organigram (Using Signavio)

BPMN, the standard process modeling language [Du11]. Thus, in the following, process models are derived from the code book activities in BPMN.

The identification of which codes belong to the same process model is based on co-occurrences and proximity of codes. Both can be analyzed by comparing overlapping code segments. Co-occurrence, frequency, and proximity can be measured by different indexes, e.g., the Jaccard’s coefficient as for the dendrogram depicted in Fig. 3.

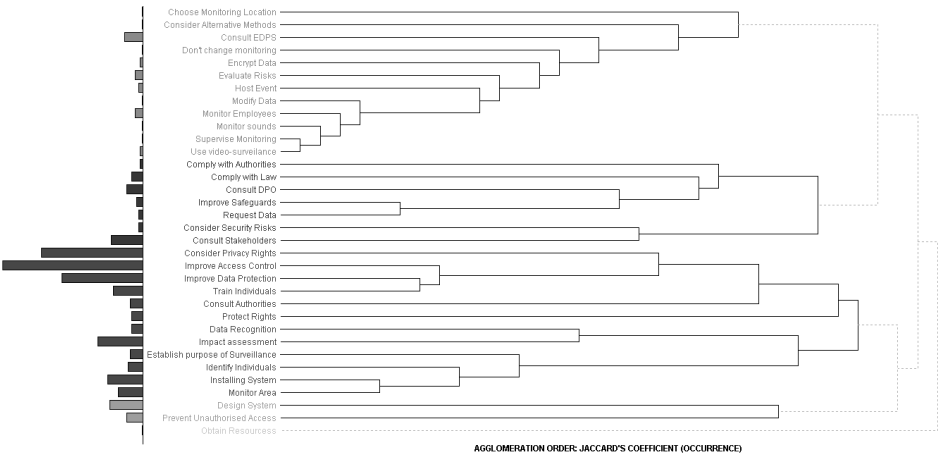


Fig. 3: Dendrogram: Co-Occurrence of Codes (Using QDA Miner)

The dendrogram is produced with 5 clusters by QDA Miner expressed by the color of the bars. One noticeable cluster is the green one where specifically activ-

ities **Consider Privacy Rights**, **Improve Access Control**, and **Improve Data Protection** show a high similarity (degree of co-occurence). This impression is supported by the proximity plot in Fig. 4 for activity **Consider Privacy Rights** which shows the a proximity of 1.0 with activities **Improve Access Control**, **Improve Data Protection**, and **Train Individual**.

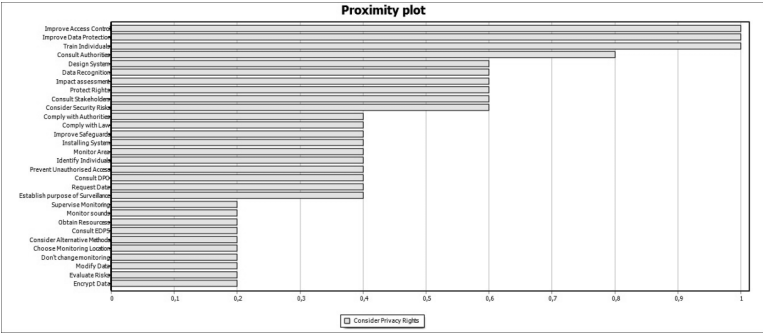


Fig. 4: Proximity Plot for Activity **Improve Access Control** (Using QDA Miner)

It is also possible to analyze Code Sequences in QDA Miner, for example, the frequencies and probability of an activity A followed by another activity B. This analysis yields, for example, that activity **Consider Privacy Rights** is followed by **Improve Data Protection** in 12.5% of the cases.

The above analysis results provide an overview of the relations between coded activities. It is difficult to directly derive process models from these analysis as codes may occur multiple times and the context of each occurrence must be taken into consideration before creating a model. Hence, the analysis results can be taken as hints for candidates when revisiting the coded text again. Selecting code **Consider Privacy Rights** and comparing the coded text fragments with the analysis results, the fragment depicted in Fig. 5 is considered a candidate for a process model reflecting a privacy requirement.

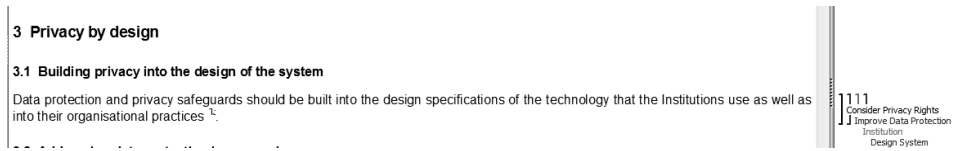


Fig. 5: Text Fragment and Codes: **Institution**, **Consider Privacy Rights**, **Improve Data Protection**, **Design System**

More precisely, the fragment contains the codes **Institution**, **Consider Privacy Rights**, **Improve Data Protection**, **Design System** whereof the three activities **Consider Privacy Rights**, **Improve Data Protection**, **Design System** are related under co-occurrence (cf. Fig. 3), proximity (cf. Fig. 4), and (partly) code sequence probability. The latter shows that **Improve Data Protection** has some probability to follow **Consider Privacy Rights**. The Frequency Matrix shows that **Design System** seems to be not in a sequence with any other activity. Thus, it

can be concluded that **Design System** occurs together with **Consider Privacy Rights** and **Improve Data Protection**, but in no specific order, whereas **Consider Privacy Rights** occurs in sequence with **Improve Data Protection**. The process model in Fig. 6 describes these orders, particularly, the parallel ordering of **Design System** with the other activities.

In the text, activities **Consider Privacy Rights**, **Improve Data Protection**, **Design System** are connected with actor **Institution**. Proximity analysis shows that **Design System** has a proximity of 0.67 and **Consider Privacy Rights** has a proximity of 0.4 (Jaccard coefficient). This assignment is reflected by positioning these two activities in the lane **Institution**. The lane where **Improve Data Protection** is positioned has been marked with ? as the assigned actor must be further investigated. Proximity analysis shows potential candidates such as **Individual**, **Group**, **Officer**, **State**, and **Staff** with a proximity of 1.0. These candidates must be further checked against the text fragments and codes. Due to space restrictions we abstain from details here. However, all lanes can be positioned in pool **Actor** according to the organigram in Fig. 2.

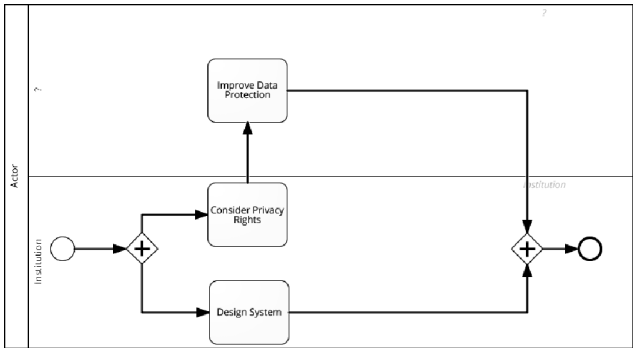


Fig. 6: Guideline Example Derived from Text Fragment and Codes in Fig. 5 (Modeled in BPMN Using Signavio)

The coded text fragment depicted in Fig. 5 is relatively simple. An interesting question is how to deal with more complex text fragments and codes as shown, for example, in Fig. 7.

In some cases where the risks of infringement of fundamental rights are particularly high (for example, in case of covert surveillance or dynamic-preventive surveillance), a privacy and data protection impact assessment should also be carried out and submitted to the EDPS for prior checking. However, apart from these exceptions, there is no need to closely involve the EDPS in the decision-making on how to design a particular system.

Data protection should not be viewed as a regulatory burden, a "compliance box" to be "ticked off". Rather, it should be part of an organisational culture and sound governance structure where decisions are made by the management of each institution based on the advice of their data protection officers and consultations with all affected stakeholders.

We hope that you will find that our Guidelines are useful in your compliance efforts.

Ennead

Fig. 7: Example Text Fragment and Codes for **Consider Privacy Rights**

4 Conclusion and Discussion

The first case study shows that a QCA is in principle an interesting knowledge engineering approach to derive business process models reflecting privacy requirements from text such as regulatory documents. It also shows that a structured methodology is necessary. As a first proposal, we suggest:

PE-QCA Methodology - first draft

1. Code the documents; categories **Actor** and **Activity**
2. Derive organigram from **Actor** hierarchy
3. Apply co-occurrence, proximity, and code sequence analysis to activities and select candidates for process task elements
4. Go back to text and codes, select phrases and codes for candidates
5. Apply sequence analysis to selected activities and derive process model
6. Select attached actors and check proximity for each activity candidate
7. Add pool and lanes respectively
8. **Discuss with experts**

The last item is crucial to validate the feasibility of the process models and is present in most of the existing methodologies. Moreover, most likely the PE-CQA methodology has to be applied iteratively. Probably, for each candidate set of activities all associated text fragments should be considered. We see process as a glue to connect human and technology as well as a vehicle to preserve and enhance privacy in various information systems. Process models can be used to facilitate many aspects of privacy engineering. Especially, they can be used to capture and present the privacy requirements and define privacy-preserving process in system design and operation. As a targeted format of knowledge engineering of privacy requirements, once created, process models can be shared, extended, and verified by domain experts (e.g. law professional, ethical experts, and system engineers) based on reusable models and reproducible procedure and techniques. As next steps, the methodology will be applied to further case studies from the privacy domain. Moreover, the case studies will be repeated with other knowledge engineering techniques such as text mining. The results of the different case studies and of the application of the different techniques will be taken as evaluation of the method proposed above. We think that the most promising way will be a combination of different techniques as all of them have specific advantages.

Another interesting question is how the findings can be transferred to other areas such as health care. Here the extraction and modeling of medical guidelines plays an important role as well [Du11]. The same holds for compliance requirements in general [Ly15]. In order to provide a comprehensive analysis of the transferability of the proposed methodology in the context of privacy requirements, at first, the literature review must be extended to cover the area of compliance requirement

engineering and approaches from other domains such as medical guidelines. For the application of content analysis, the methodology seems to be quite generic and not confined to privacy requirements. However, this statement, must be underpinned with respective case studies which will be part of our future work. Finally, it would be beneficiary to derive entire process models from textual description as process elicitation and modeling can be a tedious and costly job [KRM11]. Friedrich et al. [Fr11] provide an approach based on NLP for the derivation of process models (in BPMN) from text. It will be part of future work to apply a comprehensive analysis and comparison of existing approaches for establishing a methodology for privacy requirement elicitation.

Acknowledgments

This work was partly funded by the EC through the project PrivAcY pReserving Infrastructure for Surveillance (PARIS) (FP7-SEC-2012-1-312504).

References

- [AE00] Antón, A.; Earp, J.: Strategies for developing policies and requirements for secure electronic commerce systems. In: E-commerce security and privacy (2):29–46, 2000.
- [AM14] Ahmed, N.; Matulevicius, R.: A Method for Eliciting Security Requirements from the Business Process Models. In: CAISE Forum, 2014.
- [ANM10] Abu-Nimeh, S.; Mead, N.: Combining Privacy and Security Risk Assessment in Security Quality Requirements Engineering. In: AAAI Spring Symposium: Intelligent Information Privacy Management, 2010.
- [AZ12] Aggarwal, C.; Zhai, C.: Mining text data. Springer Science & Business Media, 2012.
- [BA08] Breaux, T.; Antón, A.: Analyzing regulatory rules for privacy and security requirements. IEEE TSE 34(1):5–20, 2008.
- [Bi08] Birnhack, M.: The EU data protection directive: an engine of a global regime. Computer Law & Security Review 24(6):508–520, 2008.
- [BM10] Bijwe, A.; Mead, N.: Adapting the square process for privacy requirements engineering. Techn. Rep. CMU/SEI-2010-TN-022, Carnegie-Mellon, 2010.
- [BVA06] Breaux, T.D.; Vail, M.W.; Anton, A.I.: Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In: Requirements Engineering, pp. 49–58, 2006.
- [Ch08] Chiasera, A.; Casati, F.; Daniel, F.; Velegrakis, Y.: Engineering privacy requirements in business intelligence applications. In: Secure Data Management, pp. 219–228. Springer, 2008.
- [Ch11] Chikh, A.; Abulaish, M.; Nabi, S.; Alghathbar, K.: An ontology based information security requirements engineering framework. In: Secure and Trust Computing, Data Management and Applications, pp. 139–146. Springer, 2011.

- [Co07] Compagna, L.; Khoury, P.; Massacci, F.; Thomas, R.; Zannone, N.: How to capture, model, and verify the knowledge of legal, security, and privacy experts: a pattern-based approach. In: *P.Artificial intelligence and law*. pp. 149–153, 2007.
- [De11] Deng, M.; Wuyts, K.; Scandariato, R.; Preneel, B.; Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.
- [dRAF05] da Rocha, S.; Abdelouahab, Z.; Freire, E.: Requirement Elicitation Based on Goals with Security and Privacy Policies in Electronic Commerce. In: *WER*. pp. 63–74, 2005.
- [Du11] Dunkl, R.; Fröschl, K.; Grossmann, W.; Rinderle-Ma, S.: Assessing Medical Treatment Compliance Based on Formal Process Modeling. In: *USAB 2011*. Springer, pp. 533–546, 2011.
- [El11] Elahi, G.; Yu, E.; Li, T.; Liu, L.: Security requirements engineering in the wild: A survey of common practices. In: *IEEE Computer Software and Applications Conference*. pp. 314–319, 2011.
- [Fa10] Fabian, B.; Gürses, S.; Heisel, M.; Santen, T.; Schmidt, H.: A comparison of security requirements engineering methods. *Requirements engineering*, 15(1):7–40, 2010.
- [Fr11] Friedrich, F.; Mendling, J.; Puhlmann, F.: Process Model Generation from Natural Language Text. In: *CAiSE*, pp. 482–496, 2011
- [Gr12] Graa, M.; Cuppens-Boulahia, N.; Autrel, F.; Azkia, H.; Cuppens, F.; Coatrieux, G.; Cavalli, A.; Mammar, A.: Using requirements engineering in an automatic security policy derivation process. In: *Data Privacy Management and Autonomous Spontaneous Security*, pp. 155–172. Springer, 2012.
- [Gü05] Gürses, S.; Jahnke, J.; Obry, C.; Onabajo, A.; Santen, T.; Price, M.: Eliciting confidentiality requirements in practice. In: *Conf. of the Centre for Advanced Studies on Collaborative research*. pp. 101–116, 2005.
- [He03] He, Q.; Antón, A. et al.: A framework for modeling privacy requirements in role engineering. In: *Proc. of REFSQ 3*, pp. 137–146, 2003.
- [In97] Introna, L.: Privacy and the computer: why we need privacy in the information society. *Metaphilosophy*, 28(3):259–275, 1997.
- [KBG11] Kalloniatis, C.; Belsis, P.; Gritzalis, S.: A soft computing approach for privacy requirements engineering: The PriS framework. *Applied Soft Computing*, 11(7):4341–4348, 2011.
- [Ki09] Kitchenham, B.; Pearl Brereton, O.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S.: Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.
- [KRM11] Kabicher, S.; Rinderle-Ma, S.: Human-centered process engineering based on content analysis and process view aggregation. In: *Advanced Information Systems Engineering*. pp. 467–481, 2011.
- [KS85] Kowalski, R.; Sergot, M.: Computer Representation of the Law. In: *IJCAI*. pp. 1269–1270, 1985.

- [Le06] Lee, S.; Gandhi, R.; Muthurajan, D.; Yavagal, D.; Ahn, G.: Building problem domain ontology from security requirements in regulatory documents. In: Workshop on Software engineering for secure systems. pp. 43–50, 2006.
- [Le14] Leitner, M.; Rinderle-Ma, S.: A systematic review on security in Process-Aware Information Systems - Constitution, challenges, and future directions. In: Information & Software Technology 56(3): 273-293, 2014
- [Ly15] Ly, L.T.; Maggi, F.M.; Montali, M.; Rinderle-Ma, S.; van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application, and tool-support. Information Systems, 2015. (in press).
- [LYM03] Liu, L.; Yu, E.; Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: IEEE Requirements Engineering Conference. pp. 151–161, 2003.
- [Ma14] Ma, Z. et al: Towards a Multidisciplinary Framework to Include Privacy in the Design of Video Surveillance Systems. In: 2nd Annual Privacy Forum - Privacy Technologies and Policy. pp. 101–116, 2014
- [MdAY14] Martin, Y.; del Alamo, J.; Yelmo, J.: Engineering privacy requirements valuable lessons from another realm. In: Evolving Security and Privacy Requirements Engineering. pp. 19–24, 2014.
- [Me10] Mellado, D.; Blanco, C.; Sánchez, L.; Fernández-Medina, E.: A systematic review of security requirements engineering. Computer Standards & Interfaces, 32(4):153–165, 2010.
- [MMZ08] Miyazaki, S.; Mead, N.; Zhan, J.: Computer-aided privacy requirements elicitation technique. In: IEEE Asia-Pacific Services Computing Conf. pp. 367–372, 2008.
- [MMZ11] Mead, N.; Miyazaki, S.; Zhan, J.: Integrating privacy requirements considerations into a security requirements engineering method and tool. Int’l Journal of Information Privacy, Security and Integrity, 1(1):106–126, 2011.
- [MPZ05] Massacci, F.; Prest, M.; Zannone, N.: Using a security requirements engineering methodology in practice: the compliance with the Italian data protection legislation. Computer Standards & Interfaces, 27(5):445–455, 2005.
- [PDG14] Paja, E.; Dalpiaz, F.; Giorgini, P.: STS-Tool: Security Requirements Engineering for Socio-Technical Systems. In: Engineering Secure Future Internet Services and Systems, pp. 65–96. Springer, 2014.
- [RGK13] Radics, P.; Gracanin, D.; Kafura, D.: Preprocess before You Build: Introducing a Framework for Privacy Requirements Engineering. In: Social Computing (SocialCom). IEEE, pp. 564–569, 2013.
- [Ri14] Riaz, M.; King, J.; Slankas, J.; Williams, L.: Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. In: Requirements Engineering Conference. pp. 183–192, 2014.
- [Sa08] Sarawagi, S.: Information extraction. Foundations and trends in databases, 1(3):261–377, 2008.
- [SBF98] Studer, R.; Benjamins, V.; Fensel, D.: Knowledge engineering: principles and methods. Data & knowledge engineering, 25(1):161–197, 1998.

- [SK12] Salini, P; Kanmani, S: Survey and analysis on security requirements engineering. *Computers & Electrical Engineering*, 38(6):1785–1797, 2012.
- [St06] Strijbos, J.; Martens, R.; Prins, F.; Jochems, W.: Content analysis: What are they talking about? *Computers & Education*, 46(1):29–48, 2006.

A Conceptual Architecture for an Event-based Information Aggregation Engine in Smart Logistics

Anne Baumgrass¹, Cristina Cabanillas², Claudio Di Ciccio²

Abstract: The field of Smart Logistics is attracting interest in several areas of research, including Business Process Management. A wide range of research works are carried out to enhance the capability of monitoring the execution of ongoing logistics processes and predict their likely evolution. In order to do this, it is crucial to have in place an IT infrastructure that provides the capability of automatically intercepting the digitalised transportation-related events stemming from widespread sources, along with their elaboration, interpretation and dispatching. In this context, we present here the service-oriented software architecture of such an event-based information engine. In particular, we describe the requisites that it must meet. Thereafter, we present the interfaces and subsequently the service-oriented components that are in charge of realising them. The outlined architecture is being utilised as the reference model for an ongoing European research project on Smart Logistics, namely GET Service.

Keywords: Smart Logistics; Service-oriented Architectures; Complex Event Processing

1 Introduction

GET Service³ is a European FP7 research project aiming at the realisation of a distributed service-oriented platform for the planning, execution and monitoring of smart transportation processes. The devised platform is meant to be adopted by Logistics Service Providers (LSPs) Europe-wide, in order to take advantage of a powerful infrastructure that allows the improvement of their core business processes, in terms of reduced CO_2 emissions, better time scheduling, more precise service time estimates and thus, reduced costs. Against this goal, we notice that such a platform must build upon the regular synchronisation of its real-world context-awareness. For instance, it is vital that the position of involved transportation means is kept under control during the shipment of goods in order to assist its run-time monitoring. Such information can be gathered by the interception, analysis and interpretation of so-called *events*.

Events are known to be detected by different sensors and reported by several sources. Due to the dynamic nature of the context domain, such information is intrinsically meant to change over time. Therefore, the GET Service core module that is in charge of extracting relevant information on the current development of transportation processes, deals with concurrent event streams stemming from various originators. The information coming from the collection and comparison of the events in the flow of updates has to be

¹ Hasso-Plattner-Institut, University of Potsdam, Germany, anne.baumgrass@hpi.de

² Vienna University of Economics and Business, name.[particle.]surname@wu.ac.at

³ <http://www.getservice-project.eu/>

interpreted to detect and possibly foresee the development of the transportation process, given its execution history and the context within which it is carried out. This paper aims at defining the architecture of the information aggregation and provisioning engine in the context of smart logistics; in particular, in the scope of the GET Service software infrastructure, which is under development at the time of writing and is henceforth referred to as *the platform*.

The remainder of this paper is structured as follows. Section 2 presents background on event processing. In particular, Section 2.1 introduces the fundamental concepts of event processing networks (EPN), processing agents (EPA), source, consumer, object, and channel. Section 2.2 explains how such concepts come into play in the context of event processing. Furthermore, it outlines how aggregation and correlation patterns contribute to the gathering of knowledge regarding the evolution of transportation processes, out of event streams. Then, Section 3 delves into the details of the functionalities that the information aggregation services must offer in the GET Service platform. The discussion is promoted to Section 4, where the architecture of the component offering those services is detailed, in conformance with the aforementioned criteria. Section 5 concludes this paper.

2 Background

This section summarises the background on event processing as well as the requirements that are necessary to design the event aggregation engine.

2.1 Event Processing Infrastructure

Events that are of importance in our context are the *transportation-related events*. They serve three main purposes: (i) tracing how a specific transportation process is executed, (ii) coordinating the different parties involved, and (iii) making appropriate decisions in relation to re-planning and rescheduling. Typically, events are produced and collected by different kind of systems spanning an *event processing network* (EPN) in which *event processing agents* (EPA) are linked by *event channels* to exchange events [EN10], (cf. Fig. 1). Each EPA may act as an *event consumer* to receive *event objects*, and as an *event source* in case it observes *events* and publishes them in a machine-readable form as *event objects*. In this way, an EPA reacts to its input by processing events and outputs events that can be fed to other EPAs over event channels [Lu01].

In the context of GET Service, the GET Service Platform should act as an event consumer to gather events from several event sources (e.g. driver's mobile devices and weather stations) and process them to generate transportation-related events, which might be provided to several consumers, e.g., Logistics Service Providers (LSPs) [Ba13b].

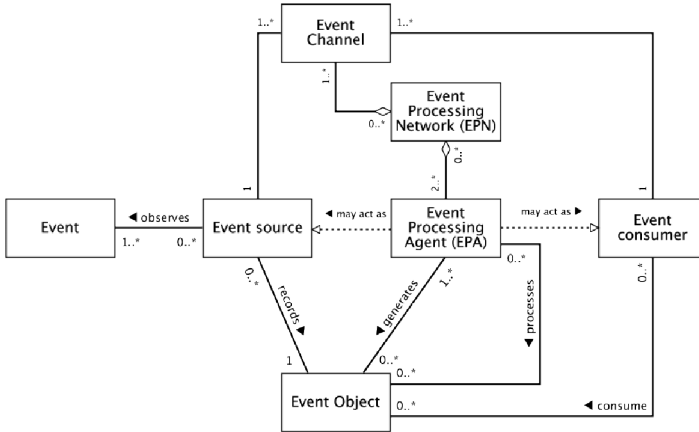


Fig. 1: Event processing infrastructure.

2.2 Event Processing Mechanism

In a smart logistics context like the one of GET Service, it is of utmost importance that all events related to the transportation of goods and corresponding transportation plans can be observed. Furthermore, such events have to be processed to derive transportation-related information, and be published to all interested consumers. To this extent, event processing is in charge of computing operations on events, including reading, creating, transforming, or discarding [EN10]. Specifically, an EPA carries out these operations.

First, an EPA contains an event adapter, which transforms events into event objects [Lu01], e.g., for importing weather forecasts in XML format. The EPA receives the events from an event channel as input in order to process them. Adapters are used to identify event objects published by several event sources, in possibly different formats. Data can be indeed encoded according to standards such as EDIFACT (United Nations/Electronic Data Interchange For Administration, Commerce and Transport [Be94]), MXML (Mining eXtensible Markup Language [vD05]), XES (eXtensible Event Stream, [GV14]), but also more general-purpose ones like CSV (Comma-Separated Values), Excel and XML (eXtensible Markup Language).

Second, events are related to each other according to event relationships. Typically, events are related by time, causality, aggregation [Lu01] or correlation [ROS11]. Event patterns are used to specify these relationships and identify them in an event stream considered by an event processing system. For example, only if `Container mounted` happened before `Goods loaded` in a certain time window and both event objects refer to the same container, they are related by a correlation relationship and can be aggregated to an event `Goods ready for transportation` (cf. Tab. 1). In this way, a correlation is specified through defining correlation attributes (e.g., time and container in the example) between event object types. Furthermore, event aggregation patterns can be used for recognising or

	Goods loaded	Container mounted	Goods ready for transportation
Description	New goods are loaded in Container1.	Container1 is mounted onto Truck1.	Truck1 is ready to start its transportation.
Occurrence time	January 2 nd 2014 07:30	January 1 st 2014 16:30	January 2 nd 2014 07:30
Occurrence location	Harbour of Rotterdam	Harbour of Rotterdam	Port of Rotterdam
Originator(s)	Container1	Truck1, Container1	GET Service Platform
Impact	Truck may start the transportation.	Goods can be loaded in the container.	Truck may start driving and transport the loaded goods to their destination.
Target(s)	Container1	Truck1	TransportationOrder1

Tab. 1: Example events in logistics. For the sake of simplicity, the example is kept simple and abstracted from the real-world. For example, in order to have a truck ready for transportation other events might be relevant as well (e.g., documentation on board).

detecting a significant group of events from among a set of events, and creating a single event that summarises their significance in its data.

Third, event patterns are also used to forward events to interested consumers. For this purpose, an event consumer subscribes to an event processing system with a defined event pattern. Events that match that pattern are sent as notifications over event channels to the consumer. A notification contains data describing an event and may additionally carry information describing the circumstances of the event. In [MFP06], the event processing infrastructure only represents the theoretical components and disregards the issues to be dealt with when implementing this infrastructure in practice, e.g., access control or messaging formats.

We aim to use the transportation context to identify events from several event sources, process them into transportation-related events, and forward them to interested parties.

3 Design of the Information Aggregation Engine

The Information Aggregation Engine is a main component of the platform, which is responsible for collecting events from different sources and processing them in order to offer a unified interface to clients, planners, information providers, and other stakeholders. Thereby, the engine supports, among others, the use cases of track&trace, vessel arrivals and capacity visualisation. The specific functionalities that this service requires in transportation are described in the following sections. Such functionalities derived from a preliminary requirements elicitation phase and thorough analysis of the typical use cases scenario in the context of the GET Service project [Tr13].

3.1 Import and Export of Event Data

It is crucial that the interfaces of the platform adhere to existing messaging standards and interchange formats of all services that are used by the involved stakeholders. For this

purpose, the messaging standards of logistics were investigated in [Ve13]. Four common message types were identified: (i) EDIFACT, used by shipping lines, terminal operators, or customs; (ii) EDIFACT XML (UN/CEFACT working groups); (iii) Business Logistics XML, used by larger Logistics Service Providers (LSPs); and (iv) Excel uploads and downloads, used by smaller Transportation Service Providers (TSPs). Furthermore, applications might use JSON as exchange format, which has less markup overhead in comparison to XML. To unify the communication in logistics for all stakeholders, the e-Freight project⁴ developed a standard framework that also needs to be considered in the platform. It is a standard for freight information exchange covering all transport modes and stakeholders. Each of the above mentioned message types can be transported over different channels using different protocols and services, for example through SOAP web services, HTTP protocols, RPC, or FTP file transfers.

Thus, to extract events from all exchanged messages and to publish transportation-related events in the aggregation engine, it requires four generic interfaces for communication:

1. An interface to import messages of events from different sources (e.g. from client devices of LSPs) provided in different formats. Based on the aforementioned message formats, the engine must be able to call external web services, connect to message queuing services, generate HTTP requests, and download files from FTP. Additionally, it has to offer an interface, to which clients can push events contained in messages.
2. An interface for identifying the event information in these kinds of messages. By implementing adapters the aggregation engine defines where and how to extract events from all the imported messages types. Thus, it must be possible to import events using EDIFACT, XML, Excel, JSON, and the e-Freight format.
3. An interface for submitting event patterns to be notified of the occurrence(s) of events that the stakeholders are interested in. For this purpose, the aggregation engine must enable all stakeholders to specify these event patterns in a well-chosen language, such as Esper⁵. Furthermore, the aggregation engine must be extendable to implement the functionality of deriving event patterns from transportation plans, logistic process models, and route descriptions.
4. An interface to forward events to interested targets. Thus, the aggregation engine must itself provide functionalities to publish events and provide them to the stakeholders involved in transportation. Community systems or other platforms might act as intermediate event distributors. Thus, the engine needs to implement a message queuing service to distribute events and also forward notifications containing information on a subset of events. This forwarding may be implemented as HTTP responses or as API, but may also be realised through emails to be shown on the mobile client devices. The format for the notifications depends on the client devices but should at least adhere to the message standards mentioned above, including EDIFACT, XML, Excel, JSON, and the e-Freight format.

⁴ <http://www.efreightproject.eu/>

⁵ <http://esper.codehaus.org/>

Each of these interfaces must provide capabilities to access and modify the functionalities in order to adapt to a changing environment. Therefore, the interfaces should provide methods to support the standard operations of Create, Read, Update, and Delete (CRUD). For example, partners interacting with the platform should be able to adapt their own aggregation rules to changing plans, or set up new event sources, but also be able to delete rules that are no longer required.

3.2 Normalisation of Events

Because events are collected from different sources that can have different formats, events need to be normalised into a common unified format for further event processing. The normalisation needs to take place in the aggregation engine in order to process events. The normalisation includes the definition of the format of the normalised events, as well as the stored event properties that are available for purposes ranging from information extraction to correlation of events based on values. The different formats and their differing structure imply that the target event format needs to be extensible and general enough to allow for incorporation of structured or unstructured information from all different sources.

The transformation into the unified format can be specified by corresponding adapters. An adapter refers to a component that formats heterogeneous event data into a suitable input format. For example, an event stream in XML format can be processed by an XML parser and events can be extracted based on conversion rules, which can include mappings for different formats of dates and timestamps to the internal format. The mapping rules should be extensible and reusable, such that the task of connecting new sources can be conveniently performed.

3.3 Integration of Event Processing

Once the events are made available to the aggregation engine in a normalised format, the actual event processing has to be performed in form of aggregation and correlation. Thus, the functional requirements for the event processing engine is to support the above mentioned relations between events, i.e. to detect relations based on time, causality, aggregation and correlation. These relations are stored as rules that allow to relate and to aggregate several events.

Furthermore, the aggregation engine is expected to capture a large amount of events and needs to be able to process them within a complex environment where many actors subscribe for their respective events. The actual Complex Event Processing (CEP) system that is used is therefore required to be *scalable*.

3.4 Predictive Functionalities in Cooperation with Discriminative Classifiers

The ability not only to monitor but also to interpret the context information can be seen as one of the main objectives of the event processing component. Indeed, streams of

transportation-related events represent a temporal snapshot over the current development of transportation processes. As an example, the consecutive coordinates and altitude levels of an aeroplane trace its movements. The events may rise from different sources (e.g., weather conditions along the route, traffic information in the arrival airport, etc.) and can altogether concur in the creation of a dangerous situation for the regular advancement of the transportation. Therefore, it is of considerable relevance to distinguish the sequences of events that lead to a disruption from those that are safe. Evaluating queries over event streams is a basic approach to this extent. Such queries would weigh the combination of events over time in order to determine whether the current evolvement of facts is likely to end up in a risky situation, or not. However, it would be impractical to predefine all such queries *a priori*. This is due both to the quantity of possible concurrent causes to check, and to the unfeasibility of foreseeing any possible anomalous sequence of events. To this extent, classifiers from the field of Machine Learning [Mi97] can be of significant help. For instance, Support-Vector Machines (SVMs [CV95]) are supervised learning models for linear classifications, i.e., able to identify a hyperplane in the space of features that separates numeric representations of input objects in two different categories. The hyperplane is determined on the basis of a learning process made on labelled historical data. In the context of transportation-related events, e.g., labelled historical data can represent the reported trajectories of aeroplanes, divided into those that were known to have landed in time and in the expected airport, and those, which were known to have been delayed or diverted. Once trained on such data, the classifier (e.g., SVM) can analyse current flights and predict whether they show an anomalous behaviour, or not. The input as events can be provided by a CEP system, as long as the transportation process specifies the information to be extracted from events to this extent. The learning systems can be used indeed to correlate available data, in order to detect anomalies based on previous knowledge. The selection of independent and dependent variables for the decision functions is thought to be determined *a priori*, since they are strongly domain-related. For instance, the SVM can recognise a possible diversion of flights on the basis of features such as gained distance from the departure airport, velocity and altitude of the aeroplane. However, the input sources (e.g., flight monitoring services) as far as the information aggregation and features extraction (e.g., from positional data to distance, velocity and altitude) are meant to be predefined.

On the basis of the prediction made by the classifier, a new event raising an alert can be generated in case of anomaly detection. Therefore, it is required for the event stream to be restructured in a way that makes it readable from an external classifier, on one hand. On the other hand, the classification returned as a result has to be treated and transformed into a new event. It is worthwhile to recall here that a framework for controlling the safe execution of tasks and signalling possible misbehaviours at runtime has already been outlined in [Ca14b], and preliminary results are already applied in the context of flight diversion detections [Ca14a].

3.5 Correlation of Events to Processes

An additional function within the aggregation engine is the (semi-)automatic derivation of correlation rules on the basis of process data and transportation plans. The intention is to analyse process models, route descriptions, or transport execution plan as input that is analysed to derive correlation and aggregation rules. For this purpose, the components outside the aggregation engine need to provide these documents in a way they can be parsed and event patterns can be derived. In case of processes modelled with the Business Process Model and Notation (BPMN) 2.0⁶ format [Du13], the approach of [Ba13a] can be implemented to correlate events to process instances and identify whether this instance of a process model was executed successfully.

3.6 Notification Mechanism

The aim of the platform is to offer services to many clients and hence, it needs to adhere to common notification paradigms. The publish/subscribe paradigm is very common in distributed systems [MFP06] and needs to be supported by the information aggregation engine. Using this paradigm allows clients and planners to subscribe to certain types of events or aggregated events. For example, a planner might subscribe to all events that are correlated to the respective transportation plans that the planner has created. Then, if events occur during execution, the planner is notified about their occurrence and can react accordingly.

Besides the publish/subscribe paradigm, regular access to events is required in the platform. That is, information providers need the option to add new events directly into the platform via the appropriate interface (push). And additionally, the option to query for recent events from the event history should be made available in that interface (pull).

3.7 Summary

The above sections point out that we aim to design appropriate filtering mechanisms at early stages, to reduce the burden on the correlation and prediction activities. Furthermore, the derivation of correlation rules based on processes range from very simplistic approaches (e.g., correlating by container id), to more sophisticated, control flow, location, and time-aware correlation mechanisms. To provide a brief overview of the requirements of the aggregation engine, a tabular representation is given in Tab. 2.

4 Architecture of the Information Aggregation Engine

This section presents the architecture of the information aggregation engine, in the light of the requirements previously explained. UML component diagrams are used to visualise the logical interconnection of its internal components.

⁶ <http://www.bpmn.org/>

Requirement	Description
R1. Heterogeneous Sources	Connection to different kinds of event sources
R2. Heterogeneous Formats	Collect events from different message formats
R3. Normalisation of Events	Store events of different formats in same normalised format
R4. Event Storage	Store normalised events in a central database
R5. Event Processing	
R5.1 Event Aggregation	Provide functionality to aggregate events of finer granularity to single events
R5.2 Query Subscription	Register queries to be informed of events of interest
R5.3 Domain-Specific Query Subscription	Register queries to be informed of transportation-related events of interest
R5.4 Domain-Specific Event Correlation	Automatically correlate events to transportation processes
R6. Notification Mechanism	Notify subscribed clients of respective events
R7. Event Classifiers	Determine criticality of an event for transportation

Tab. 2: Summary of required capabilities of the event-based information aggregation engine.

4.1 Design of External Interfaces

The information aggregation engine offers four interfaces to be used either by external event sources (e.g., driver, weather stations) or event consumers (e.g., planner, driver). Furthermore, it implements an interface to access the information store to request static information. All five interfaces are required to provide the functionalities described in Section 3. These are shown in the component diagram in Fig. 2 and summarised as follows.

EventAdministrationInterface. The EventAdministrationInterface receives the structural description of an event type and offers further administrative tasks related to events. The communication through this interface has to be implemented in two ways. It should be either initiated by any event source sending the event type description of the events it publishes (push) or it can be configured inside the corresponding event source adapter (cf. EventSourceAdapter, Fig. 4). The implementation is meant to be realised by means of a web service to which the event source can push the event type description.

EventSourceAdapterInterface. Through the EventSourceAdapterInterface the aggregation engine is able to receive events (resp. implementing R1 in Tab. 2). Each event source is intended to be connected through a specific adapter. This adapter then offers an interface of its own that can be used by the event source. Each adapter has to internally use the EventImportInterface (cf. Fig. 4), i.e., the interface through which the AggregationService can take as input and process new events.

EventSubscriptionInterface The EventSubscriptionInterface is used to register subscriptions to the aggregation engine. These subscriptions can be arbitrarily complex, i.e., they may be composed of specific event processing queries. The subscriptions should be pushed to the aggregation engine. Therefore, the aggregation engine provides an implementation of a request-response pattern to register subscriptions in the platform. The events being imported via the EventImportInterface are forwarded to the event consumers by the aggregation engine based on registered subscriptions.

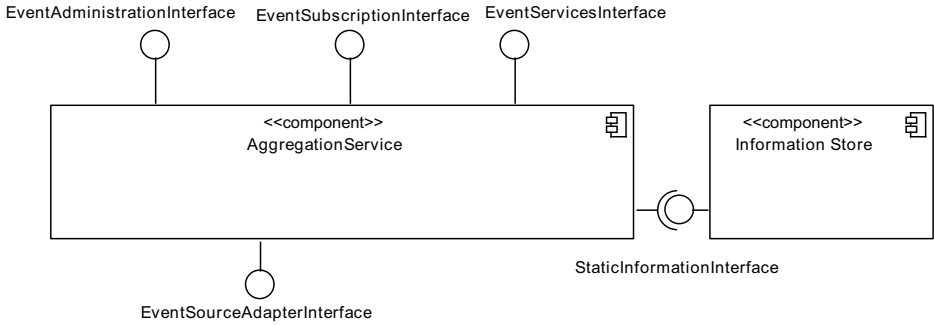


Fig. 2: Interfaces of the information aggregation engine.

EventServicesInterface. The EventServicesInterface combines all services of the aggregation engine that are visible to the external consumers. For example, a consumer may submit routes to the aggregation engine, which can be used in subscriptions later on.

StaticInformationInterface. The StaticInformationInterface offers access to information to enrich events. The aggregation engine uses it to receive all types of information, e.g., about transportation plans and schedules.

Tab. 3 summarises the interfaces with a short description, their inputs and outputs, the interaction pattern realised, and how errors should be handled.

4.2 Structure and Functionality of the Information Aggregation Engine

In this section the three components realising the aggregation engine are described in detail: they are EventHandler, EventProcessing, and EventServices. Fig. 3 shows the three components of the aggregation engine, derived from Fig. 2. The interfaces that they implement are also depicted, along with the interconnecting associations. In the middle, the EventProcessing component handles event transformations and querying. Thus, it includes the functionality of event processing and implements the requirements R5.1 and R5.2 and provides the functionality to implement R5.3 and R5.4 shown in Tab. 2.

4.2.1 The EventHandler

The EventHandler is meant to be implemented to collect, receive, and handle events from different kinds of systems in different formats. This means, it implements the requirements R1, R2, R3, and R4. For that purpose, it provides the EventAdministratorInterface and the EventSourceAdapterInterface. The internal structure of the EventHandler is represented by the following four components (see also Fig. 4).

Interface ID	EventAdministrationInterface
Description	Push event type definitions to the platform, necessary in order to import events of this type, and conduct administrative tasks on events.
Input	Structural description of an event type (e.g. as XSD) or task execution
Output	Confirmation
Interaction Patterns	Synchronous request/response
Error Handling	Synchronous confirmation
Interface ID	EventSourceAdapterInterface
Description	Events are pushed by event source, pulled from event sources or received by a subscription to event sources.
Input	Events including a reference to its event type (e.g., XML)
Output	None
Interaction Patterns	Synchronous push or pull, or publish/subscribe (always depends on the adapter)
Error Handling	No error handling
Interface ID	SubscriptionInterface
Description	Subscribe for events by queries (or other criteria)
Input	event processing query (e.g., String or EPL) or other event criteria
Output	ID of an event channel from which the events are pushed, events
Interaction Patterns	Synchronous request/response, publish/subscribe
Error Handling	Synchronous response or exception, retransmission on publish/subscribe communication
Interface ID	EventServicesInterface
Description	Additional services are offered in relation to events, e.g. process model monitoring or route handling.
Input	Process models (e.g., BPMN), transport orders (e.g., XML), or routes (e.g., JSON)
Output	ID of an event channel from which the events are pushed, events
Interaction Patterns	Synchronous request/response
Error Handling	Synchronous response or exception
Interface ID	StaticInformationInterface
Description	Request/response interface to access information, e.g., about route information and timetables.
Input	Database queries or function calls to databases
Output	Route, timetable, transportation plan
Interaction Patterns	Synchronous request/response
Error Handling	Synchronous response or exception

Tab. 3: Overview of the Interfaces of the information aggregation engine.

EventSourceAdapter Each kind of event sources requires an EventSourceAdapter, which is able to retrieve events from any kind of event source (over the EventSourceAdapterInterface). Event sources differ in the mechanism they use to provide events, e.g., downloads of event information from a FTP server or offering a web service to request events. Thus, all mechanisms to request events from event sources are considered by implementing a corresponding event source adapter through which requirement R1 is met (cf. Tab. 2).

EventReceiver The EventReceiver is responsible for converting the events of an event source into event objects that the aggregation engine can process. For example, one EventSourceAdapter receives events in form of an XML document and another adapter in the JSON format (cf. Section 2.1 and R2 in Tab. 2). Thus, the EventReceiver normalises

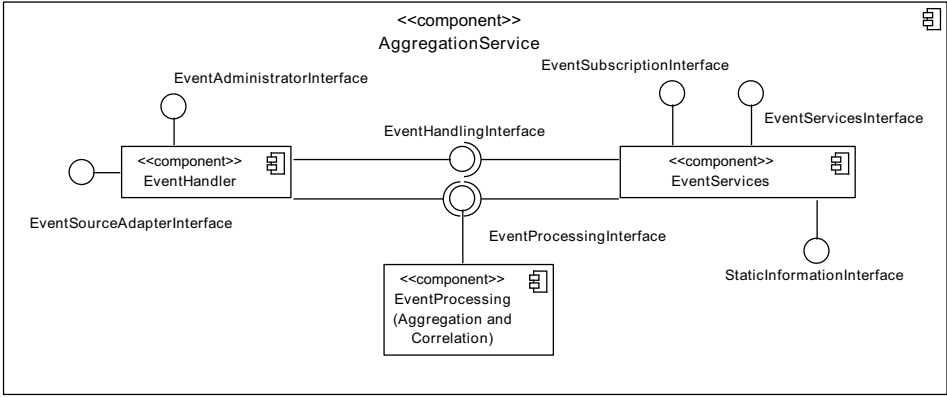


Fig. 3: Architecture Overview of the information aggregation engine.

events in different formats and converts them into the internal structure for processing, i.e., implementing requirement R3 shown in Tab. 2.

EventStore Events are stored in the EventStore, which realises requirement R4 in Tab. 2 of the aggregation engine.

EventManager The EventManager handles all operations on events. This component is the connection between the Event Receiver, the stores and the EventProcessing component via the EventProcessingInterface. In the same way, the connection to the EventServices component is established via the EventHandlingInterface. Thus, the EventManager is responsible to both save and load events and event types from the stores and thereby enables a synchronized access to events and event types.

In summary, the EventHandler is the central component of the Information Aggregation Engine.

4.2.2 The EventServices

The **EventServices** component handles the associations of events to information stored in the event store and handles the communication to event consumers. To this extent, it includes the EventSubscriptionInterface and the EventServicesInterface to external consumers as well as the EventProcessingInterface and the StaticInformationInterface. For internal communication to the EventHandler also the EventHandlingInterface is required, e.g., to reference a specific event type within a subscription. The following three main components are required for its realisation (cf. Fig. 5).

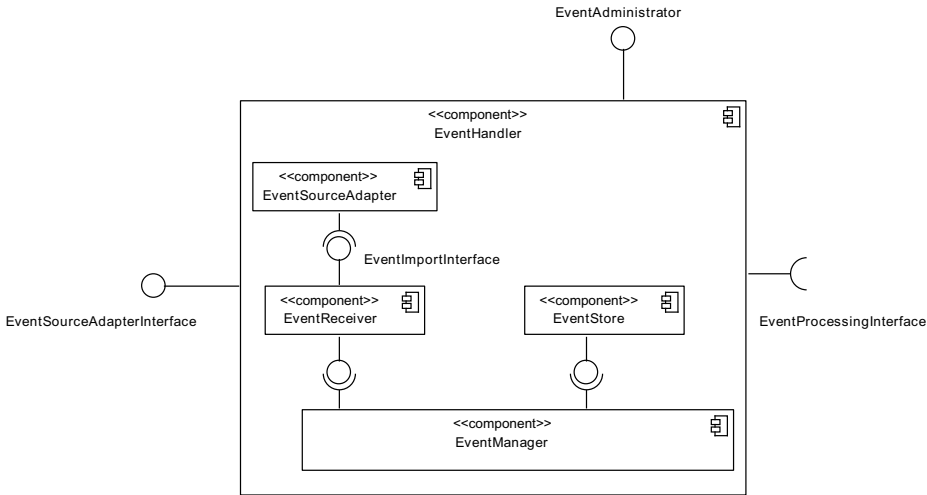


Fig. 4: Structure of the EventHandler component.

SubscriptionManager The SubscriptionManager handles the publication of events to the event consumers based on subscriptions they provided and which are stored in the subscription store. For this purpose, each subscription must include an address to which the events are pushed.

SubscriptionStore All subscriptions are administered in the SubscriptionStore. It thus mainly gathers the requests for receiving updates on events of interests, and serves as a repository where the targets for dispatching events are recorded.

ServiceUnits The ServiceUnits component is a placeholder for all upcoming functionalities that enrich events with external knowledge. For example, the coordinates given by an event may be used to identify the city in which the event occurs. However, this requires that an external knowledge source to be accessible, where the boundaries of cities are given. A first idea of such enhanced event processing is published in [Me13].

Furthermore, predicting algorithms should be developed in this component, to implement the functionality discussed in Section 3.4. In particular, ServiceUnits are meant to be used to meet requirement R7.

In summary, the purpose of the EventServices component is to correlate events to logistics processes but also to external knowledge sources. It is therefore used to extend the platform and realise the requirements R5.3, R5.4, and R7 shown in Tab. 2. Furthermore, it is meant to be used to allow the subscription to events, thus implementing requirements R5.2 and R6.

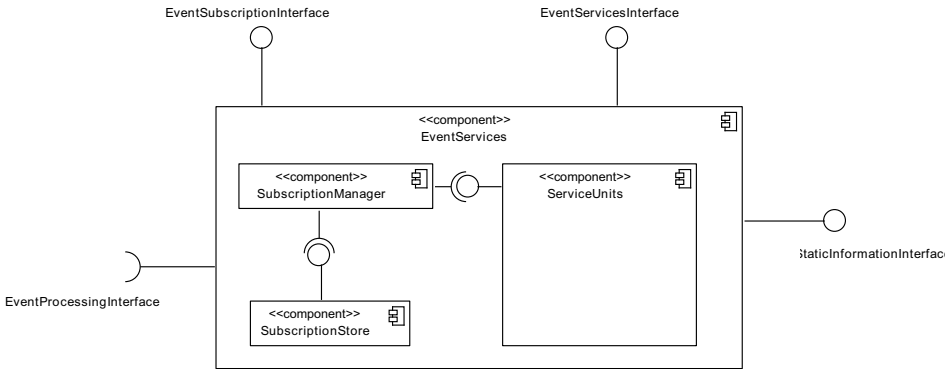


Fig. 5: Structure of the EventServices component.

5 Conclusion

Throughout this paper, the architecture of an event-based information aggregation engine in the context of smart logistics has been described. In particular, the requirements that the information aggregation engine must fulfil have been detailed. They serve as the basis according to which the architecture of the component is designed. Indeed, this paper ends with a thorough analysis of the interfaces offered by the event processing module, along with the description of its internal components and the functionalities offered.

Although all functional requirements are given, challenges may be faced during the implementation. This is due to the dynamic nature of the development process. These dynamics might occur during the implementation of the single components of the aggregation engine and their interaction. More integration effort and dynamics are expected by the integration of the aggregation service in the core GET Service platform. Challenges may also arise from technical requirements (hard- or software) or from necessary event sources that are not publicly available. Furthermore, the complexities of data integration for unifying data, messages, information, and events have to be faced.

Future work will be dedicated to the implementation of the described software components, with a particular focus on the enhancement of their interoperability and extensibility. Efforts will be also put in the devising of the automated process-model-to-queries task for monitoring and processing events, and on the realisation of prediction modules that foresee plausible delays or disruptions during the run-time execution of the transportation activities.

Acknowledgement

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

References

- [Ba13a] Backmann, Michael; Baumgrass, Anne; Herzberg, Nico; Meyer, Andreas; Weske, Mathias: Model-Driven Event Query Generation for Business Process Monitoring. In: ICSOC Workshops. S. 406–418, 2013.
- [Ba13b] Baumgrass, Anne; Cabanillas, Cristina; Di Ciccio, Claudio; Meyer, Andreas; Schmiele, Jürgen: GET Service D6.1: Taxonomy of transportation-related events. <http://getservice-project.eu/en/project/public-deliverables>, 2013.
- [Be94] Berge, John: The EDIFACT standards. Blackwell Publishers, Inc., 1994.
- [Ca14a] Cabanillas, Cristina; Campara, Enver; Di Ciccio, Claudio; Koziel, Bartholomäus; Mendling, Jan; Paulitschke, Johannes; Prescher, Johannes: Towards a Prediction Engine for Flight Delays based on Weather Delay Analysis. In: EMoV. Jgg. 1185 in CEUR Workshop Proceedings. CEUR-WS.org, S. 49–51, March 2014.
- [Ca14b] Cabanillas, Cristina; Di Ciccio, Claudio; Mendling, Jan; Baumgrass, Anne: Predictive Task Monitoring for Business Processes. In: BPM. Jgg. 8659 in Lecture Notes in Computer Science. Springer, S. 424–432, September 2014.
- [CV95] Cortes, Corinna; Vapnik, Vladimir: Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
- [Du13] Dumas, Marlon; La Rosa, Marcello; Mendling, Jan; Reijers, Hajo A.: Fundamentals of Business Process Management. Springer, 2013.
- [EN10] Etzion, Opher; Niblett, Peter: Event Processing in Action. Manning Publications Co., Greenwich, CT, USA, 1st. Auflage, 2010.
- [GV14] Günther, Christian W.; Verbeek, Eric: XES Standard Definition, 2014.
- [Lu01] Luckham, David C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [Me13] Metzke, Tobias; Rogge-Solti, Andreas; Baumgrass, Anne; Mendling, Jan; Weske, Mathias: Enabling Semantic Complex Event Processing in the Domain of Logistics. In: ICSOC Workshops. S. 419–431, 2013.
- [MFP06] Mühl, Gero; Fiege, Ludger; Pietzuch, Peter R.: Distributed Event-based Systems. Springer, 2006.
- [Mi97] Mitchell, Thomas M.: Machine Learning. McGraw Hill series in computer science. McGraw-Hill, Inc., New York, NY, USA, 1. Auflage, 1997.
- [ROS11] Rozsnyai, Szabolcs; Obweiger, Hannes; Schiefer, Josef: Event Access Expressions: A Business User Language for Analyzing Event Streams. In: AINA. IEEE Computer Society, S. 191–199, 2011.
- [Tr13] Treitl, Stefan; Rogetzer, Patricia; Hrušovský, Martin; Burkart, Christian; Bellovoda, Bruno; Jammernegg, Werner et al.: GET Service D1.2: Use Cases, Success Criteria and Usage Scenarios, 2013.
- [vD05] van Dongen, Boudewijn: The MXML standard. <http://www.processmining.org/WorkflowLog.xsd>, 2005.
- [Ve13] van der Velde, Marten; Rook, Hans; Saraber, Paul; Grefen, Paul; Ernst, Albert Charrel: GET Service D2.1: Report Message Standards. <http://getservice-project.eu/en/project/public-deliverables>, 2013.

Process Model Matching Contest

The Process Model Matching Contest 2015

Goncalo Antunes,¹ Marzieh Bakhshandeh,¹ Jose Borbinha,¹ Joao Cardoso,¹ Sharam Dadashnia,² Chiara Di Francescomarino,³ Mauro Dragoni,³ Peter Fettke,² Avigdor Gal,⁴ Chiara Ghidini,³ Philip Hake,² Abderrahmane Khiat,⁵ Christopher Klinkmüller,⁶ Elena Kuss,⁷ Henrik Leopold,⁸ Peter Loos,² Christian Meilicke,⁷ Tim Niesen,² Catia Pesquita,⁹ Timo Péus,¹⁰ Andreas Schoknecht,¹¹ Eitam Sheetrit,⁴ Andreas Sonntag,² Heiner Stuckenschmidt,⁷ Tom Thaler,² Ingo Weber,¹² Matthias Weidlich¹³

Abstract: Process model matching refers to the automatic identification of correspondences between the activities of process models. Application scenarios of process model matching reach from model validation over harmonization of process variants to effective management of process model collections. Recognizing this, several process model matching techniques have been developed in recent years. However, to learn about specific strengths and weaknesses of these techniques, a common evaluation basis is indispensable. The second edition of the Process Model Matching Contest in 2015 hence addresses the need for effective evaluation by defining process model matching problems over published data sets. This paper summarizes the setup and the results of the contest. Next to a description of the contest matching problems, the paper provides short descriptions of all matching techniques that have been submitted for participation. In addition, we present and discuss the evaluation results and outline directions for future work in the field of process model matching.

Keywords: Process matching, model alignment, contest, matching evaluation

1 Introduction

To achieve control over their business operations, organizations increasingly invest time and effort in the creation of process models. In these process models, organizations capture the essential activities of their business processes together with the activity's execution

¹ Instituto Superior Tecnico, Universidade de Lisboa and INESC-ID, Lisbon, Portugal, marzieh.bakhshandeh|joao.m.f.cardoso|goncalo.antunes|jose.borbinha@tecnico.ulisboa.pt

² Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany, Sharam.Dadashnia|Peter.Fettke|Philip.Hake|Peter.Loos|Tim.Niesen|Andreas.Sonntag|Tom.Thaler@dfki.de

³ Fondazione Bruno Kessler, Trento, dragoni|dfmchiara|ghidini@fbk.eu

⁴ Technion - Israel institute of Technology, Technion City, Haifa, Israel, avigal |eitams@ie.technion.ac.il

⁵ LITIO Lab, University of Oran, Oran, Algeria, abderrahmane.khiat@yahoo.com

⁶ University of Leipzig, Leipzig, Germany, klinkmueller@wifa.uni-leipzig.de

⁷ Universität Mannheim, Mannheim, Germany, elena|christian|heiner@informatik.uni-mannheim.de

⁸ VU University Amsterdam, Amsterdam, The Netherlands, h.leopold@vu.nl

⁹ LaSIGE, Faculdade de Ciencias, Universidade de Lisboa, Portugal, cpesquita@di.fc.ul.pt

¹⁰ Technische Hochschule Mittelhessen, KITE - Kompetenzzentrum für Informationstechnologie, Friedberg, Germany, timo.peus@mnd.thm.de

¹¹ Karlsruhe Institute of Technology, Institute AIFB, Karlsruhe, Germany, andreas.schoknecht@kit.edu

¹² Software Systems Research Group, NICTA, Sydney, Australia, Ingo.Weber@nicta.com.au

¹³ Humboldt Universität zu Berlin, Berlin, Germany, matthias.weidlich@informatik.hu-berlin.de

dependencies. The increasing size of process model repositories in industry and the resulting need for automated processing techniques has led to the development of a variety of process model analysis techniques. One type of such analysis techniques are process model matching approaches, which are concerned with supporting the creation of an alignment between process models, i.e., the identification of correspondences between their activities. The actual importance of process model matching techniques is demonstrated by the wide range of techniques that build on an existing alignment between process models. Examples for such techniques include the validation of a technical implementation of a business process against a business-centered specification model [Br12], delta-analysis of process implementations and a reference model [KKR06], harmonization of process variants [WMW11, La13], process model search [DGBD09, KWW11, Ji13], and clone detection [Ek12].

In this paper, we report on the setup and results of the Process Model Matching Contest (PMMC) 2015. It was the second edition of this event after the first PMMC in 2013 [Ca13a] and took place on September 4, 2015, at the 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA) in Innsbruck, Austria. The Contest Co-Chairs were Elena Kuss, Henrik Leopold, Christian Meilicke, Heiner Stuckenschmidt, and Matthias Weidlich.

The Process Model Matching Contest (PMMC) 2015 addresses the need for effective evaluation of process model matching techniques. The main goal of the PMMC is the comparative analysis of the results of different techniques. By doing so, it further aims at providing an angle to assess strengths and weaknesses of particular techniques. Inspired by the Ontology Alignment Evaluation Initiative (OAEI)³, the PMMC was organized as a controlled, experimental evaluation. In total, three process model matching problems were defined and published with respective data sets. Then, participants were asked to send in their result files with the identified correspondences along with a short description of the matching technique. The evaluation of these results was conducted by the Contest Co-Chairs.

There have been 12 submissions to the contest covering diverse techniques for addressing the problem of process model matching. All submissions provided reasonable results and could, therefore, be included in the evaluation and this paper. For each submitted matching technique, this paper contains an overview of the matching approach, details on the specific techniques applied, and pointers to related implementations and evaluations.

We are glad that the contest attracted interest and submissions from a variety of research groups. We would like to thank all of them for their participation.

The remainder of this paper is structured as follows. The next section provides details on the process model matching problems of the PMMC 2015. Section 3 features the short descriptions of the submitted matching approaches. Section 4 presents the evaluation results. Section 5 concludes and discusses future directions.

³ <http://oaei.ontologymatching.org>

2 Data Sets

The contest included three sets of process model matching problems:

- **University Admission Processes (UA & UA_S):** This set consists of 36 model pairs that were derived from 9 models representing the application procedure for Master students of nine German universities. The process models are available in BPMN format. Compared to the 2013 version of the dataset, we have fixed several issues with the models that have to be matched, changed the format of the models, and have strongly improved the quality of the gold standard. With respect to the gold standard, we have distinguished between equivalence matches and subsumption matches (a general activity is matched on a more specific activity). We use in our evaluation both a strict version of the gold standard which contains only equivalence correspondences (UA) and a relaxed version which contains additionally a high number of subsumption correspondences (UA_S).
- **Birth Registration Processes (BR):** This set consists of 36 model pairs that were derived from 9 models representing the birth registration processes of Germany, Russia, South Africa, and the Netherlands. The models are available as Petri-Nets (PNML format). This version of the dataset has also been used in the 2013 contest.
- **Asset Management (AM):** This set consist of 36 model pairs that were derived from 72 models from the SAP Reference Model Collection. The selected process models cover different aspects from the area of finance and accounting. The models are available as EPCs (in EPML-format). The dataset is new to the evaluation contest. The evaluation of this dataset is done blind, i.e., the participants do not know the gold standard of the dataset in advance.⁴

Characteristic	UA	UA _S	BR	AM
No. of Activities (min)	12	12	9	1
No. of Activities (max)	45	45	25	43
No. of Activities (avg)	24.2	24.2	17.9	18.6
No. of 1:1 Correspondences (total)	202	268	156	140
No. of 1:1 Correspondences (avg)	5.6	7.4	4.3	3.8
No. of 1:n Correspondences (total)	30	360	427	82
No. of 1:n Correspondences (avg)	0.8	10	11.9	2.3

Tab. 1: Characteristics of Test Data Sets

Table 1 summarizes the main characteristics of the three data sets. It shows the minimum, maximum, and average number of activities per model as well as the total and average number of 1:1 and 1:n correspondences. A 1:1 correspondence matches two activities A and A' such that no other correspondence in the gold standard matches A or A' to some

⁴ This dataset was developed by Christopher Klinkmüller based on the SAP Reference Model. We thank Christopher for making that dataset available to the contest.

other activity. Contrary to this, 1:n correspondences match an activity A to several other activities A_1, \dots, A_n . This can, for example, happen when an activity has to be matched to a sequence of activities. A high number of 1:n correspondences indicates that the matching task is complex and that the models describe processes on a different level of granularity.

The numbers show that the model sets differ with regard to the number of 1:n correspondences. Obviously, adding subsumption correspondences results in a high number of 1:n correspondences, while the restriction to equivalence correspondences suppresses 1:n correspondences (compare the data sets UA and UA_S). The highest fraction of 1:n correspondences can be found in the BR data set. Even though the number of activities of the models is quite close ranging from 9 to 25, the modeling style seems to differ, because only $\approx 27\%$ of all correspondences are 1:1 correspondences.

3 Matching Approaches

In this section, we give an overview of the participating process model matching approaches. In total, 12 matching techniques participated in the process model matching contest. Table 2 provides an overview of the participating approaches and the respective authors. In the following subsections, we provide a brief technical overview of each matching approach.

No.	Approach	Authors
1	AML-PM	Marzieh Bakhshandeh, Joao Cardoso, Goncalo Antunes, Catia Pesquita, Jose Borbinha
2	BPLangMatch	Eitam Sheetrit, Matthias Weidlich, Avigdor Gal
3	KnoMa-Proc	Mauro Dragoni, Chiara Di Francescomarino, Chiara Ghidini
4	Know-Match-SSS (KMSSS)	Abderrahmane Khiat
5	Match-SSS (MSSS)	Abderrahmane Khiat
6	RefMod-Mine/VM ² (RMM/VM2)	Sharam Dadashnia, Tim Niesen, Philip Hake, Andreas Sonntag, Tom Thaler, Peter Fettke, Peter Loos
7	RefMod-Mine/NHCM (RMM/NHCM)	Tom Thaler, Philip Hake, Sharam Dadashnia, Tim Niesen, Andreas Sonntag, Peter Fettke, Peter Loos
8	RefMod-Mine/NLM (RMM/NLM)	Philip Hake, Tom Thaler, Sharam Dadashnia, Tim Niesen, Andreas Sonntag, Peter Fettke, Peter Loos
9	RefMod-Mine/SMSL (RMM/SMSL)	Andreas Sonntag, Philip Hake, Sharam Dadashnia, Tim Niesen, Tom Thaler, Peter Fettke, Peter Loos
10	OPBOT	Christopher Klinkmüller, Ingo Weber
11	pPalm-DS	Timo Péus
12	TripleS	Andreas Schoknecht

Tab. 2: Overview of Participating Approaches

3.1 AML-PM

3.1.1 Overview

The AgreementMakerLight (AML) [Fa13] is an ontology matching system which has been optimized to handle the matching of larger ontologies. It was designed with flexibility and extensibility in mind, and thus allows for the inclusion of virtually any matching algorithm. AML contains several matching algorithms based both on lexical and structural properties, and also supports the use of external resources and alignment repair. These features have allowed AML to achieve top results in several OAEI 2013 and 2014 tracks [Dr14]. The modularity and extensibility of the AML framework made it an appropriate choice to handle the matching of the datasets of this contest. However, AML works over OWL ontologies, so there was a need to pre-process the input data and translate it into OWL. Then a matching pipeline was applied that included several lexical-based matchers and a global similarity optimization step to arrive at a final alignment.

3.1.2 Specific techniques

The workflow we used is composed of four steps (see Figure 1):

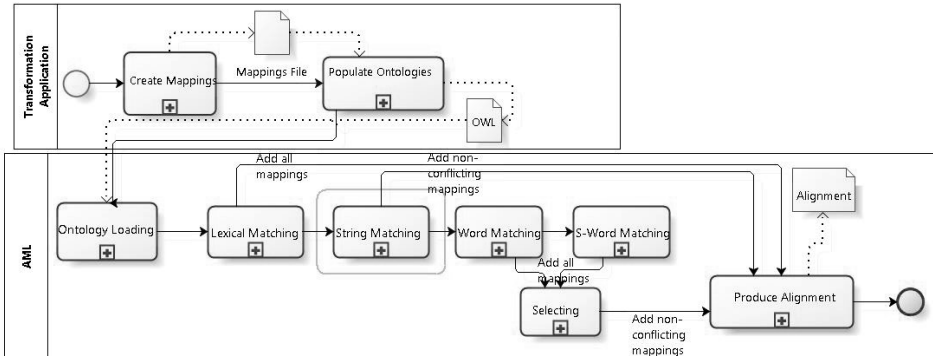


Fig. 1: Transformation Application-AML model matching process

- **Transformation:** Since the contest involved three datasets represented using three different modelling languages, an application for the transformation of the datasets into an ontological representation was used. This transformation application uses data to create and populate ontologies, independently from the schema used for organizing source data. Independence is achieved by resorting to the use of a mappings specification schema. This schema defines mappings to establish relations between data elements and the various ontology classes. Those relations are then used to create and populate an ontology with individuals (instances), thus representing the original data in the form of an OWL ontology.

- **Ontology Loading:** We updated AML to load individuals, which up until now were not handled by this system. When loading an ontology, AML creates efficient data structures that store lexical, structural and semantic information. These include a lexicon, that includes all the labels used in the ontology, and also derived synonyms, by removing leading and trailing stop words.
- **Ontology Matching:** We employed three distinct matchers: The Lexical Matcher, which is one of the simplest and most efficient matching algorithms, looks for literal name matches in the Lexicons of the input ontologies; the String Matcher, which implements a variety of string similarity metrics; and the Word Matcher, which measures the similarity between two individuals through a weighted Jaccard index between the words present in their names. These three matchers are employed in a four step sequential pipeline: first we apply the lexical matcher, and since this is a high-confidence matcher and we include all mappings above a given threshold in our final alignment; then, we apply the string matcher, and all mappings above a threshold that are not in conflict with the mappings already in the alignment are added; finally we apply the word matcher with and without stemming of words. These mappings, given their lower confidence are then run through a selection step before being added to the final alignment.
- **Selection:** Selectors are algorithms used to trim an alignment by excluding mappings below a given similarity threshold and excluding competing mappings to obtain the desired cardinality, typically one-to-one. The selector algorithm sorts the mappings in the Alignment in descending order of their similarity values, then adds mappings to the final alignment, as long as they do not include individuals already selected, until it hits the desired cut-off threshold.

3.2 BPLangMatch

3.2.1 Overview

This matching technique is tailored towards process models that feature textual descriptions of activities, introduced in detail in [We13]. Using ideas from language modeling in Information Retrieval, the approach leverages those descriptions to identify correspondences between activities. More precisely, we combine two different streams of work on probabilistic language modeling. First, we adopt passage-based modeling such that activities are passages of a document representing a process model. Second, we consider structural features of process models by positional language modeling. Combining these aspects, we rely on a novel positional passage-based language model to create a similarity matrix. The similarity scores are then adapted based on semantic information derived by Part-Of-Speech tagging, before correspondences are derived using second line matching. Figure 2 illustrates the various steps of our approach.

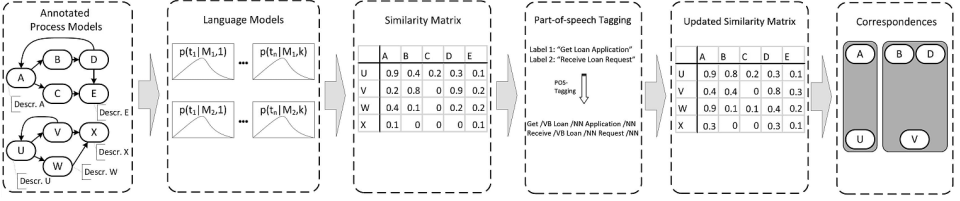


Fig. 2: Overview of the process model matching steps

3.2.2 Specific Techniques

Activities as Passages. Let \mathcal{T} be a corpus of terms. For a process model P , we create a document $d = \langle T_1, \dots, T_n \rangle$ as a sequence of length $n \in \mathbb{N}$ of passages, where each passage $d(i) = T_i \subseteq \mathcal{T}$, $1 \leq i \leq n$, is a set of terms. The set $d(i)$ comprises all terms that occur in the label or description of the activity at position i . The length of d is denoted by $|d|$. We denote by \mathcal{D} a set of processes, represented as documents.

Our model is built on a cardinality function $c : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow \{0, 1\}$, such that $c(t, d, i) = 1$ if $t \in d(i)$ (term t occurs in the i -th passage of d) and $c(t, d, i) = 0$ otherwise. To realize term propagation to close-by positions, a proximity-based density function $k : (\mathbb{N} \times \mathbb{N}) \rightarrow [0, 1]$ is used to assign a discounting factor to pairs of positions. Then, $k(i, j)$ represents how much of the occurrence of a term at position j is propagated to position i . We rely on the Gaussian Kernel $k^g(i, j) = e^{-(i-j)^2/(2\sigma^2)}$, defined with a spread parameter $\sigma \in \mathbb{R}^+$ [LZ09]. In this contest we used $\sigma = 1$. Adapting function c with term propagation, we obtain a function $c' : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow [0, 1]$, such that $c'(t, d, i) = \sum_{j=1}^n c(t, d, j) \cdot k^g(i, j)$. Then, our positional, passage-based language model $p(t|d, i)$ captures the probability of term t occurring in the i -th passage of document d ($\mu \in \mathbb{R}$, $\mu > 0$, is a weighting factor):

$$p_\mu(t|d, i) = \frac{c'(t, d, i) + \mu \cdot p(t|d)}{\sum_{t' \in \mathcal{T}} c'(t', d, i) + \mu}. \quad (1)$$

Derivation of Passage Positions. To instantiate the positional language model for process models, we need to specify how to order the passages in the document to represent the order of activities in a process. In this matching contest, we chose to use a Breadth-First Traversal over the process model graph starting from an initial activity that creates the process instance (we insert a dummy node connect to all initial activities if needed).

Similarity of Language Models. Using the language models, we measure the similarity for document positions and, thus, activities of the process models, with the Jensen-Shannon divergence (JSD) [Li91]. Let $p_\mu(t|d, i)$ and $p_\mu(t|d', j)$ be the smoothed language models of two process model documents. Then, the probabilistic divergence of position i in d with

position j in d' is:

$$jsd(d, d', i, j) = \frac{1}{2} \sum_{t \in \mathcal{T}} p_{\mu}(t|d, i) \lg \frac{p_{\mu}(t|d, i)}{p^{+}(t)} + \frac{1}{2} \sum_{t \in \mathcal{T}} p_{\mu}(t|d', j) \lg \frac{p_{\mu}(t|d', j)}{p^{+}(t)} \quad (2)$$

with $p^{+}(t) = \frac{1}{2}(p_{\mu}(t|d, i) + p_{\mu}(t|d', j))$

When using the binary logarithm, the JSD is bound to the unit interval $[0, 1]$, so that $sim(d, d', i, j) = 1 - jsd(d, d', i, j)$ can be used as a similarity measure.

Increasing Similarity Scores. In many cases, when we encounter textual heterogeneity in the label and description of two similar activities, the nouns remain the same, and the heterogeneity is limited to verbs, adjectives, and other words. Thus, once a similarity matrix has been derived for two process models, we increase score of activities who share the same nouns. For identifying the nouns of each activity, we rely on the Stanford Log-linear Part-Of-Speech Tagger [To03].

Derivation of Correspondences. Finally, we derive correspondences from a similarity matrix over activities, which is known as second line matching. Here, we rely on two strategies, i.e., *dominants* and *top-k*, see [GS10]. The former selects pairs of activities that share the maximum similarity value in their row and column in the similarity matrix. The latter selects for each activity in one model, the k activities of the other process that have the highest similarity values.

3.3 KnoMa-Proc

3.3.1 Overview

The proposed KnoMa-Proc system addresses the process model matching problem in an original way. It implements an approach based on the use of information retrieval (IR) techniques for discovering candidate matches between process model *entities*⁵. The use of IR-based solutions for matching knowledge-based entities is a recent trend that has already shown promising results in the ontology matching [ES07] field [Dr15] and in the process matching one [We13].

The idea of the work is based on the construction and exploitation of a structured representation of the entity to map and of its “context”, starting from the associated textual information. In case of ontologies, the notion of “context” refers to the set of concepts that are directly connected (via a “is-a” property) to the concept to map, or that have a distance from it (in terms of “is-a” relations to traverse) lower than a certain degree. When considering processes, the semantics of “context” has to be revised. In the proposed implementation, the “context” of a process entity is the set of entities that are *directly connected* to it, i.e., for which there exists a path in the process model that does not pass through any other entity.

⁵ Here on we use the term *entity* in a wider sense to denote process model flow elements that do not control the flow, e.g., activities and events in BPMN, transitions in Petri-Nets, functions and events in EPC.

In the current prototype, only flow elements that do not control the flow of the process model diagram (e.g., activities and events) have been considered, abstracting from other flow elements (e.g., BPMN gateways and Petri-Net conditions).

3.3.2 Specific Techniques

The matching operation is performed in two different steps: (i) creation of an index containing a structured description of each entity, and (ii) retrieval procedure for finding candidate matches.

Index Creation The index creation phase consists in exploiting information about entities and their “contexts” for building an inverted index for each process model to be matched (i.e., for each process in the challenge dataset). To this aim, for each process and for each entity of the process, the system extracts: (i) the entity label; (ii) the set of labels of the entities that *directly precede* the current one (`inputlabel`) if any; and (iii) the set of labels of the entities that *directly follow* the current one (`outputlabel`), if any. Intuitively, an entity e_1 directly precedes an entity e if there exists a path from e_1 to e (and no other entity occurs in the path). Similarly, an entity e_2 directly follows an entity e if there exists a path from e to e_2 (and no other entity occurs in the path). In the current implementation the system explores only the set of entities that directly precede and follow the current entity. In the future more sophisticated techniques will be investigated for improving the effectiveness of the system.

Once the information has been extracted, the textual information contained in each label is processed in order to obtain the lemmatized version of each textual token and the structured representation of each entity is built (Fig. 3) and indexed.

```
label: entity_label
inputlabel: input_label_1, ..., input_label_n
outputlabel: output_label_1, ..., output_label_n
```

Fig. 3: Entity structured representation

Match Search The matching operation inherits part of the procedure adopted for creating the index. Given two processes that have to be mapped (for example “Process 1” and “Process 2”), the structured representation of each entity of “Process 1” is transformed in a query performed on the indexes of the entities of the other process. The matching operation between two processes consists in performing queries by using entities of “Process 1” on the index of entities of “Process 2” and vice versa. Once all queries in both directions have been performed, the two sets of identified matches (M_{12} and M_{21}) are analyzed to compute the set M of the best matches, i.e., the set of matches that will be returned by the system.

To this purpose, the following three rules are applied by the system in the given order:

1. if a match m is identified for a given entity in both sets ($m \in M_{12}$ and $m \in M_{21}$), it is automatically stored in M ;
2. if a match m is identified for a given entity only in one set (either $m \in M_{12}$ or $m \in M_{21}$), if the confidence score (computed during the retrieval) is higher than a threshold $th = 0.75$, the match is automatically stored in M ;
3. if an entity is matched with several entities but none of the two conditions above apply (i.e., none of the matches is present in both sets), the two matches with the highest confidence score from $M_{12} \cup M_{21}$ are stored in M .

Purpose of the second and the third rules is avoiding to have a too restrictive system. The set of the best matches M is finally stored in the output file.

3.4 Match-SSS and Know-Match-SSS

3.4.1 Overview

The Match-SSS (MSSS) system uses NLP techniques to normalize the activity descriptions of the two models to be matched. It first uses string-based and WordNet-based algorithms. Finally, the approach selects the similarities calculated by these two matchers based on a maximum strategy with a threshold to identify equivalent activities. The Know-Match-SSS (KMSSS) system is similar, but uses another technique based on the category of words.

3.4.2 Specific Techniques

Extraction and Normalization The systems take as input the two process models to be matched and extract their labels. Then, NLP [Ma14] techniques are applied to normalize these labels. In particular, three preprocessing steps are performed: (1) case conversion (conversion of all words in same upper or lower case) (2) lemmatization stemming and (3) stop word elimination. Since String and WordNet based algorithms are used to calculate the similarities between labels, these steps are necessary.

Similarity Calculation In this step, both approaches calculate the similarities between the normalized labels using various base matchers. More precisely, the edit distance as string-based algorithm and the Lin algorithm [Li98] for WordNet-based similarity are applied. The Know-Match-SSS additionally uses another matcher based on the category of words. This matcher calculates the similarities between words based on their categories using a dictionary.

Aggregation and Identification In this step, our two systems select the similarity values calculated by different matchers using the maximum strategy. Finally, we apply a filter on similarity values retained in order to select the correspondences (equivalent activities between the two models) using a threshold.

Implementation To parse the process models, we used the jDOM API. For the normalization step, we made use of the Stanford CoreNLP API. To implement our matcher, we used the edit distance and the Lin WordNet-based Similarity. The retained similarity between words of a sentence is based on a maximum strategy.

3.5 RefMod-Mine/VM²

3.5.1 Overview

The RefMod-Mine/VM² approach to business process model matching presented in the following is a refinement of our concept outlined in [NH15]. It focuses on the labels of a process model to determine mappings between activities based on their textual similarity. Therefore, established techniques from the field of Information Retrieval are combined with Natural Language Processing (NLP) to leverage information from text statistics.

As a preparatory step, every model to be compared is imported and transformed into a generic model format, where importers for BPMN, EPC and Petri-Nets are provided. As the notion of distinct 1:1 matches – i. e. a node label from a model *A* cannot be mapped to more than one node label from a model *B* – is underlying, possible multiple matches are removed from the final mapping as a last step.

3.5.2 Specific Techniques

The general procedure is defined by a three-step process, which is referred to as *multi-stage matching approach*. This process is carried out on each pairwise combination of all node labels that constitute the process models that are to be compared. A subsequent stage is only reached if the proceeding stage does not determine an appropriate match.

Trivial Matching First, a *trivial matching* is performed to identify identical labels as well as labels that are substrings of each other. Since this kind of similarity is valued most important, it constitutes the first step in our approach. Two labels *A* and *B* are considered “similar” if either $A == B$ or A is substring of B || B is substring of A .

Lemma-based Matching As an extension to the trivial matching approach, labels are further processed by NLP methods to harmonize the set of label terms and, thus, reach a

higher level of abstraction. First we split labels into constituting words – so-called *tokens* – and subsequently perform *lemmatization* on those tokens to unify different inflected word forms. Labels are then compared based on their set of lemmas, i. e. the intersection of terms in the label lemma sets is computed while abstracting from a specific word order (*bag of words* [Wa06]). In order to ensure high precision during matching, lemma sets may only differ by a small amount of terms (parameter i) and must have a certain length (parameter j) to be considered a match. The ratio between identical lemmas and absolute lemma set size depicts another threshold (parameter t_1). Values of i , j and t_1 have been determined iteratively using the provided gold standards with respect to high precision. As this stage only aims to identify “mostly identical” labels with a different order or inflection of words, thresholds are set very tight.

Vector-based detail matching At the centerpiece of this contribution is a *vector space model* (VSM) approach that enables both the retrieval of similar models to a given query model as well as the calculation of similarities between labels within these models. This procedure is three-part: *First*, for each combination of two models that have to be matched, the *k-nearest neighbors* (k-NN) are determined per model [CD07]. This is done by computing the *cosine similarity* between the vectors spanning across all lemmas within the set of all process models with respect to the particular query model. *Second*, label vectors are built per label pair combination within the two models to be matched, i. e. the number of dimensions of these vectors equals the sum of distinct lemmas in the two labels. To weight vector dimensions, the k-NN set is considered a new sub-corpus, which is in turn used to calculate *tf-idf* values for every label term lemma t in document d in corpus D according to formula (1) [Ra03].

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) = count(t \in d) \times \log \frac{|D|}{df(t, d)} \quad (3)$$

Third, cosine similarity sim_{cos} is then calculated between label vectors and checked against a predefined threshold (parameter t_2) as depicted in formula (2).

$$t_2 \leq sim_{cos}(\theta) = \frac{\sum_{i=1}^n v_i \times w_i}{\sqrt{\sum_{i=1}^n (v_i)^2} \times \sqrt{\sum_{i=1}^n (w_i)^2}} \quad (4)$$

By using this approach as a third stage in the overall matching procedure, the approach seeks to exploit statistical information from word occurrences and, thus, to reflect the importance of specific terms within the corpus. By including the k-NN of a model it further seeks to broaden the scope of consideration in order to obtain significant information about term distributions.

3.6 RefMod-Mine/NHCM

3.6.1 Overview

This matcher enhances the RefMod-Mine/NSCM approach presented at the PMC 2013 and consists of 3 general phases. In the pre-processing phase (1), the input models are transformed into a generic format, which allows an application of the matching approach to models of different modeling languages. In the processing phase (2), all available models of a dataset are used as an input for the *n-ary cluster matcher*, which uses a *natural language based similarity measure* for a pairwise node comparison. As a result, several sets of clusters containing nodes of all considered models are produced, which are then being extracted to *binary complex mappings* between two models. Finally, that binary complex mappings are being post-processed (3) in order to eliminate non corresponding maps resulting from the clusters.

The technique has been implemented in the form of a php command line tool and can publicly checked out at <https://github.com/tomson2001/refmodmine>. It is also available as an online tool in the context of the *RefMod-Miner as a Service* at <http://rmm.dfki.de>.

3.6.2 Specific Techniques

In the pre-processing phase of the approach, the input models are *transformed* into a generic format which constructs are similar to the extended EPC. At the moment, the transformation of BPMN, Petri-Net and EPCs is supported, whereby it is generally tried to lose as few information as possible. Especially in the case of BPMN it is important to keep the information caused by the variety of constructs, since they might be very useful in the context of process matching. Additionally, the pre-processing phase contains a *semantic error detection*, where defects of modeling are being identified and automatically corrected. This also includes a mechanism modifying the models concerning a consistent modeling style within a dataset and the solution of abbreviations, which are learned from the dataset.

The processing phase consists of the following components.

N-Ary cluster matching In contrast to existing matching techniques, the authors use a n-ary clustering instead of a binary matching. The nodes of all models are being pairwise compared using a semantic similarity measure. Since the cluster algorithm is agglomerative [JMF99], it starts with clusters of size 1 (=node) and consolidates two nodes to a cluster if their similarity is approved by the matching algorithm. If two nodes are being clustered and both are already part of different clusters, the two clusters are being merged. Thus, the resulting clusters are hard and not fuzzy [JMF99].

Semantic similarity measure The used similarity measure consists of three phases. The first phase splits node labels L into single words (stop words are being removed) w_{iL} , so that $split(L) = \{w_{1L}, \dots, w_{nL}\}$. The second phase computes the Porter Stem [Po97] $stem(w_{iL})$ for each word and compares the stem sets of both labels. The number of stem matchings is being divided by the sum of all words.

$$sim(L_1, L_2) = \frac{|\{stem(w_{1L_1}), \dots, stem(w_{nL_1})\} \cap \{stem(w_{1L_2}), \dots, stem(w_{mL_2})\}|}{|split(L_1) + split(L_2)|}$$

If $sim(L_1, L_2)$ passes a user-defined threshold, the labels are being checked for antonyms using the lexical database WordNet [Mi95] and checking the occurrence of negation words like "not".

Homogeneity-based detail matching Since the *semantic similarity measure* is not able to match synonyms, it is necessary to apply an additional technique. Based on the homogeneity degree of the model set, it is decided, whether and which further node pairs are being considered as potentially equivalent. The homogeneity degree is defined as:

$$HD = \frac{|multi.occuring.label| - |min.multi.occuring.label|}{|max.multi.occuring.label| - |min.multi.occuring.label|}$$

with $|multi.occuring.label|$ is the number of different node labels occurring in at least two models, $|max.multi.occuring.label|$ is the number of all nodes minus the number of different node labels and $|min.multi.occuring.label| = \frac{2 * |min.multi.occuring.label|}{|num.epcs.in.dataset|}$.

The potential node pairs are now analyzed in detail. It is checked whether verb, object and further elements of the labels are equivalent by using WordNet [Mi95] and Wiktionary ⁶.

Binary matching extraction For each model pair all clusters are being scanned for the occurrence of nodes of both models. The containing node set of the first model is then being matched to the node set of the second model. This returns a binary complex (N:M) mapping for each model pair.

Since the matching approach might produce transitive correspondences over several models which are not meaningful in all cases, the binary mappings are additionally checked for antonyms and verb-object-correspondences using WordNet and Wiktionary as well as for organizational mismatches. Therefore, the bag-of-words [K113] of the nodes related to the organizational units are being calculated in order to match the organization units. Finally and depending on the *homogeneity degree*, the arity of the complex matches is being justified. This bases on the assumption, that a growing homogeneity degree leads to a reduction of the mapping complexity (the arity). Thus, the models describe the processes on a similar granularity.

⁶ <http://www.wiktionary.org>

3.7 RefMod-Mine/NLM

3.7.1 Overview

The Natural Language Matcher (NLM) identifies corresponding process model labels and consequently corresponding nodes. It is predominantly based on natural language processing techniques using a bag of words concept. In contrast to the existing bag of words matching approach [K113], the NLM makes use of word classification. The matcher is capable of identifying simple matches as well as complex matches between two process models. Since the approach mainly relies on the labels used in process models, it can be applied to any kind of process modeling language. The matcher is implemented in Java 1.8 and embedded in the *RefMod-Mine*⁷ toolset.

3.7.2 Specific Techniques

The approach is divided into two major steps. In the first step the natural language that is contained in the labels is processed. This includes a tokenization of the labels to identify the words contained in a label, a part-of-speech analysis to determine the syntactic category, and a lemmatization of the identified words. The second step represents the actual matching. Given two models M_1 and M_2 with their respective node sets N_1 and N_2 . Based on the node types and the extracted linguistic information of step one, the matcher decides in the second step which pairs $(n_1, n_2) \in N_1 \times N_2$ are considered a match.

At first, the matcher checks the feasibility of a node pair. A node pair is considered feasible if the node types are marked as corresponding. These type correspondences can be parametrized and unless otherwise specified only identical node types are considered corresponding. Let NN be the list of nouns, VB the list of verbs, and JJ the list of adjectives that a label l can contain. A feasible node pair (n_1, n_2) is considered a match if their labels l_1, l_2 containing the word lists NN_1, VB_1, JJ_1 and NN_2, VB_2, JJ_2 meet at least one of the conditions listed:

- *identical condition*
 - each noun of NN_1 corresponds to at least one noun of NN_2 and vice versa
 - each verb of VB_1 corresponds to at least one verb of VB_2 and vice versa
 - each adjective of JJ_1 corresponds to at least one adjective of JJ_2 and vice versa
- *cross-category condition*
 - l_1 only contains one adjective or one verb and l_2 contains at most two words of which at least one word is a noun that corresponds to the single word contained in l_1 , or

⁷ <http://refmod-miner.dfki.de>

- l_2 only contains one adjective or one verb and l_1 contains at most two words of which at least one word is a noun that corresponds to the single word contained in l_2

The conditions are based on the assumption that identical nodes share the same nouns, verbs and adjectives. However, similar nodes might only share a subset of words in their respective word categories. Therefore, the cross-category condition is applied. Independent of the word category the lexical relation between two words determines their correspondence. The words w_1, w_2 correspond if their lemmata meet at least one of the following conditions:

- w_1 is identical to w_2
- w_1 is a synonym of w_2 or w_2 is a synonym of w_1
- w_1 is a hyponym of w_2 or w_1 is a hyponym of w_2
- w_1 is an etymologically related term of w_2 or w_2 is an etymological related term of w_1

Beside the identity relation, a synonym and a hyponym relation are considered appropriate lexical relations to determine similar words. The etymological relation is primarily used to determine similar words of different word categories.

3.7.3 Implementation

The presented approach uses the *Stanford CoreNLP API*⁸ [Ma14] for Java to perform the language processing. The matcher determines the lexical relations based on *Wiktionary*⁹ and the *Java-based Wiktionary Library*¹⁰.

3.8 RefMod-Mine/SMSL

3.8.1 Overview

RefMod-Mine/SMSL is a semantic matching algorithm based on a supervised machine learning approach. The approach consists of two stages: (1) First the algorithm is given a repository of process models and its gold standard. The algorithm identifies the tokens of the process labels and determines their tags (verb, noun, ...). Then it performs a search for semantically related words in the Wordnet [Mi95]. As a measure of the quantification of the semantic relation of two words, a composed function is used that depends on the

⁸ <http://nlp.stanford.edu/software/corenlp.shtml>

⁹ <https://en.wiktionary.org>

¹⁰ <https://www.ukp.tu-darmstadt.de/software/jwktl>

semantic distance between both words and the intermediate words in Wordnet. All tags and the semantic distance are weighted. When the algorithm calculated all semantic relations as matchings, it stores all weights of the function and the reached precision, recall and F-value. These weights are then optimized by the resulting F-value in a local search. (2) When the weights have been stored/learned, the algorithm applies the best found weights on new given matchings.

3.8.2 Specific Techniques

(1) At the beginning, the node labels of the process models are divided in tokens by the Stanford tokenizer [Ma14]. The tokens are lemmatised so that grammatical forms are neutralized. Then for each token its tag is determined by the Stanford tagger [To03]. After all tokens with their tags are determined, a similarity function is defined. This function calculates the similarity between two tokens $t1, t2$ and is composed of the LIN score by [Li98], the path length between $t1, t2$ in Wordnet and the weights of the tokens tags. More exactly, the similarity between tokens $t1, t2$ is equal to $weight_LIN * LIN(t1, t2) + weight_pathLen * pathLength(t1, t2) + weight_tag(t1) + weighted_tag(t2)$ with $weighted_tag(token) = weight_tag(getTagFromToken(token))$. Each tag has its own weight. So a verb can have another weight than a noun or a gerund.

RefMod-Mine/SMSL seeks to find the weights that reach the highest F-value by local search. Therefore the algorithm calculates the token similarity function with different weight combinations and records the associated F-value. First the weights are defined with a wide range and then the weight combination with the highest F-value is the basis for refining the weights until no better F-value appears. (2) Then the algorithm has completed and can now apply its learned weights on new matchings.

3.8.3 Implementation

The matching algorithm itself has been implemented in Java 1.8 and the local search has been implemented in Python 2.7.9.

3.9 OPBOT

3.9.1 Overview

The *Order Preserving Bag-Of-Words Technique* (OPBOT) is based on two cornerstones: improved label matching and order preservation. To improve the effectiveness of label matching, we first identify equally labeled activities and then reduce the level of detail in the remaining labels. The former is motivated on the observation that equally labeled activities most often constitute 1:1-correspondences. The latter builds upon our previous work [K113] where label pruning was used to increase the recall. Here, we employ our

maximum-pruning bag-of-words similarity (MPB) that performed well in the previous iteration of the matching contest [Ca13a]. *Order preservation* builds on the idea that multiple correspondences between two models occur in the same order in both models. To this end, we employ the *relative start node distance* (RSD) [K114]. OPBOT processes all model pairs in a single run. It is based on the general matching workflow for schema matching [Ra11]. Below we discuss the processing steps in more detail.

3.9.2 Specific Techniques

In the pre-processing step, the process models from the collection are loaded as business process graphs [Di09]. Then, label normalization, tokenization, and stemming¹¹ are applied to transform each label into a bag-of-words. Finally, we count how many times two words co-occur in the same bag-of-words.

Next, a filter assigns a similarity score of 1 to all equally labeled activity pairs. In case an activity is part of such an activity pair, a similarity value of 0 is assigned to any other activity pair that includes this activity. A second filter assigns a value of 0 to all remaining activity pairs whose RSD difference yields an absolute value of at least 0.5. The RSD difference of two activities a and b is defined as $\Delta_{RSD}(a, b) := RSD(a) - RSD(b)$.

For the activity pairs with a similarity of neither 0 nor 1, three matchers then calculate similarity scores independently – resulting in three alignments per model pair. All matchers rely on the MPB, but employ different word similarity measures and threshold values t . Given an activity pair, each matcher computes a similarity score. If it is greater or equal to the respective t , it will be assigned to the activity pair; 0 otherwise. The *syntactic matcher* uses the longest common subsequence similarity [ES07], with $t = 0.76$ in the experiments. The *paradigmatic sense relation matcher* is based on Lin’s similarity metric [Li98] and WordNet [Mi95], with $t = 0.76$. The *syntagmatic sense relation matcher* utilizes the co-occurrence counts from the pre-processing. To determine the similarity for a pair of words it identifies – for each word individually – the two most frequently co-occurring words and then calculates the cosine co-occurrence similarity [Na09], with $t = 0.84$.

Next, the three matchers are ranked based on their *order preservation score* (OPS). In the calculation of OPS, only correspondences (activity pairs with a similarity score not equal to 0) are considered. A pair of correspondences $((a_1, b_1), (a_2, b_2))$ yields an OPS of 1 if it is order preserving, i.e., $\Delta_{RSD}(a_1, a_2)$ and $\Delta_{RSD}(b_1, b_2)$ are either both positive or negative; and 0 otherwise. The OPS is determined for all possible correspondence pairs and averaged per alignment. Then, the overall OPS for a matcher is the average of the OPSs of its proposed alignments. The correspondences proposed by the matcher with the highest overall OPS are chosen, along with the according similarity scores. Each correspondence that was not selected, but is in the intersection of the results of the other two matchers. Its similarity is the maximum score yielded by any matcher.

¹¹ We use the stemmer from the JW1 library (<http://projects.csail.mit.edu/jw1/>).

Subsequently, the resulting alignments are revised. For data sets that includes roles, pools, or lanes – like the University Admission data set – a first filtering step removes correspondences where the respective roles mismatch. That is, all correspondences where the role names do not contain at least one overlapping word are removed. Afterwards, the alignments are optimized by a greedy algorithm that maximizes the average similarity of the correspondences and the average OPS for each alignment. It iterates over each correspondence and computes both scores in case the correspondence is removed. The correspondence that improves the scores from the previous iteration is removed. The algorithm will stop once a fixpoint is reached, i.e., there is no further improvement. All remaining correspondences are then returned as the final alignment.

Acknowledgements. This work was partly funded by the German Federal Ministry of Education and Research under the project LSEM (BMBF 03IPT504X). NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

3.10 pPalm-DS

3.10.1 Overview

With this approach, we provide a rudimentary basis for process matching, concentrating on finding an alignment between activities (nodes) with semantic similar labels. We do not consider information like structure, behaviour or different node types within the process models. In a nutshell, from each process p , we retrieve the set of relevant nodes, hereafter called *activities* (node types used for the alignment in the according gold standard). From the set of activities we obtain the according set of labels $l \in L_p$. To compute the matches of a process-pair (p_1, p_2) , we compare each label $l \in L_{p_1}$ to each label $l' \in L_{p_2}$ by a similarity function. If the similarity of both labels is equal or greater than a certain threshold, ($\text{sim}(l, l') \geq \text{threshold}$), we include the corresponding activity-pair to the set of matches.

3.10.2 Specific Techniques

For deriving $\text{sim}(l, l')$ we use, differing from most of the existing approaches, a vector based approach from the field of distributional semantics. The idea behind distributional semantics is the *distributional hypothesis*: “[...] words that occur in similar contexts tend to have similar meaning.”[Pa05]. To be prepared for process models of different domains, we need a large as possible cross-domain set of words with corresponding contexts. A broad coverage of label terms, independent from the basing domain, we ensure by using a corpus of 7.8B words which we derived from Gigaword [Pa11] and the contents of the English Wikipedia (Version 20140102). We utilised word2vec¹² to extract semantic relationships between words and their contexts. Therefore word2vec uses a local context window to capture co-occurrences of words [Mi13]. For each word having a sufficient number of

¹² <https://code.google.com/p/word2vec/>

occurrences within the corpus, this context information is concentrated to a semantic vector having 300 contextual-dimensions. After we trained word2vec on the corpus mentioned before, the resulting database consists of more than 1.6m 300-dimensional semantic vectors.

Finally, we compute $\text{sim}(l, l')$ as follows: Given a process label l consisting of words w_1, \dots, w_n , for each word w_i we collect its vector \mathbf{x}_{w_i} from the database and we perform the element-wise sum to obtain one final vector \mathbf{x}_l for the label. Words missing in the database we treat as null-vectors in the calculation. Given two labels l and l' , we derive similarity by taking the respective final vectors for computing cosine similarity (see [MRS08]):

$$\text{sim}(l, l') = \cos(\theta) = \frac{\mathbf{x}_l \cdot \mathbf{x}_{l'}}{\|\mathbf{x}_l\| \|\mathbf{x}_{l'}\|} = \frac{\sum_{i=1}^n \mathbf{x}_{l,i} \times \mathbf{x}_{l',i}}{\sqrt{\sum_{i=1}^n \mathbf{x}_{l,i}^2} \times \sqrt{\sum_{i=1}^n \mathbf{x}_{l',i}^2}} \quad (5)$$

We include all label pairs having $\text{sim}(l, l') \geq \text{threshold}$ to the final alignment. For this matching contest, we used a threshold of 0.77 which performed best according to the combination of dataset1 and dataset2.

Finally it should be remarked that this approach is not intended as standalone matcher. Rather it aims at being used as basis for further alignments respecting structure, behaviour and different node types within process models.

3.11 Triples

3.11.1 Overview

The matching approach used in the second Process Model Matching Contest in 2015 is essentially the same as the one used in 2013. The Triple-S matching approach [Ca13b] still adheres to the KISS principle by avoiding complex matching techniques and *keeping it simple and stupid*. This years version has been extended to match not only transitions in Petri-Nets but also functions of EPC models and tasks of models in BPMN notation, i.e. the “active” components of process models are matched.

3.11.2 Specific Techniques

The following three levels and scores are considered:

- **Syntactic level - $\text{SIM}_{\text{syn}}(a, b)$:** For the syntactic analysis of active components labels we perform two preprocessing steps: (1) tokenization and (2) stop word elimination. The actual analysis is based on the calculation of Levenshtein [Le66] distances between each combination of tokens (i.e. words) from the labels of active components a and b . The final syntactic score is the minimum distance over all tokens divided by the number of tokens, i.e. the minimum average distance between each token.

- **Semantic level - $SIM_{sem}(a, b)$:** First, we perform the same preprocessing steps as mentioned above. Subsequently, we apply the approach of Wu & Palmer [WP94] to calculate the semantic similarity between each token of labels of active components a and b based on path length between the corresponding concepts. The final semantic score is the maximum average similarity analogous to the final syntactic score.
- **Structural level - $SIM_{struc}(a, b)$:** At this level, we investigate the similarity of active components a and b through a comparison of (i) the ratio of their in- and outgoing arcs and (ii) their relative position in the complete model. The two values are combined through the calculation of a weighted average.

These three scores are combined to the final score $SIM_{total}(a, b)$ which represents the matching degree between two active components a and b from different process models. It is calculated according to the following formula:

$$SIM_{total}(a, b) = \omega_1 * SIM_{syn}(a, b) + \omega_2 * SIM_{sem}(a, b) + \omega_3 * SIM_{struc}(a, b)$$

The three parameters ω_1 , ω_2 and ω_3 define the weight of each similarity level. A threshold value θ is used to determine whether active components actually match, i.e. iff $SIM_{total} \geq \theta$, two transitions positively match.

3.11.3 Implementation

The Triple-S approach has been implemented using Java. For the calculation of the semantic score with the approach of Wu & Palmer, the *WS4J Java API*¹³ has been used to query Princeton's English *WordNet* 3.0 lexical database [Mi95]. Relative positions of transitions are calculated using the implementation of Dijkstras algorithm by Vogella¹⁴.

During our experiments we tried to approximate optimal results based on the gold standard examples. For the contest, we have used the following values: $\omega_1 = 0.5$, $\omega_2 = 0.35$, $\omega_3 = 0.15$ and $\theta = 0.7$. Thereby, the weights for $SIM_{struc}(a, b)$ have been set to 0.25 for value (i) and 0.75 for value (ii).

Acknowledgement. This work has been developed with the support of DFG (German Research Foundation) under the project SemReuse OB 97/9-1.

4 Results

For assessing the submitted process model matching techniques, we compare the computed correspondences against a manually created gold standard. Using the gold standard, we classify each computed activity match as either true-positive (TP), true-negative (TN), false-positive (FP) or false-negative (FN). Based on this classification, we calculate the

¹³ <https://code.google.com/p/ws4j/>

¹⁴ <http://www.vogella.com/articles/JavaAlgorithmsDijkstra/article.html>

precision ($TP/(TP+FP)$), the recall ($TP/(TP+FN)$), and the f-measure, which is the harmonic mean of precision and recall ($2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$).

Tables 3 to 6 give an overview of the results for the datasets. For getting a better understanding of the result details, we report the average (\emptyset) and the standard deviation (SD) for each metric. The highest value for each metric is marked using bold font. In our evaluation we distinguish between micro and macro average. Macro average is defined as the average of precision, recall and f-measure scores over all testcases. On the contrary, micro average is computed by summing up TP, TN, FP, and FN scores applying the precision, recall and f-measure formula once on the resulting values. Micro average scores take different sizes of test cases into account, e.g., bad recall on a small testcase has only limited impact on the micro average recall scores.

Some agreements are required to compute macro average scores for two special cases. It might happen that a matcher generates an empty set of correspondences. If this is the case, we set the precision score for computing the macro average to 1.0, due to the consideration that an empty set of correspondences contains no incorrect correspondences. Moreover, some of the testcases of the AM data set have empty gold standards. In this case we set the recall score for computing the macro average to 1.0, because all correct matches have been detected.

Approach	Precision			Recall			F-Measure		
	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD
RMM/NHCM	.686	.597	.248	.651	.61	.277	.668	.566	.224
RMM/NLM	.768	.673	.261	.543	.466	.279	.636	.509	.236
MSSS	.807	.855	.232	.487	.343	.353	.608	.378	.343
OPBOT	.598	.636	.335	.603	.623	.312	.601	.603	.3
KMSSS	.513	.386	.32	.578	.402	.357	.544	.374	.305
RMM/SMSL	.511	.445	.239	.578	.578	.336	.543	.477	.253
TripleS	.487	.685	.329	.483	.297	.361	.485	.249	.278
BPLangMatch	.365	.291	.229	.435	.314	.265	.397	.295	.236
KnoMa-Proc	.337	.223	.282	.474	.292	.329	.394	.243	.285
AML-PM	.269	.25	.205	.672	.626	.319	.385	.341	.236
RMM/VM2	.214	.186	.227	.466	.332	.283	.293	.227	.246
pPalm-DS	.162	.125	.157	.578	.381	.38	.253	.18	.209

Tab. 3: Results of University Admission Matching

The results for the UA data set (Table 3) illustrate large differences in the quality of the generated correspondences. Note that we ordered the matchers in Table 3 and in the other results tables by micro average f-measure. The best results in terms of f-measure (micro-average) are obtained by the RMM/NHCM approach (0.668) followed by RMM/NLM (0.636) and MSSS (0.608). At the same time three matching systems generate results with an f-measure of less than 0.4. When we compare these results against the results achieved in the 2013 edition of the contest, we have to focus on macro-average scores, which have been computed also in the 2013 edition. This year, there are several matchers with a macro average of >0.5 , while the best approach achieved 0.41 in 2013. This improvement indicates that the techniques for process matching have progressed over the last two years. Anyhow, we also have to take into account that the gold standard has been improved and the format of the models has been changed to BPMN. Thus, results are only partially comparable.

Comparing micro and macro f-measure averages in 2015, there are, at times, significant differences. In most cases, macro scores are significantly lower. This is caused by the existence of several small testcases (small in numbers of correspondences) in the collection that seem to be hard to deal with for some matchers. These testcases have a strong negative impact on macro averages and a moderated impact on micro average. This is also one of the reasons why we prefer to discuss the results in terms of micro average.

It is interesting to see that the good results are not only based on a strict setting that aims for high precision scores, but that matchers like RMM/NHCM and OPBOT manage to achieve good f-measure scores based on well-balanced precision/recall scores. Above, we have described the gold standard of this data set as rather strict in terms of 1:n correspondences. This might indicate that the matching task should not be too complex. However, some of the approaches failed to generate good results. Note that this is caused by a low precision, while at the same time recall values have not or only slightly been affected positively. A detailed matcher specific analysis, that goes beyond the scope of this paper, has to reveal the underlying reason.

Approach	Precision			Recall			F-Measure		
	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD
RMM/NHCM	.855	.82	.194	.308	.326	.282	.452	.424	.253
OPBOT	.744	.776	.249	.285	.3	.254	.412	.389	.239
RMM/SMSL	.645	.713	.263	.277	.283	.217	.387	.36	.205
KMSSS	.64	.667	.252	.273	.289	.299	.383	.336	.235
AML-PM	.385	.403	.2	.365	.378	.273	.375	.363	.22
KnoMa-Proc	.528	.517	.296	.282	.281	.278	.367	.319	.25
BPLangMatch	.545	.495	.21	.247	.256	.228	.34	.316	.209
RMM/NLM	.787	.68	.267	.211	.229	.308	.333	.286	.299
MSSS	.829	.862	.233	.19	.212	.312	.309	.255	.318
TripleS	.543	.716	.307	.205	.224	.336	.297	.217	.284
RMM/VM2	.327	.317	.209	.27	.278	.248	.296	.284	.226
pPalm-DS	.233	.273	.163	.316	.328	.302	.268	.25	.184

Tab. 4: Results of University Admission Matching with Subsumption

The results for the UA data set where we used the extended gold standard including subsumption correspondences are shown in Table 4. Due to the experimental status of this gold standard the results shown are thus less conclusive. However, we decided finally to include these results because subsumption correspondences will often occur when two process models differ in terms of granularity. A comparison against the strict version of the gold standard (Table 3) reveals that there are some slight changes in the f-measure based ordering of the matchers. OPBOT climbs up from rank #4 to rank #2, AML-PM climbs from up from rank #10 to rank #5, while other matchers are only slightly affected. This shows that some of the implemented methods can be used to detect subsumption correspondences, while other techniques are in particular designed to focus on direct 1:1 correspondences only.

The BR data set has not been modified compared to its 2013 version. Thus, we can directly compare the 2015 results against the 2013 results. Again, we have to focus on the macro average scores. In 2013, the top results were achieved by RefMod-Mine/NSCM with an macro average f-measure of 0.45. In 2015 the best performing matcher on this data set is the

Approach	Precision			Recall			F-Measure		
	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD
OPBOT	.713	.679	.184	.468	.474	.239	.565	.54	.216
pPalm-DS	.502	.499	.172	.422	.429	.245	.459	.426	.187
RMM/NHCM	.727	.715	.197	.333	.325	.189	.456	.416	.175
RMM/VM2	.474	.44	.2	.4	.397	.241	.433	.404	.21
BPLangMatch	.645	.558	.205	.309	.297	.22	.418	.369	.221
AML-PM	.423	.402	.168	.365	.366	.186	.392	.367	.164
KMSSS	.8	.768	.238	.254	.237	.238	.385	.313	.254
RMM/SMSL	.508	.499	.151	.309	.305	.233	.384	.342	.178
TripleS	.613	.553	.26	.28	.265	.264	.384	.306	.237
MSSS	.922	.972	.057	.202	.177	.223	.332	.244	.261
RMM/NLM	.859	.948	.096	.189	.164	.211	.309	.225	.244
KnoMa-Proc	.234	.217	.188	.297	.278	.234	.262	.237	.205

Tab. 5: Results of Birth Certificate Matching

OPBOT approach with macro average f-measure of 0.54, which is a significant improvement compared to 2013. The systems on the follow-up positions, which are pPalm-DS (0.426), RMM/NHCM (0.416), and RMM/VM2 (0.402), could not outperform the 2013 results. However, the average approach (≈ 0.35) in 2015 is clearly better than the average approach in 2013 (≈ 0.29), which can be understood as an indicator for an overall improvement.

While it is possible for the UA data set to generate high f-measures with a balanced approach in terms of precision and recall, the BR data set does not share this characteristics. All matchers, with the exception of KnoMa-Proc, favor precision over recall. Moreover, a high number of non-trivial correspondences cannot be found by the participants of the contest. We conducted an additional analysis where we computed the union of all matcher generated alignments. For this alignment we measured a recall of 0.631. This means that there is a large fraction of non-trivial correspondences in the BR data set that cannot be found by any of the matchers. Note that we measured the analogous score also for the other data sets, with the outcome of 0.871 for the UA dataset (0.494 for the extended UA data set) and 0.68 for the AM data set. These numbers illustrate that the BR data set is a challenging data set, which requires specific methods to overcome low recall scores. This can also be the reason why some of the systems that perform not so well on the UA data set are among the top-5 systems for the BR data set. These systems are OPBOT, pPalm-DS, and RMM/VM2.

The results for the AM data set are presented in Table 6. The top performing matchers in terms of macro f-measure are AML-PM (0.677), RMM/NHCM (0.661), and RMM/NLM (0.653). While these systems are close in terms of f-measure, they have a different characteristics in terms of precision and recall. The two RMM-based systems have a high precision in common. Especially RMM/NLM has a precision of 0.991, which means that less than 1 out of 100 correspondences are incorrect. AML-PM, the top performing system, has only a precision of .786 and a (relatively high) recall of .595. It is notable that these results have been achieved by the use a standard ontology matching systems instead of using a specific approach for process model matching. For the details we refer the reader to the respective system description in the previous section. The best results in terms of recall have been

Approach	Precision			Recall			F-Measure		
	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD	\emptyset -mic	\emptyset -mac	SD
AML-PM	.786	.664	.408	.595	.635	.407	.677	.48	.422
RMM/NHCM	.957	.887	.314	.505	.521	.422	.661	.485	.426
RMM/NLM	.991	.998	.012	.486	.492	.436	.653	.531	.438
BPLangMatch	.758	.567	.436	.563	.612	.389	.646	.475	.402
OPBOT	.662	.695	.379	.617	.634	.409	.639	.514	.403
MSSS	.897	.979	.079	.473	.486	.432	.619	.519	.429
RMM/VM2	.676	.621	.376	.545	.6	.386	.603	.454	.384
KMSSS	.643	.834	.282	.527	.532	.417	.579	.482	.382
TripleS	.614	.814	.261	.545	.546	.434	.578	.481	.389
pPalm-DS	.394	.724	.348	.595	.615	.431	.474	.451	.376
KnoMa-Proc	.271	.421	.383	.514	.556	.42	.355	.268	.279
RMM/SMSL	.722	.84	.307	.234	.37	.366	.354	.333	.327

Tab. 6: Results of Asset Management Matching

achieved by the OPBOT matcher (0.617). Looking at the recall scores in general, it can be concluded that it is hard to top a recall of 0.6 without a significant loss in precision.

The results of our evaluation show that there is a high variance in terms of the identified correspondences across the different data sets. However, there are also some systems that perform well over all three data sets (we exclude the UA_S data set in this consideration due to its experimental character). These systems are RMM/NHCM and OPBOT. RMM/NHCM is ranked #1, #3 and #2, OPBOT is ranked #4, #1, and #5 in terms of macro-average. None of the other approaches is among the top-five with respect to all three data sets. This illustrates again how hard it is to propose a mechanism that works well for the different modeling styles and labeling conventions that can be found in our test data collection.

5 Conclusion

In this paper, we reported on the setup and the results of the Process Model Matching Contest 2015. We provided three different process model matching problems and received automatically generated results of twelve different techniques. The high number of participants showed that there is a vivid process matching community that is interested in an experimental evaluation of the developed techniques to better understand its pros and cons. We are also happy that we were able to attract participants from the ontology community (e.g., AML-PM). We believe that both research fields (Process Model Matching and Ontology Matching) are closely related and can mutually benefit from each other in the future.

The results of our experimental evaluation show that there is an overall improvement compared to the results that have been achieved in 2013. This becomes obvious from the results of the UA and BR data set. The underlying reasons for these improvements cannot be detailed in the aggregated view that we presented in the results section. It requires a detailed analysis of the techniques implemented by the top performing approaches which are presented in Section 3. However, the presented results can give some hints on the

different characteristics of the proposed approaches, which helps to better understand the concrete impact on a given matching task.

The results show also that many proposed approaches do not generate good results for all data sets. An counterexample for our claim are the two matching systems RMM/NHCM and OPBOT. These two systems are among the top-5 systems for all data sets used in our evaluation. However, the results also show that the test data collection is heterogeneous in terms of specifics that need to be considered and problems that need to be solved for generating high quality correspondences. Especially the BR data set seems to be challenging. Here we discovered that more than 36% of all correspondences in the gold standard have not been generated by any of the participating matchers. This number shows that there is still large room for improvement related to methods that aim at a high recall without suffering too much in terms of precision.

For 2015 we did not define a strict set of rules for participation. However, this creates a certain bias in the comparison of the results. In particular, we have noticed that a wide range of different techniques have been proposed, including approaches that rely on joint matching of all process models from a data set as well as supervised machine learning techniques. All these techniques have been described precisely and in a transparent way. Yet, they postulate slightly different settings for process model matching, which are all reasonable from an application point of view, but shall be evaluated separately. That is, results obtained when relying solely on the two models to be matched may differ significantly from results obtained when considering a whole corpus of process models that should be aligned. Hence, in future editions, we plan to evaluate these scenarios separately.

For the future we consider establishing a fully automated evaluation procedure similar to the one that is applied since 2011/2012 in the context of the Ontology Alignment Evaluation Initiative [Ag12]. In such a setting, the matching tools are submitted to the organizers instead of submitting the generated correspondences. The submitted tools are then executed by the organizers in a controlled environment. Such an evaluation approach has several advantages. First, the generated results are a 100% reproducible and it can be guaranteed that no data set specific parameter settings have been chosen. Second, the matching tools themselves become available as executable tools. So far, they often represent academic prototypes that are not available to the public. We believe that this is an important step for the adoption of process matching tools to solve real world matching problems.

References

- [Ag12] Aguirre, José Luis; Grau, Bernardo Cuenca; Eckert, Kai; Euzenat, Jérôme; Ferrara, Alfio; Van Hague, Robert Willem; Hollink, Laura; Jiménez-Ruiz, Ernesto; Meilicke, Christian; Nikolov, Andriy et al.: Results of the ontology alignment evaluation initiative 2012. In: Proc. 7th ISWC workshop on ontology matching (OM). No commercial editor., pp. 73–115, 2012.
- [Br12] Branco, Moisés Castelo; Troya, Javier; Czarnecki, Krzysztof; Küster, Jochen Malte; Völzer, Hagen: Matching Business Process Workflows across Abstraction Levels. In (France, Robert B.; Kazmeier, Jürgen; Breu, Ruth; Atkinson, Colin, eds): MoDELS. volume 7590 of Lecture Notes in Computer Science. Springer, pp. 626–641, 2012.

- [Ca13a] Cayoglu, Ugur; Dijkman, Remco; Dumas, Marlon; Fettke, Peter; Garcia-Banuelos, Luciano; Hake, Philip; Klinkmüller, Christopher; Leopold, Henrik; Ludwig, André; Loos, Peter et al.: The process model matching contest 2013. In: 4th International Workshop on Process Model Collections: Management and Reuse (PMC-MR'13). 2013.
- [Ca13b] Cayoglu, Ugur; Oberweis, Andreas; Schoknecht, Andreas; Ullrich, Meike: Triple-S: A Matching Approach for Petri Nets on Syntactic, Semantic and Structural level. Technical report, 2013.
- [CD07] Cunningham, P.; Delany, S. J.: k-Nearest neighbour classifiers. *Multiple Classifier Systems*, pp. 1–17, 2007.
- [DGBD09] Dumas, Marlon; García-Bañuelos, Luciano; Dijkman, Remco M.: Similarity Search of Business Process Models. *IEEE Data Eng. Bull.*, 32(3):23–28, 2009.
- [Di09] Dijkman, R.; Dumas, M.; Garcia-Banuelos, L.; Kaarik, R.: Aligning Business Process Models. In: *Enterprise Distributed Object Computing Conference*. 2009.
- [Dr14] Dragisic, Zlatan; Eckert, Kai; Euzenat, Jérôme; Faria, Daniel; Ferrara, Alfio; Granada, Roger; Ivanova, Valentina; Jiménez-Ruiz, Ernesto; Kempf, Andreas Oskar; Lambrix, Patrick et al.: Results of the ontology alignment evaluation initiative 2014. In: *Proceedings of the 9th International Workshop on Ontology Matching Collocated with the 13th International Semantic Web Conference (ISWC 2014)*. 2014.
- [Dr15] Dragoni, Mauro: Exploiting Multilinguality For Creating Mappings Between Thesauri. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. SAC 2015. ACM, pp. 382–387, 2015.
- [Ek12] Ekanayake, Chathura C.; Dumas, Marlon; García-Bañuelos, Luciano; Rosa, Marcello La; ter Hofstede, Arthur H. M.: Approximate Clone Detection in Repositories of Business Process Models. In (Barros, Alistair P.; Gal, Avigdor; Kindler, Ekkart, eds): *BPM*. volume 7481 of *Lecture Notes in Computer Science*. Springer, pp. 302–318, 2012.
- [ES07] Euzenat, Jérôme; Shvaiko, Pavel: *Ontology Matching*. Springer-Verlag New York, Inc, Secaucus, NJ, USA, 2007.
- [Fa13] Faria, Daniel; Pesquita, Catia; Santos, Emanuel; Cruz, Isabel F; Couto, Francisco M: AgreementMakerLight results for OAEI 2013. In: *OM*. pp. 101–108, 2013.
- [GS10] Gal, Avigdor; Sagi, Tomer: Tuning the ensemble selection process of schema matchers. *Inf. Syst.*, 35(8):845–859, 2010.
- [Ji13] Jin, Tao; Wang, Jianmin; Rosa, Marcello La; ter Hofstede, Arthur H.M.; Wen, Lijie: Efficient querying of large process model repositories. *Computers in Industry*, 64(1), 2013.
- [JMF99] Jain, A.K.; Murty, M.N.; Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31:264–323, 1999.
- [KKR06] Küster, Jochen Malte; Koehler, Jana; Ryndina, Ksenia: Improving Business Process Models with Reference Models in Business-Driven Development. In (Eder, Johann; Dustdar, Schahram, eds): *Business Process Management Workshops*. volume 4103 of *Lecture Notes in Computer Science*. Springer, pp. 35–44, 2006.
- [K113] Klinkmüller, Christopher; Weber, Ingo; Mendling, Jan; Leopold, Henrik; Ludwig, André: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In (Daniel, Florian; Wang, Jianmin; Weber, Barbara, eds): *Business Process Management*, volume 8094 of *Lecture Notes in Computer Science*, pp. 211–218. Springer Berlin Heidelberg, 2013.

- [Kl14] Klinkmüller, Christopher; Leopold, Henrik; Weber, Ingo; Mendling, Jan; Ludwig, André: Listen to Me: Improving Process Model Matching through User Feedback. In: Business Process Management. pp. 84–100, 2014.
- [KWW11] Kunze, Matthias; Weidlich, Matthias; Weske, Mathias: Behavioral Similarity - A Proper Metric. In (Rinderle-Ma, Stefanie; Toumani, Farouk; Wolf, Karsten, eds): BPM. volume 6896 of Lecture Notes in Computer Science. Springer, pp. 166–181, 2011.
- [La13] La Rosa, Marcello; Dumas, Marlon; Uba, Reina; Dijkman, Remco: Business Process Model Merging: An Approach to Business Process Consolidation. ACM Trans. Softw. Eng. Methodol., 22(2):11:1–11:42, 2013.
- [Le66] Levenshtein, Vladimir: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 10(8):707–710, 1966.
- [Li91] Lin, Jianhua: Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory, 37(1):145–151, 1991.
- [Li98] Lin, Dekang: An Information-Theoretic Definition of Similarity. In: International Conference on Machine Learning. pp. 296–304, 1998.
- [LZ09] Lv, Yuanhua; Zhai, ChengXiang: Positional language models for information retrieval. In (Allan, James; Aslam, Javed A.; Sanderson, Mark; Zhai, ChengXiang; Zobel, Justin, eds): SIGIR. ACM, pp. 299–306, 2009.
- [Ma14] Manning, Christopher D; Surdeanu, Mihai; Bauer, John; Finkel, Jenny; Bethard, Steven J; McClosky, David: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60, 2014.
- [Mi95] Miller, George A.: WordNet: A Lexical Database for English. Communications of the ACM, 38(11):39–41, 1995.
- [Mi13] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR, abs/1301.3781, 2013.
- [MRS08] Manning, C. D.; Raghavan, P.; Schütze, H.: Introduction to information retrieval. Cambridge University Press, New York, 2008.
- [Na09] Navigli, Roberto: Word Sense Disambiguation: A Survey. ACM Computing Surveys, 41(2):10:1–10:69, 2009.
- [NH15] Niesen, T.; Houy, C.: Zur Nutzung von Techniken der Natürlichen Sprachverarbeitung für die Bestimmung von Prozessmodellähnlichkeiten – Review und Konzeptentwicklung. In (Thomas, O; Teuteberg, F, eds): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik (WI-15). Springer, Osnabrück, pp. 1829–1843, 2015.
- [Pa05] Pantel, P.: Inducing Ontological Co-occurrence Vectors. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 125–132, 2005.
- [Pa11] Parker, R.: English gigaword fifth edition. Linguistic Data Consortium, [Philadelphia, PA], 2011.
- [Po97] Porter, M.F.: An algorithm for suffix stripping. Readings in information retrieval, pp. 313–316, 1997.

- [Ra03] Ramos, J.: Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning, 2003.
- [Ra11] Rahm, Erhard: Towards Large-Scale Schema and Ontology Matching. In: Schema Matching and Mapping. pp. 3–27, 2011.
- [To03] Toutanova, Kristina; Klein, Dan; Manning, Christopher D.; Singer, Yoram: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: IN PROCEEDINGS OF HLT-NAACL. pp. 252–259, 2003.
- [Wa06] Wallach, H. M.: Topic modeling: beyond bag-of-words. Proceedings of the 23rd international conference on Machine learning, 2006.
- [We13] Weidlich, Matthias; Sheetrit, Eitam; Branco, Moises; Gal, Avigdor: Matching Business Process Models Using Positional Language Models. In: 32nd International Conference on Conceptual Modeling, ER 2013. Hong Kong, 2013.
- [WMW11] Weidlich, Matthias; Mendling, Jan; Weske, Mathias: A Foundational Approach for Managing Process Variability. In (Mouratidis, Haralambos; Rolland, Colette, eds): CAiSE. volume 6741 of Lecture Notes in Computer Science. Springer, pp. 267–282, 2011.
- [WP94] Wu, Zhibiao; Palmer, Martha: Verbs Semantics and Lexical Selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 133–138, 1994.

GI-Edition Lecture Notes in Informatics

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheimer (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher und Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheimer, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan 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 Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoît 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 Rannenber, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, 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 Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006
- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-tern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS'06
- P-82 Heinrich C. Mayr, Ruth 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 Magenheimer, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik
- P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis
- P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 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 management 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 Special Interest Groups PARS and PARVA
- P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung
- P-126 Rainer 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 Dreö Rodosek (Hrsg.)
1. DFN-Forum Kommunikationstechnologien 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 und E-Voting. CC
- P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik
- P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 1
- P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 2
- P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse
- P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008
- P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures
- P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008
- P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik
- P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics
- P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management
- P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung
- P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

- P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)
- P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management
- P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)
- P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009
- P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I²CS 2009
- P-149 Paul Müller, Bernhard Neumair, Gabi Dreö 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 Brömmе, 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. Claudepein, 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 Brömmе, 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 (I²CS) – Jubilee Edition 2010 –
- P-166 Paul Müller, Bernhard Neumair, Gabi Dreö 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 Fährnich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1
- P-176 Klaus-Peter Fährnich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2
- P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fährnich, 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 (I²CS)
- P-187 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.)
4. DFN-Forum Kommunikationstechnologien, 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 Poustchi, Frédéric Thiesse (Hrsg.)
MMS 2012: Mobile und Ubiquitäre Informationssysteme
- P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreö Rodosek (Hrsg.)
5. DFN-Forum Kommunikationstechnologien
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 und 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 automobilen 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 Brömmel, 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 Mitschang, 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 Dreö Rodosek (Hrsg.)
6. DFN-Forum Kommunikationstechnologien
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 Dreö 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 Sassi,at,
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
Ottjacques (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
- P-236 Martin Engstler, Eckhart Hanser,
Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik der
Gesellschaft für Informatik e.V., Stuttgart
2014
- P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany
- P-238 Arno Ruckelshausen, Hans-Peter
Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim
- P-239 Uwe Aßmann, Birgit Demuth, Thorsten
Spitta, Georg Püschel, Ronny Kaiser
(Hrsg.)
Software Engineering & Management
2015
17.-20. März 2015, Dresden
- P-240 Herbert Klenk, Hubert B. Keller, Erhard
Plödereder, Peter Dencker (Hrsg.)
Automotive – Safety & Security 2015
Sicherheit und Zuverlässigkeit für
automobile Informationstechnik
21.–22. April 2015, Stuttgart
- P-241 Thomas Seidl, Norbert Ritter,
Harald Schöning, Kai-Uwe Sattler,
Theo Härder, Steffen Friedrich,
Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg
- P-242 Norbert Ritter, Andreas Henrich,
Wolfgang Lehner, Andreas Thor,
Steffen Friedrich, Wolfram Wingerath
(Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015) –
Workshopband
02. – 03. März 2015, Hamburg
- P-243 Paul Müller, Bernhard Neumair, Helmut
Reiser, Gabi Dreö Rodosek (Hrsg.)
8. DFN-Forum
Kommunikationstechnologien
06.–09. Juni 2015, Lübeck

- P-244 Alfred Zimmermann,
Alexander Rossmann (Eds.)
Digital Enterprise Computing
(DEC 2015)
Böblingen, Germany June 25-26, 2015
- P-246 Douglas W. Cunningham, Petra Hofstedt,
Klaus Meer, Ingo Schmitt (Hrsg.)
INFORMATIK 2015
28.9.-2.10. 2015, Cottbus
- P-247 Hans Pongratz, Reinhard Keil (Hrsg.)
DeLFI 2015 – Die 13. E-Learning
Fachtagung Informatik der
Gesellschaft für Informatik e.V. (GI)
1.-4. September 2015
München
- P-248 Jens Kolb, Henrik Leopold, Jan
Mendling (Eds.)
Enterprise Modelling and
Information Systems Architectures
Proceedings of the 6th Int. Workshop
on Enterprise Modelling and
Information Systems Architectures,
Innsbruck, Austria
September 3-4, 2015
- P-249 Jens Gallenbacher (Hrsg.)
Informatik
allgemeinbildend begreifen
INFOS 2015 16. GI-Fachtagung
Informatik und Schule
20.–23. September 2015

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

