Gesellschaft für Informatik e.V. (GI)

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

Broken down into
- seminars
- proceedings
- dissertations
- thematics

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

The volumes are published in German or English.

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

ISSN 1617-5468
ISBN 978-3-88579-600-8

This volume contains the Proceedings of EMISA 2012 which is the annual workshop of the special interest group EMISA in the Gesellschaft für Informatik e.V. (GI). EMISA 2012 takes place from September 13 to 14 at the University of Vienna, Austria. The special theme of EMISA 2012 is "People in the center of Modeling – Der Mensch im Zentrum der Modellierung". This volume features 11 high-quality papers centering around the special theme addressing issues such as visualization, modeling guidelines, languages, and tools.

Stefanie Rinderle-Ma, Mathias Weske (Hrsg.): EMISA 2012

206

GI-Edition

Lecture Notes in Informatics

Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)

EMISA 2012
Der Mensch im Zentrum der Modellierung

13. – 14. September 2012
Wien

Proceedings

Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)

EMISA 2012

Der Mensch im Zentrum der Modellierung

13. – 14. September
in Wien, Österreich

Gesellschaft für Informatik e.V. (GI)

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

Volume P-206

ISBN 978-3-88579-600-8
ISSN 1617-5468

Volume Editors
Univ.-Prof. Dr. Stefanie Rinderle-Ma
Forschungsgruppe Workflow Systems and Technology
Universität Wien
A-1090 Wien
Email: stefanie.rinderle-ma@univie.ac.at
Prof. Dr. Mathias Weske
Hasso-Plattner-Institute für IT Systems Engineering
Universität Potsdam
Prof.-Dr,-Helmertstr. 2-3
D-14482 Potsdam
Email: mathias.weske@hpi.uni-potsdam.de

printed by Köllen Druck+Verlag GmbH, Bonn

EMISA 2012 Workshop

Message From the PC Chairs

The EMISA 2012 workshop takes place from 13 to 14 of September 2012 at the University of Vienna, Austria, in conjunction with the 4th International Workshop on the Business Process Model and Notation, the BPMN Anwendertag (both organized by the Vienna University of Economics and Business), and the AWPN Workshop.

The EMISA workshop series addresses all questions regarding the modeling and design of Information Systems and their applications. This year's issue has the special theme "People in the center of Modeling – Der Mensch im Zentrum der Modellierung". This human-centered approach covers all phases of the Information System life cycle, starting from adequate modeling methods over interaction with users during run time to the evolution of Information Systems. It aims at including all stakeholders of an Information System such as people working with the system, modelers, and analysts.

The EMISA 2012 received 19 full paper submissions. The review process was thorough resulting in 3 reviews per paper on the average. The discussions among the reviewers were very lively, reflecting the high interest in the research topics addressed and the commitment of the PC members. Finally, 11 full papers could be accepted that will be presented in four sessions called Languages & Extensions, Modeling Guidelines, Visualization & Tools, and Audit & Conformance.

We want to thank all people who contributed to make the EMISA 2012 workshop a success. Specifically, we are grateful for the timely and detailed reviews and active discussions by the PC members. We would also like to thank Jan Mendling as General Co-Chair of this year's joint EMISA / BPMN / BPMN Anwendertag / AWPN event as well as the Organization Chairs Maria Leitner and Ralph Vigne for their excellent work.

August 2012

Stefanie Rinderle-Ma
Mathias Weske
EMISA 2012 PC Chairs

Leitung des Programmkomitees

Stefanie Rinderle-Ma, University of Vienna
Mathias Weske, University of Potsdam

Programmkomitee

Thomas Allweyer, FH Kaiserslautern
Patrick Delfmann, European Research Center for Information Systems
Jörg Desel, Fernuniversität in Hagen
Fernand Feltz, Centre de Recherche Public - Gabriel Lippmann
Andreas Gadatsch, Hochschule Bonn-Rhein-Sieg
Wilhelm Hasselbring, Universität Kiel
Heinrich Jasper, Technical University Bergakademie Freiberg
Reinhard Jung, University of St. Gallen
Stefan Klink, Karlsruher Institut für Technologie
Ralf Klischewski, German University in Cairo
Horst Kremers, CODATA-Germany
Peter Loos, IWi at DFKI, Saarland University
Heinrich C. Mayr, Universität Klagenfurt
Jan Mendling, Wirtschaftsuniversität Wien
Bela Mutschler, University of Applied Sciences Ravensburg-Weingarten
Markus Nüttgens, Universität Hamburg
Andreas Oberweis, Universitaet Karlsruhe
Sven Overhage, University of Augsburg
Hansjuergen Paul, Institut Arbeit und Technik
Erik Proper, Public Research Centre - Henri Tudor
Manfred Reichert, University of Ulm
Stefanie Rinderle-Ma, University of Vienna
Michael Rosemann, Queensland University of Technology
Frank Rump, FH Oldenburg/Ostfriesland/Wilhelmshaven
Eike Schallehn, University of Magdeburg
Carlo Simon, Provadis School of International Management and Technology
Stefan Strecker, Fernuniversität in Hagen
Oliver Thomas, Chair in Information Management and Information Systems, University of Osnabrueck
Klaus Turowski, Otto-von-Guericke-University Magdeburg
Gottfried Vossen, ERCIS Muenster
Barbara Weber, Univ. of Innsbruck

Hans Weigand, Tilburg University
Mathias Weske, University of Potsdam
Robert Winter,University of St. Gallen

Weitere Gutachter

Dirk van der Linden
Peer C. Brauer
Martin Schultz
Michel Matijacic
Stephan Aier
Tobias Hildebrandt
Yannic Domigall
Thomas Tamisier
Nicolas Mundbrod
Reinhold Dunkl
Michael Fellmann

Lokale Organisation

Maria Leitner, Universität Wien
Ralph Vigne, Universität Wien

Sponsoren

Inhaltsverzeichnis

Languages & Extensions

Modeling Guidelines

Visualization and Tools

Audit & Conformance

Workflow Charts and Their Semantics Using Abstract State Machines

Theodorich Kopetzky, Verena Geist

Software Competence Center Hagenberg GmbH
Softwarepark 21, 4232 Hagenberg, Austria
{theodorich.kopetzky,verena.geist}@scch.at

Abstract: Workflow charts are a novel way to describe business processes and a way of putting more emphasis on the human-computer interaction. They introduce a typed approach to workflow specification in order to improve flexibility in business process technology by proposing a concept for integrating workflow definition and dialogue programming, being also open to business process modelling. Although their precursor has its semantic specification based on Unified Modelling Language semantics, workflow charts currently lack a complete formal semantic specification. Instead of enhancing the Unified Modelling Language semantics, the approach of specifying the semantics of workflow charts using the formalism of Abstract State Machines has been chosen. The resulting description provides a precise operational semantics.

1 Introduction

Business process-related topics are an active field of research, one subset being the different languages to describe business processes. While many languages are more focused on the control flow of the business process, e.g. Yet Another Workflow Language (YAWL) [vdAtH05] or the Business Process Model and Notation (BPMN) [OMG11], there are languages which focus on different aspects, for example subject-oriented process management [FSS$^+$11], where the focus is more on the subjects and their tasks. Modeling of the user interaction is usually not in the focus of those languages.

While workflow charts [Dra10] still model the control flow they put more emphasis on user interaction by implementing a submit/response-style user interaction pattern. The interaction with a submit/response-style system consists in a continuous interchange of report presentations and form submissions. This user interface paradigm is widely used in form-based applications, ranging from simple web applications to complex ERP systems. The concept of workflow charts relies on a methodology to model form-based applications called formcharts [DW04]. However, they lack a complete formal semantic specification. To meet the needs of business processes, formcharts are extended to workflow charts in due consideration of the worklist paradigm, concurrency, and actors as well as roles. A formalisation of workflow charts will provide a precise operational semantics. Thus, workflow charts represent a technology-independent, conceptual modelling language and are at the same time designed for integrating workflow definitions and dialogue programming to create an executable specification language, providing a more flexible process technology [Gei11]. The reason for creating such formalisation is twofold. On the one side, we want to be able to reason about workflow charts and simulate models without writing a prototype in a conventional programming language. The implementation with CoreASM (see Section 4 on page 11) enables us to do that in a short and concise way. On the other side, seeing how languages as BPMN lack a precise formal specification and the problems that may lead to (e.g., contradictions in the BPMN-specification in [Nat11]), or Unified Modelling Language (UML) which has unclarities in its semantics [FSKdR05], such a specification seems the prudent way to avoid those problems.

In this paper we formalise the concept of workflow charts using Abstract State Machines (ASMs). In Section 2, we first discuss the nature of workflow charts, using a descriptive example. In Section 3.1 we give a very short introduction to ASMs. In Section 3, we discuss the semantics of workflow charts using ASMs including descriptions of the assumed environment, required nodes and associations, and miscellaneous static functions. We also provide the definition of the ASM agent that operates on a given workflow chart. A brief overview of the implementation of the specified semantics using CoreASM is given in Section 4. In Section 5, we provide a short survey of related work. We sum our main findings up in Section 6 and also comment on the future work.

2 Workflow Charts

Workflow charts as a modelling language for business processes have been introduced in [Dra10]. They are specified as a tripartite graph, follow a submit/response-style user interaction pattern, and consist of the following node types:

Client Pages (Client Pages (CPs)) represent computer screens that show information and provide forms for user input. They are specified using ellipses in workflow charts.

Immediate Server Actions (ISAs) represent forms that appear on computer screens and are specified using rectangles.

Deferred Server Actions (DSAs) appear as links in the worklists of the corresponding actors and are specified as rectangles with a gradient fill.

The edges, or associations, have a twofold meaning: On the one hand, they communicate the flow of control through the graph and, on the other hand, they function as associations defining which data is to be presented to an actor. The associations are usually guarded by differently named kinds of conditions, which correspond to guards in the ASM ground model. A very simple workflow charts graph can be seen in Fig. 1 on the next page.

One of the fundamental metaphors of workflow charts is the worklist metaphor. The worklist is the one point where actors in a workflow[1] choose the next step in the workflow to

[1] We use the notion of a workflow as an executable business process specification synonymously to the notion of a business process. This seems more natural here as this word is part of the name "workflow charts", too.

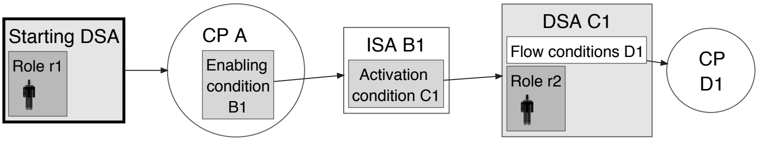


Figure 1: A very simple workflow chart.

execute. Each actor has his own worklist resp. his own view of the global worklist. The worklist is divided into two areas, i.e., the workflow area and the task area. The workflow area (start menu) contains all registered workflows that can be started by the actor. The entries in the task area are related to DSAs in a way that each DSA available for execution has a representation in the worklist. Starting a workflow leads to the appearance of a new entry, represented by the starting DSA, in the task area. If the worklist for all actors is empty, the workflow as a whole has finished. In addition, a dialogue processor and a worklist processor are assumed as parts of a hypothetical workflow system which executes the workflow chart specification. The details of this system can be looked up in [Gei11].

CPs and ISAs are both typed. Both node types represent the basic user interaction of a workflow. CPs specify through their type and their reporting action which data is displayed to the actor, whereas ISAs specify through their type the data an actor can possibly enter to modify the system state on submission. CPs and ISAs are associated via so called enabling conditions. If an enabling condition evaluates to *true*, the respective data input possibility according to the associated ISA is presented to the actor. The representation and evaluating of conditions is managed by the aforementioned dialogue processor. It is a characteristic of workflow charts that the CPs and the ISAs are modelled as nodes but that they are not visited separately during execution of the workflow. Instead one CP and its corresponding ISAs are handled as a kind of a meta-node.

If the actor enters and submits data, he can only submit data for one ISA, even in the presence of multiple ISAs to choose from. The workflow system evaluates the different activation/show conditions (we call them activation conditions from now on) which are associated to the activated ISA. One ISA can have $0, \dots, n$ activation conditions, and they may all evaluate to *true*. For each activation condition there is an associated DSA. If m activation conditions of an ISA evaluate to *true*, m corresponding DSAs are activated, resp. inserted into the actor's worklist.

Active DSAs are represented as workflow items in the worklist which an actor can choose from. Typically the worklist processor shows these items as links to the actor and thus the type of a DSA is void. If the actor chooses a workflow item, the corresponding DSA activates exactly one CP from the available n CPs. This is determined by the flow conditions, which are specified with the DSA. Only one of them may evaluate to *true*. The determined CP in combination with the associated enabled ISAs is then presented to the actor.

Fig. 2 on the following page shows the different elements of workflow charts in one diagram. In addition to the already specified nodes, links, and conditions one can also see roles. Roles are used to specify which actor has access to which worklist item (DSA). The different conditions (enabling conditions, activation conditions, and flow conditions) are evaluated against the global system state, which is not specified in detail here.

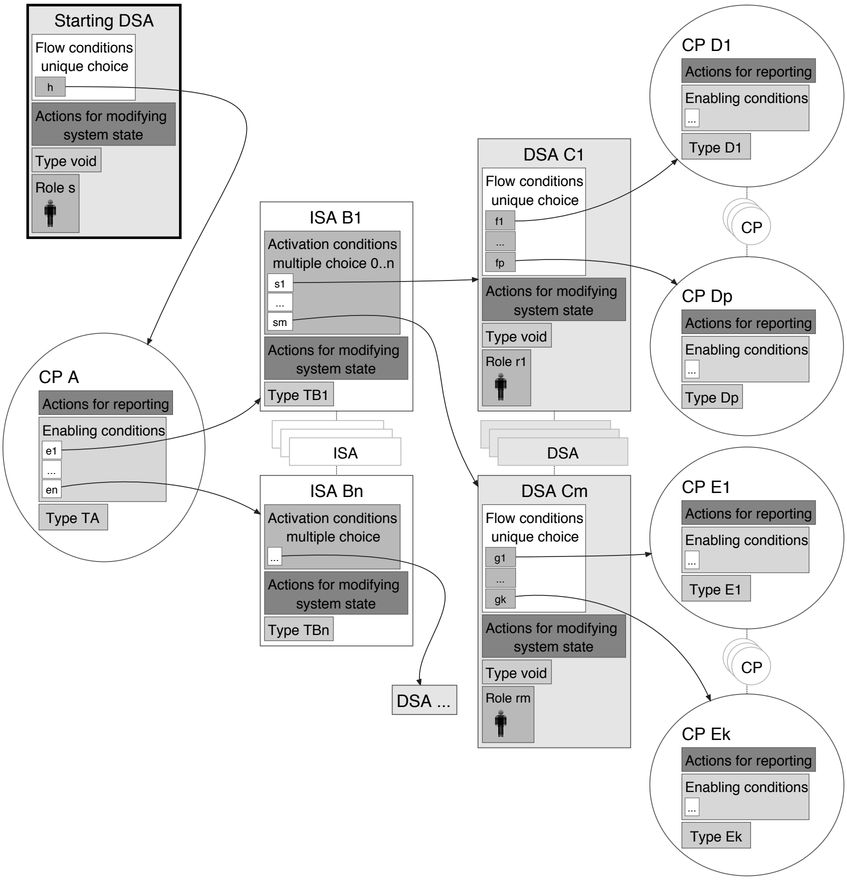


Figure 2: Complete workflow chart at a glance.

Example of Workflow Charts

To stress the understanding of workflow charts we present a small example, the beginning of a Business Trip Application Process (BTAP)[2] in Fig. 3 on the next page. The given example presents a business trip (trip from now on) workflow that deals with the tasks of trip application, review, approval and cancellation. The workflow is the scientific outcome of a comprehensive case study in association with the Austrian Social Insurance Company

[2]We are aware of the many different styles of BTAPs and that this model falls short in covering all of them.

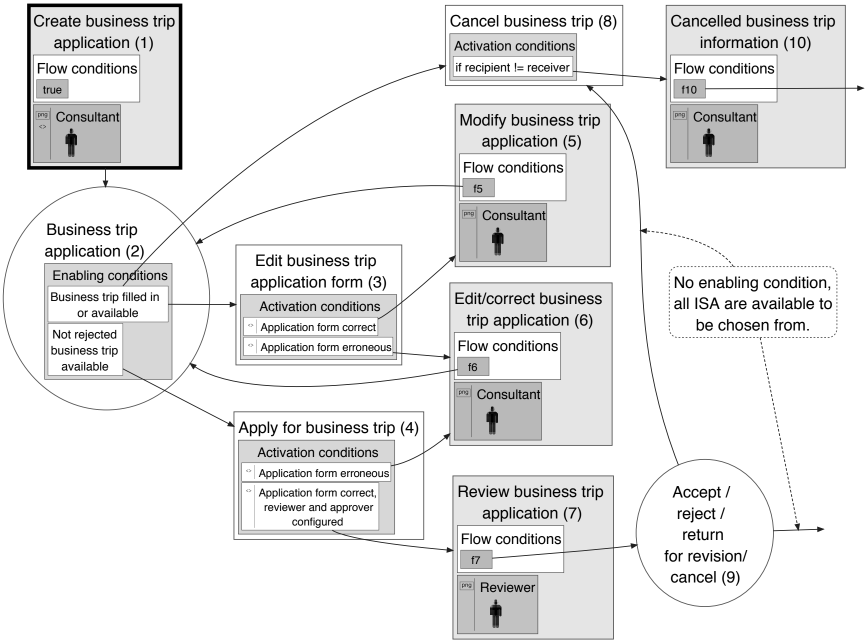


Figure 3: Beginning of a BTAP using workflow charts.

for Occupational Risks (AUVA) [DN08]. The different nodes of the workflow chart in the figure are numbered in parenthesis after the name of the node.

For the example we assume that the actor, in the figure named “consultant”, can select items from his worklist and that he uses a user interface, which allows in- and output of data and which is serviced by a system that is unspecified otherwise. In addition, there is a second actor in the diagram, “reviewer”, whom we will need at the end of the example.

Additionally, we assume that the consultant starts with no pre-existing trip applications. Then possible actions in this model are as follows:

- The consultant starts the workflow by selecting a representation of node (1) of Fig. 3 from his worklist.
- The system presents the information specified in node (2), a CP, to the consultant. In addition, the presentation contains an entry form for the data specified in node (3) and node (8), both an ISA, as these are the two ISAs whose enabling conditions evaluate to *true* at this point of time. The consultant can submit data of only one of the presented entry forms.
- When the consultant submits the data for ISA (3), the system evaluates the activation

conditions. If the system does not detect erroneous application data, a representation of DSA (5) is added to the worklist of the consultant. Otherwise, a representation of DSA (6) is added to the worklist of the consultant.

- Now the consultant sees the representation of either node (5) or node (6) in his worklist, depending on the previously taken path.[3]
- Let us assume that the consultant sees DSA (5) and activates it. The system presents to him the information from CP (2) and offers him three forms to submit to the system: A form to cancel the trip (8), a form to edit the data of the trip (3), and a form to apply for the trip (4). Now the consultant again can submit only one form, resp. one set of data, to the system. Depending on his choice the workflow continues accordingly. In case he applies for the trip (node (4)), an item (corresponding to node (7)) is created in the worklist of the reviewer.

3 Semantics of Workflow Charts using Abstract State Machines

3.1 Abstract State Machines

Some introductions to ASMs have already been written, so we only give a very brief introduction to this concept.

An ASM is a finite set of association rules of the form **if** *Condition* **then** *Updates* which transforms abstract states. The *Condition* - or *guard* - is an arbitrary predicate logic formula without free variables, which evaluates to *true* or *false*. *Updates* is a finite set of assignments of the form $f(a_1,\ldots,a_n) := v$ whose execution is interpreted as changing (or defining if undefined) in parallel the value of the occurring functions f at the indicated arguments $a_1,\ldots a_n$ to the indicated value v.

The notion of ASM *states* is the classical notion of mathematical structures where data comes as abstract objects, i.e., as elements of sets (domains, universes, one for each category of data) which are equipped with basic operations (partial functions) and predicates (attributes or relations). By default, it includes equality sign, the nullary operations *true*, *false*, and *undef* and the boolean operations.

The notion of ASM *run* is the classical notion of computation of transition systems. An ASM computation step in a given state consists in executing simultaneously all updates of all transition rules whose guard is true in the state, if these updates are consistent. For the evaluation of terms and formulae in an ASM state, the standard interpretation of function symbols by the corresponding functions in that state is used.

Functions are classified as *basic* or *derived* functions, where basic functions are part of the state, while derived function are are a kind auxiliary functions, which may vary over time but are not updatable directly by neither the ASM nor the environment.

[3] We modelled the cancel operation in a way that a consultant is presented with this option when selecting a trip for modification or for editing/correcting. We could as well model the cancel operation by making it available from the worklist.

This very short introduction is taken from the section “ASMs in a Nutshell” in [Bör04]. A full tract of ASMs can be found in [BS03]. We decided to apply ASMs because they can be seen as “a rather intuitive form of abstract pseudo-code”, though based on a precise but minimal mathematical theory of algorithms, but also as “Virtual Machine programs working on abstract data” [BS03]. This obviously brings the notation very close to that of programming languages and is thus easily understandable by programmers.

3.2 Workflow Charts and Abstract State Machines

Parts of the semantics of formcharts, the predecessor of workflow charts, are specified using UML diagrams [DWL06], so it seems natural to draw from the specification of UML in ASMs, for example from Börger et al. [BCR00], [BCR04], who specify UML 1.3 state machines as ASMs, or from Sarstedt and Guttmann [SG07], who specify UML 2 activity diagrams with ASMs. Additionally, this specification is somewhat inspired by the specification of BPMN with ASMs as presented in [BT08a], [BT08b].

Since ASMs specify an operational semantics, we specify a machine which operates on a given workflow chart specification, a WfCAgent[4]. In addition, we assume an environment which provides the events an actor can cause, namely those for selecting an element from the worklist and submitting data.

As workflow charts relay the responsibility for synchronising different execution paths of the workflow to the interplay of side effects and activation conditions [Dra10], which need to be specified by the designer of the workflow model, no general synchronisation semantics is currently given for workflow charts in [Dra10]. If one needs a specific synchronisation behaviour, the desired behaviour can be modelled as part of the guard conditions of the associations (see section 3.5 on the next page).

3.3 The Environment

The ASM we will specify later on will operate in an environment, which we will specify here. As the environment is not our main focus, we will specify elements of the environment only as detailed as necessary.

Actor. An actor is someone who activates tasks from the worklist and processes them. We will model actors as agents and define a set *Actors* which contains all defined actors.

Worklist. The worklist stores all tasks available for activation. Tasks in the worklist are associated with actors. We define the worklist as follows:

$worklist : WfCAgent \rightarrow List\ of\ DSA\_Nodes$

UI. We assume that there is a UI component available which displays information to the actor and receives input from him.

[4] This indicates that we will use ASM agents as ASM of choice. More on that later on.

3.4 Nodes

Workflow charts specify tripartite finite directed graphs whose nodes belong to the abstract set *Nodes*. The set *Nodes* is partitioned into a set for CP, ISA, and DSA nodes. The corresponding subsets of *Nodes* are *CP_Nodes*, *ISA_Nodes*, and *DSA_Nodes*.

CP_Nodes are of the form $node(type, reporting\_action)$, where the parameter *type* denotes the type of the node and *reporting_action* denotes the action used to provide the data to be reported.[5]

ISA_Nodes are of the form $node(type, action)$, where *type* denotes the type of the node and *action* denotes the function modifying the system state.

DSA_Nodes are of the form $node(action)$, where *action* has the same meaning as with ISA nodes.

We understand actions as used here as pieces of executable code which depend on the specific workflow instance. In a final system these actions thus could and would be executed in the system context while here they serve as generic place-holder which denote when the actions are to be executed.

3.5 Associations

Workflow charts have two types of associations although they are drawn identically in the diagrams: associations which are triggered by events[6], as those from DSAs to CPs or from ISAs to DSAs, and associations with a more structural aspect, as those from CPs to ISAs. Both types are modelled with one set named *Associations*. We could partition that set according to the types given above but for our purpose the unpartitioned set suffices.

Associations are of the form $assoc(source, targets, guards)$, where the parameter *source* represents the source of the association, *targets* represents a finite sequence of target nodes, and *guards* is a sequence of the same length as *targets* of Boolean expressions which guard the association from *source* to *targets*. Guards may evaluate information available in the environment as well as the available input data.

3.6 Additional Functions for Nodes an Associations

For the parameters of each type of node and association (e.g., *action* for *Nodes*, *guard* for *Associations*), we use a static function *param* which applied to the related elements yields the corresponding parameter. For example, the static function $action(node)$ yields the

[5]This action is not mandatory but it seems appropriate that there is an action preparing data to be reported. This action does not change the system state.

[6]These events are usually caused by the actor by selecting an item from the worklist.

action associated to the given node, $guard(association)$ yields the guard of an association, etc.

In addition, we define a function *ActorOfDSA* which is defined as follows:

$ActorOfDSA : DSA\_Nodes \rightarrow Actors$

The task of the *ActorOfDSA* is to store which DSA is associated with which actor. This function could be seen as part of a rudimental authorisation system.

3.7 The WfCAgent ASM

The WFCAGENT ASM represents an actor who chooses tasks from the worklist. The WFCAGENT is implemented as an ASM agent, thus a special function *self*, denoting the currently active agent, is available. Multiple agents can process multiple items of the worklist.

The main rule of a WFCAGENT has basically two states for "selection" of a worklist item and "dataInput" for processing the actor-supplied data. Associated with these states are the following events:

$Event = \{TaskFromWorklistSelected, DataSubmitted\}$

The ASM uses an *event*-function as specified in [BG94]:

$event : Daemon \rightarrow Event$

The *Daemon* responsible for emitting the events is out of scope of this paper.

In addition, we define two *in*-functions, *SelectedNode* and *DataInput*. The *Daemon* will provide by means of the *SelectedNode* function the DSA or ISA node resp. which node corresponds to the selection from the worklist or which has been activated by data input. In case of the event *DataSubmitted*, *DataInput* will contain the data associated with the event. The function is defined as:

$DataInput : Event \rightarrow DATA*$

Finally, we define the control states of the agent:

$CtlState : WfCAgent \rightarrow \{waitingForSelection, waitingForDataInput\}$

The agent will start in the $CtlState = waitingForSelection$.

Now we can define the main rule of the WFCAGENT:

WFCAGENT =
  if $event = TaskFromWorklistSelected$
    $\wedge\ (CtlState(self) = waitingForSelection)$ **then**
      PROCESSCLIENTPAGE$(SelectedNode)$
      $CtlState(self) := waitingForDataInput$

if $event = DataSubmitted$

$\wedge\ (CtlState(self) = waitingForDataInput)$ **then**

$\text{PROCESSDATAINPUT}(SelectedNode)$

$CtlState(self) := waitingForSelection$

To process the selected node from the worklist, we need to know which CP node is implicitly selected with the DSA node. Thus, we define a function *EnabledNodes* which yields the enabled CP as follows:

$$EnabledNodes(source, sourceNodeSet, targetNodeSet) =$$
$$\{target \in targetNodeSet \mid$$
$$source \in sourceNodeSet \wedge$$
$$\exists_{a \in Associations} source(a) = source \wedge target(a) = target \wedge guard(a)\}$$

To compute the CP with this function we call it in the following way:

$EnabledNodes$(currently active DSA, set of all DSAs, set of all CPs)

Processing of the resulting CP node is simply done by evaluating all enabling conditions for the corresponding ISA nodes and presenting a UI to the actor which contains the input fields corresponding to the types of the enabled ISA nodes. PRESENTUI is an additional ASM not specified here which is responsible for supplying the UI for the actor.

What we need to know is which ISAs are enabled by their conditions. We can use the function *EnabledNodes* for this as well by calling it with these parameters:

$EnabledNodes$(currently active CP, set of all CPs, set of all ISAs)

Now we can specify the macro PROCESSCLIENTPAGE:

PROCESSCLIENTPAGE(DSA) =

seq

$action(Dsa)$

$cp := EnabledNodes(Dsa, Dsa\_Nodes, CP\_Nodes)$

$\text{PRESENTUI}(cp, EnabledNodes(cp, CP\_Nodes, ISA\_Nodes))$

endseq

In order to wait for any action to be finished before a UI is presented, the statements of this macro are executed in sequential order.

In the second state of the WFCAGENT we need to process the data the environment resp. the actor has provided. The environment provides us with the selected node – which corresponds to an ISA node – and with the submitted data in *DataInput*.

We need to know which activation conditions are enabled, and again we can utilize the function *EnabledNodes* for this as well by calling it with these parameters:

$$EnabledNodes(\text{currently active ISA, set of all ISAs, set of all DSAs})$$

The macro *ProcessDataInput* can then be written as follows:

PROCESSDATAINPUT(ISA) =
  seq
    action(*Isa*)
    forall $dsa \in EnabledNodes(Isa, ISA\_Nodes, DSA\_Nodes)$ **do**
      add dsa to worklist(*ActorOfDSA*(*dsa*))
      where "add *x* to *y*" adds the DSA as possible task to the worklist
        of the corresponding agent
    endseq

We assume that processing of the via the function *DataInput* provided input happens in *action*(*isa*).

Note that a sequence for the action and the insertion of the next items into the worklist is specified this way. Thus, the action is completed before new items are added to the worklist. The insertion of worklist items can happen in parallel again.

4 Implementation

We implemented the specified semantics using CoreASM [SoCS11]. Due to space limitations we only give a short overview over the implementation, which is divided into two major parts: One part simulating actors and actor input – the ACTOREMULATORAGENT – and the other part being the workflow processor – the WFCAGENT. Both parts are modelled as agents, which are already supported by CoreASM via a plug-in.

The ACTOREMULATORAGENT basically waits for elements in the worklist. Then it randomly selects an element from the worklist of the active actor and submits this information to the WFCAGENT. The WFCAGENT computes the CP and all active ISAs. This information is submitted to the ACTOREMULATORAGENT to choose an ISA from. The ACTOREMULATORAGENT randomly chooses an ISA (this is the part where information would be submitted to the workflow system in a real implementation) and tells the WFCAGENT the chosen ISA. The WFCAGENT uses this choice to compute the new elements which should be added to the worklist.

5 Related Work

Formcharts as state history diagrams have been introduced in [DWL06]. State history diagrams are state transition and class diagrams at the same time. Although formcharts are the precursor to workflow charts, no precise formal semantics has been given in [Dra10]. Furthermore, formcharts show limitations regarding workflow execution as they only support single-user scenarios of two-staged human computer interaction (HCI) without concurrency [BDLW05]. Therefore, workflow charts extend formcharts with workflow semantics to develop an integrated specification for workflow definition and dialogue programming.

Another way to specify workflow charts would be Petri nets [Mur89], as they are used, for example, to specify the formal semantics of BPMN [DDO07]. There the proposed mapping from BPMN to Petri nets lacks features which coincide with the limitation of Petri nets that in turn motivated the design of YAWL [vdAtH05]. YAWL is a workflow definition language that extends Petri nets with a number of high-level features. In addition, modelling with Petri nets can become very complex very fast, as can be seen in [Tak08], where a relatively simple transaction in a travel agency's business process leads to a very complex Petri net representation. Another way to describe the semantics would have been using Communicating Sequential Processes (CSP). This has been done for BPMN in [WG08].

Finally, ASMs have been successfully used to specify the semantics of BPMN in [BT08a], [BT08b], and [BS11]. The approaches present extensive specifications trying to cover all aspects of the BPMN standard. This work on formalising workflow charts aims at specifying a plain workflow definition language, which includes exact semantics that allows generating executable code from formal descriptions of process-oriented enterprise applications.

6 Conclusion and Future Work

The semantics of workflow charts using Abstract State Machines has been introduced. The work presents a typed approach to workflow specification to improve flexibility in business process technology by proposing a concept for integrating workflow definition and dialogue programming that is also open to business process modelling. Workflow charts can be used as a technology-independent, conceptual modelling language for planning and documenting submit/response-style systems [ADG10]. Hence, workflow charts represent a platform independent model in the context of the model driven architecture community and add value by grasping the essential structure of a workflow system. By elaborating a programming language for specifying dialogue constraints, side effects and the type system, workflow charts can be exploited as a domain-specific language, i.e., a high-level programming language that overcomes the artificial separation of workflows and dialogues in business process management suites. Workflow charts extend formcharts [DW04] that represent a sufficient basis for the executable specification and the resulting description using Abstract State Machines provides a precise operational semantics for specifying human-computer interaction.

The proposed concept includes the support of actors and role models as well as the introduction of the worklist paradigm in order to present actors with their currently enabled tasks. The formalisation contributes to an exact semantics for reasoning about workflow charts and simulating models without the need to write a conventional prototype. Furthermore, having such a precise formal specification avoids common problems such as problems with OR-joins or further unclarities in the semantics of business process modelling languages. It is also possible to specify sub-workflows in workflow charts. As the semantics of sub-workflows is currently not clearly specified in the original work, a specification of the semantics regarding this aspect will be addressed later on. Additionally, the semantics of synchronisation could be specified explicitly.

Acknowledgement. The publication has been written within the project "Vertical Model Integration (VMI)" which is supported within the program "Regionale Wettbewerbsfähigkeit OÖ 2007-2013" by the European Fund for Regional Development as well as the State of Upper Austria. This work has also been supported in part by the Austrian Science Fund (FWF) under grant no. TRP 223-N23.

References

[ADG10] C. Atkinson, D. Draheim, and V. Geist. Typed Business Process Specification. In *Proc. of the 14th IEEE int. Enterprise Distributed Object Computing Conf. (EDOC 2010)*, pages 69 – 78. IEEE, 2010.

[BCR00] E. Börger, A. Cavarra, and E. Riccobene. Modeling the Dynamics of UML State Machines. In *Abstract State Machines - Theory and Applications*, volume 1912 of *Lecture Notes in Computer Science*, pages 167–186. Springer, 2000.

[BCR04] E. Börger, A. Cavarra, and E. Riccobene. On formalizing UML state machines using ASMs. *Information and Software Technology*, 46(5):287–292, April 2004.

[BDLW05] S. Balbo, D. Draheim, C. Lutteroth, and G. Weber. Appropriateness of User Interfaces to Tasks. In *Proc. of TAMODIA 2005 – 4th int. Workshop on Task Models and Diagrams for User Interface Design - For Work and Beyond*, pages 111–118. ACM Press, 2005.

[BG94] E. Börger and U. Glässer. A Formal Specification of the PVM Architecture. In *IFIP Congress (1)*, pages 402–409, 1994.

[Bör04] E. Börger. The ASM Ground Model Method as a Foundation of Requirements Engineering. In N. Dershowitz, editor, *Verification: Theory and Practice*, volume 2772 of *Lecture Notes in Computer Science*, pages 274–275. Springer, 2004.

[BS03] E. Börger and R. Stärk. *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer, 1 edition, June 2003.

[BS11] E. Börger and O. Sörensen. BPMN Core Modeling Concepts : Inheritance-Based Execution Semantics. In D. W Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice and Research Challenges*. Springer, 2011.

[BT08a] E. Börger and B. Thalheim. A Method for Verifiable and Validatable Business Process Modeling. In *Advances in Software Engineering: Lipari Summer School 2007, Lipari Island, Italy, July 8-21, 2007, Revised Tutorial Lectures*, pages 59–115. Springer, 2008.

[BT08b] E. Börger and B. Thalheim. Modeling Workflows, Interaction Patterns, Web Services and Business Processes: The ASM-Based Approach. In *Proc. of the 1st Int. Conf. on Abstract State Machines, B and Z*, pages 24–38, London, UK, 2008. Springer.

[DDO07] R. M Dijkman, M. Dumas, and C. Ouyang. Formal Semantics and Analysis of BPMN Process Models using Petri Nets. Technical Report 7115, Queensland University of Technology, Brisbane, Australia, 2007.

[DN08] D. Draheim and C. Natschläger. A context-oriented synchronization approach. In *Electronic Proc. of the 2nd int. Workshop in Personalized Access, Profile Management, and Context Awarness: Databases (PersDB 2008) in conjunction with the 34th VLDB conf.*, pages 20–27, 2008.

[Dra10] D. Draheim. *Business Process Technology: A Unified View on Business Processes, Workflows and Enterprise Applications*. Springer, 1 edition, June 2010.

[DW04] D. Draheim and G. Weber. *Form-Oriented Analysis. A New Methodology to Model Form-Based Applications*. Springer, Berlin, 1 edition, October 2004.

[DWL06] D. Draheim, G. Weber, and C. Lutteroth. Finite State History Modeling and Its Precise UML-Based Semantics. In *Advances in Conceptual Modeling - Theory and Practice*, volume 4231 of *Lecture Notes in Computer Science*, pages 43–52. Springer, 2006.

[FSKdR05] H. Fecher, J. Schönborn, M. Kyas, and W.-P. de Roever. 29 new unclarities in the semantics of UML 2.0 state machines. In *Proc. of the 7th Int. Conf. on Formal Methods and Software Engineering*, ICFEM05, pages 52–65, Berlin, Heidelberg, 2005. Springer.

[FSS+11] A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier, and E. Börger. *Subjektorientiertes Prozessmanagement: Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern*. Carl Hanser Verlag GmbH & CO. KG, July 2011.

[Gei11] V. Geist. *Integrated Executable Business Process and Dialogue Specification*. PhD thesis, Johannes Kepler University Linz, 2011.

[Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, 77(4):541–580, 1989.

[Nat11] Christine Natschläger. Towards a BPMN 2.0 Ontology. In Remco Dijkman, Jörg Hofstetter, and Jana Koehler, editors, *Business Process Model and Notation*, volume 95, pages 1–15. Springer, Berlin, Heidelberg, 2011.

[OMG11] OMG. BPMN 2.0 specification, final, January 2011.

[SG07] S. Sarstedt and W. Guttmann. An ASM semantics of token flow in UML 2 activity diagrams. In *Proc. of the 6th int. Andrei Ershov memorial Conf. on Perspectives of systems informatics*, pages 349–362, Novosibirsk, Russia, 2007. Springer.

[SoCS11] Canada School of Computing Science, Simon Fraser University. The CoreASM Project, 2011. http://www.coreasm.org/.

[Tak08] T. Takemura. Formal Semantics and Verification of BPMN Transaction and Compensation. In *2008 IEEE Asia-Pacific Services Computing Conf.*, pages 284–290, Yilan, Taiwan, December 2008.

[vdAtH05] Wil M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, June 2005.

[WG08] P. Y. Wong and J. Gibbons. A Process Semantics for BPMN. In *Proc. of the 10th Int. Conf. on Formal Methods and Software Engineering*, ICFEM '08, pages 355–374. Springer, 2008.

A UML Extension for Modeling Break-Glass Policies

Sigrid Schefer-Wenzl, Mark Strembeck

Institute for Information Systems, New Media Lab
Vienna University of Economics and Business (WU Vienna)
Augasse 2-6, 1090 Vienna, Austria
{firstname.lastname}@wu.ac.at

Abstract: In emergency situations, certain subjects sometimes have to perform important tasks although they are usually not authorized to perform these tasks. Break-glass policies have been introduced as a sophisticated exception handling mechanism to resolve such situations. They enable selected subjects to break or override the standard access control policies of an information system in a controlled manner. However, modeling support for break-glass policies is largely missing. In this paper, we present an approach to provide modeling support for break-glass policies in the context of process-related RBAC models. In particular, we provide a UML2 extension that allows for the integrated modeling of processes and break-glass policies. Additional constraints are formally specified via OCL. We also implemented our approach as an extension to the BusinessActivity library and runtime engine. The source code of our implementation is available for download.

1 Introduction

Process modeling languages provide primitives and construction rules to model the sequence of tasks performed in a business process. Corresponding access control models specify which subjects are authorized to perform the tasks that are included in the business processes (see, e.g., [Str10, SM11, WBK03]). However, most process diagrams and corresponding access control models only visualize standard task sequences and do not consider exceptional situations, such as emergency scenarios, where no authorized subject is available to execute a particular task (see, e.g., [RvdAH06, WRR07]). This results from the fact that proper modeling support for emergency scenarios is largely missing [vdARD07].

In many organizational environments some critical tasks exist which – in exceptional cases – must be performed by a subject although he/she is usually not authorized to perform these tasks. For example, a junior physician shall be able to perform certain tasks of a senior physician in case of emergency. *Break-glass policies* can be used to flexibly handle this kind of exceptional situations by breaking or overriding the standard access controls in a controlled manner (see, e.g., [BP09, BPW10, FCCA+06, MCMD11]). The term "break-glass" is a metaphor referring to the act of breaking the glass to pull a fire alarm. Accordingly, process-related break glass policies define override rules for subjects to allow the execution of certain tasks in exceptional cases. Applying a break-glass policy implies that the resulting task executions need to be carefully recorded for later audit and review. Typically, a special review process is triggered to monitor such break-glass executions.

In order to model process-related break-glass policies, we need an approach that integrates the break-glass concept into a modeling language. However, standard process modeling languages, such as BPMN [OMG11a] or UML Activity diagrams [OMG11b], do not provide native language constructs to model break-glass policies. In current practice, the lack of native modeling support for exception handling mechanisms can result in very complex diagrams depicting all possible exceptional execution paths and scenarios (see, e.g., [vdARD07]). In this paper, we therefore present a break-glass extension for UML that can help to specify graphical break-glass models. We define a domain-specific extension for the Unified Modeling Language (UML) [OMG11b] for modeling process-related break-glass models. In particular, we integrate our break-glass models into the BusinessActivity extension that supports the definition of process-related RBAC models [SM11]. Moreover, we also implemented the corresponding break-glass concepts as an extension to the BusinessActivity library and runtime engine (see [SM10, SM11]). The source code of our implementation is available for download[1].

The remainder of this paper is structured as follows. In Section 2, we give an overview of the *BusinessActivities* extension and motivate the need for integrating break-glass policies into business process models. Section 3 introduces our extension for modeling break-glass models via extended UML2 Activity diagrams. Moreover, we formally define the semantics of our newly introduced modeling elements via OCL constraints. Next, Section 4 presents an example business process model including break-glass policies. Section 5 discusses our approach in comparison to related work and Section 6 concludes the paper.

2 Background and Motivation

2.1 BusinessActivities: Modeling Support for Process-Related RBAC Models

In recent years, role-based access control (RBAC) [FKC07, SCFY96] has developed into a de facto standard for access control in both, research and industry. In RBAC, roles correspond to different job-positions and scopes of duty within a particular organization or information system [Str10]. Access permissions are assigned to roles according to the tasks this role has to accomplish, and subjects (e.g., human users) are assigned to roles. Thereby, each subject acquires all permissions that are necessary to fulfill its duties. Several extensions for RBAC exist for different application domains.

The *BusinessActivities* extension was designed for the integrated modeling of business processes and access control concepts by providing modeling support for process-related RBAC models [SM11]. Figure 1 shows an example of a simple medical examination process modeled as a BusinessActivity. The process starts when a patient arrives at the hospital. Subsequently, the “Medical examination” task (t_1) is conducted to reach a med-

[1]`http://wi.wu.ac.at/home/mark/BusinessActivities/library.html`

ical diagnosis. Next, the "Determine treatment options" task (t_2) is executed to devise an appropriate treatment plan. This treatment plan has to be confirmed by a second physician (t_3). In case the treatment plan includes errors or is incomplete, it must be revised before it is resubmitted for confirmation. Finally, the "Medical treatment" task (t_4) is performed. Each of the tasks (e.g., medical examination) is typically associated with certain access permissions (e.g., to read the patient record). Therefore, *subjects* participating in this workflow must be authorized to perform the tasks needed to complete the process (see, e.g., [GMPT01, OP03]).

In Figure 1, members of the Junior Physician role are permitted to perform the tasks t_1, t_2, and t_4. Task t_3 ("Confirm treatment") can only be performed by subjects assigned to the Senior Physician role. Furthermore, the Junior Physician role is defined as junior-role of the Senior Physician role via a role-to-role assignment ("rrAssign"). In RBAC, a senior-role inherits all permissions of its junior-roles.

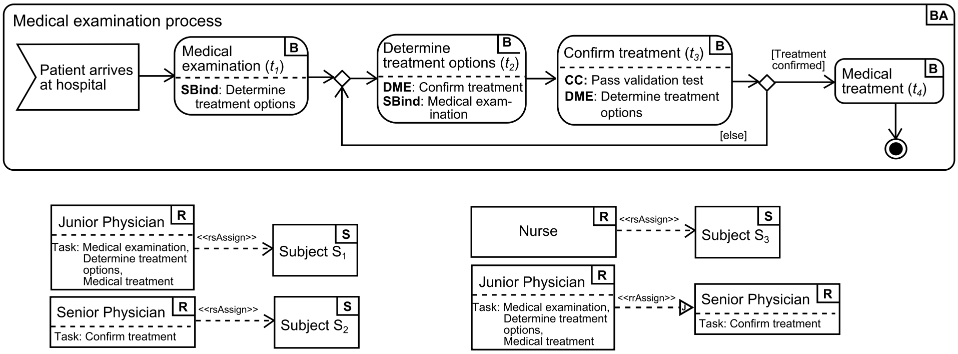


Figure 1: Simple medical examination process

In addition, the BusinessActivities extension supports the definition of different types of entailment constraints. A *task-based entailment constraint* places some restriction on the subjects who can perform a $task_x$ given that a certain subject has performed $task_y$. Thus, task-based entailment constraints have an impact on the combination of subjects and roles who are allowed (or required) to execute particular tasks (see, e.g., [RHE05, SSMB11, SM10, SM11, TCG04, WA06, WSM08]). Examples of entailment constraints include static mutual exclusion (SME), dynamic mutual exclusion (DME), subject-binding (SB), and role-binding (RB) constraints. A SME constraint defines that two statically mutual exclusive tasks must never be assigned to the same subject. In turn, DME tasks can be assigned to the same role, but within the same process instance they must be executed by different subjects. A SB constraint defines that two bound tasks must be performed by the same individual within the same process instance. A RB constraint defines that bound tasks must be performed by members of the same role, but not necessarily by the same individual.

In the example process shown in Figure 1, we define a subject-binding between the tasks t_1 and t_2 to ensure that the same physician who performed the examination in the "Medical examination" task also evaluates appropriate medical treatment options. This subject-binding is indicated via *SBind* entries in the corresponding task symbols (see Figure 1). In addition, we define a DME constraint on the tasks t_2 and t_3 to enforce the four-eyes-principle on medical examinations. Thus, for each medical examination the "Determine treatment options" and the "Confirm treatment" tasks must always be conducted by two different individuals. This is an essential quality and safety measure in hospitals to guard against mistakes and malpractice.

Moreover, in an IT-supported workflow, *context constraints* can be defined as a means to consider context information in access control decisions (see, e.g. [BBF01, SN04]). Typical examples for context constraints in organizational settings regard the temporal or spatial context of task execution, user-specific attributes, or the task execution history of a user (see, e.g., [CCB08]). In this paper, context constraints define that certain contextual attributes must meet certain predefined conditions to permit the execution of a specific task [SN04, SWS12]. In the example process, a context constraint (CC) can be defined on the "Confirm treatment" task which specifies several conditions that must be met in order to successfully validate the medical treatment plan (see Figure 1).

2.2 Motivation for Modeling Break-Glass Policies

Let us consider three potential *emergency scenarios* for the medical examination process shown in Figure 1:

(1) In case of emergency, no senior physician is available. However, only senior physicians are allowed to perform the "Confirm treatment" task (see Figure 1). To be able to start the "Medical treatment", a break-glass policy can authorize junior physicians to perform the "Confirm treatment" in case of emergency.

(2) Only one authorized subject is available to perform the tasks "Determine treatment options" and "Confirm treatment". However, due to a DME constraint these two tasks must be executed by different subjects. A break-glass policy can authorize the only available subject to override the DME constraint in case of emergency.

(3) If no physician is available to perform the "Medical treatment" task in case of emergency, subject s_3 – who is assigned to the nurse role – is allowed to execute this task. However, all other members of the nurse role do not have the necessary skills to perform this task. Therefore, a break-glass policy directly authorizes subject s_3 to perform the "Medical treatment" in a break-glass scenario.

Standard UML elements do not provide modeling support for process-related break-glass policies. For example, in Figure 1, these emergency scenarios would have to be included into the same process model. Apparently, this would result in a very complex diagram (see, e.g., [vdARD07]). Alternatively, each of the scenarios either needs to be modeled in a separate process model, or the break-glass information is included as a comment into the UML activity diagram. This simple example already shows that it is difficult to describe all connections and implications of process-related break-glass policies in a textual manner. For this reason, we define a UML extension for the integrated modeling of process-related break-glass policies.

3 A UML Extension for Process-Related Break-Glass Policies

The Unified Modeling Language (UML) [OMG11b] is a de facto standard for the specification of information systems. Modeling support for break-glass policies via a standard notation can help to bridge the communication gap between software engineers, security experts, experts of the application domain, and other stakeholders (see, e.g., [MJ10]). Our domain-specific modeling extension for break-glass policies serves as an enabler to document and communicate how certain emergency scenarios can be handled in a business process.

UML2 Activity models offer a process modeling language that allows to model the control and object flows between different actions. The main element of an Activity diagram is an Activity. Its behavior is defined by a decomposition into different Actions. An UML2 Activity thus models a process while the Actions included in the Activity are used to model tasks (for details on UML2 Activity models, see [OMG11b]). However, sometimes diagrams can not provide all relevant aspects of a specification. Therefore, there is a need to define additional constraints about the modeling elements. The Object Constraint Language (OCL) provides a formal language that enables the definition of constraints on UML models [OMG12]. We apply the OCL to define additional break-glass specific constraints for our UML extension. In particular, the OCL invariants defined in Section 3.2 ensure the consistency and correctness of UML models using our new modeling elements.

The UML standard basically provides two options to adapt its metamodel to a specific area of application [OMG11b]: a) defining a UML profile specification using stereotypes, tag definitions, and constraints. A UML profile must not change the UML metamodel but can only extend existing UML meta-classes for special domains. Thus, UML profiles are not a first-class extension mechanism (see [OMG11b, page 660]); b) extending the UML metamodel, which allows for the definition of new elements with customized semantics. In this paper, we apply the second option because the newly defined modeling elements for break-glass policies require new semantics which are not available in the UML metamodel. Thus, we introduce the *BreakGlassBusinessActivities* extension for the UML metamodel which is designed for modeling process-related break-glass policies (see Section 3.1). In particular, we extend the *BusinessActivities* package [SM11], which provides UML modeling support for process-related RBAC models. We also implemented the extended metamodel presented in Section 3.1 as well as the corresponding constraints provided in Section 3.2 as a break-glass extension to the BusinessActivity library and runtime engine[1] (see [SM10, SM11]).

3.1 Metamodel overview

A *BusinessActivity* [SM11] is a specialized UML Activity (see Figure 2). A *BusinessAction* corresponds to a task and comprises all permissions to perform the task. *Roles* and *Subjects* are linked to BusinessActions. For a detailed discussion on how mutual exclu-

[1]Available at `http://wi.wu.ac.at/home/mark/BusinessActivities/library.html`

sion, binding, and context constraints are integrated into the BusinessActivities extension, see [SM11, SWS12].

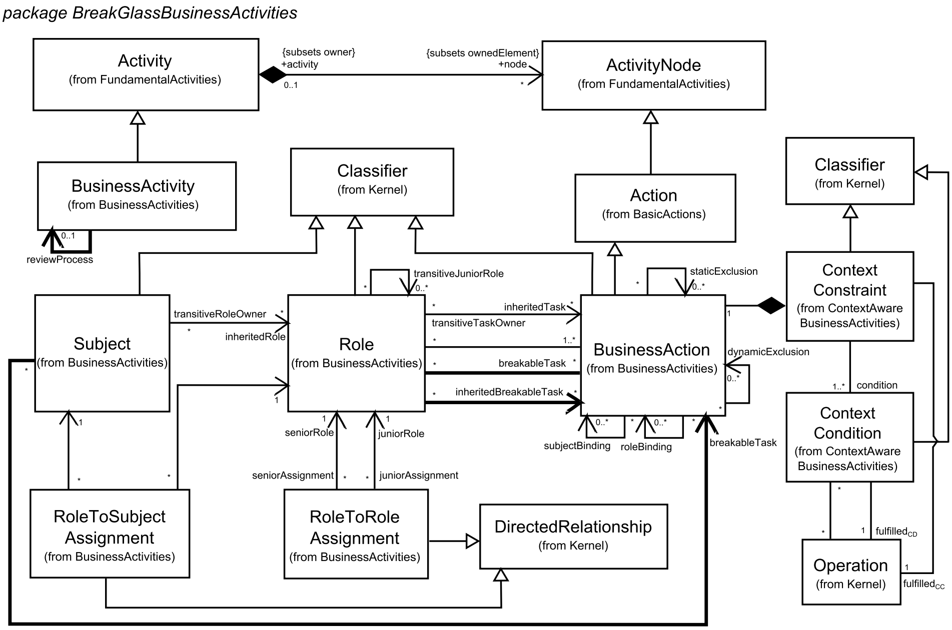


Figure 2: UML metamodel extension for process-related break-glass RBAC models

To support the definition of break-glass policies in business process models, we specify that certain *breakable tasks* can be performed by subjects who are usually not allowed to execute these tasks. For this purpose, override rules regulate that members of a certain role are permitted to perform a certain task in case of emergency (*breakable-by-role* override). In addition to role-based break-glass rules, our approach enables the definition of subject-specific break-glass rules, i.e. only a certain subject is authorized to execute a task in case of emergency (*breakable-by-subject* override). Breakable-by-subject override rules are used in cases where only certain individuals have all necessary competencies to perform the breakable task. Each break-glass execution will be recorded and subsequently be monitored via a corresponding review process.

For integrating break-glass policies into the UML metamodel, we introduce the following new relations: Each Role can include *breakableTasks* and *inheritedBreakableTasks*, which are inherited from its junior-roles. These two relationships can be used to visualize that members of a certain role are authorized to perform the assigned tasks only in case of emergency. Similarly, each Subject can be related to *breakableTasks* to show that a particular subject is permitted to perform these tasks in case of emergency. Figure 3 illustrates presentation options to visualize the breakable-by-role and breakable-by-subject override relations via "Breakable" entries. Note that these relations are formally defined through our UML metamodel extension and therefore exist independent of their actual graphical representation. Moreover, each BusinessActivity is related to a *reviewProcess* (see below).

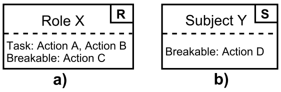


Figure 3: Visualizing (a) breakable-by-role and (b) breakable-by-subject override relations

Each instance of a BusinessAction and the corresponding BusinessActivity instance are marked as *broken* if the BusinessAction has been executed by a subject via a break-glass override assignment (see OCL constraints 1 and 2 in Section 3.2). For each broken BusinessActivity, there has to exist a corresponding reviewProcess (see Figure 2 and OCL constraint 3). A particular reviewProcess can be assigned to an arbitrary number (one or more) of BusinessActivity processes in an organization. Roles and subjects can own a task either regularly or via a break-glass override assignment (see OCL constraints 4 and 5). Moreover, in a break-glass scenario, the corresponding entailment constraints do not have to be fulfilled (see Constraints 1 and 2 as well as OCL constraints 7, 8, 9, 10).

3.2 OCL constraints

A structural UML model cannot capture certain types of domain-specific constraints which are relevant for describing a target domain. Thus, additional constraints can be defined, for example, by using a constraint expression language, such as the OCL [OMG12]. Below, we use OCL invariants to define the semantics by encoding break-glass specific constraints.

OCL Constraint 1 *Each BusinessAction defines an attribute called "broken" stating if a certain BusinessAction instance is executed via a break-glass override assignment:*

```
context BusinessAction inv:
self.instanceSpecification->forAll(i |
   i.slot->exists(b |
      b.definingFeature.name = broken ))
```

OCL Constraint 2 *Each BusinessActivity defines an attribute called "broken" stating if a certain BusinessActivity instance includes at least one broken BusinessAction:*

```
context BusinessActivity inv:
self.instanceSpecification->forAll(i |
   i.slot->exists(b |
      b.definingFeature.name = broken ))
```

OCL Constraint 3 *For each broken BusinessActivity instance, there has to exist a corresponding reviewProcess:*

```
context BusinessActivity inv:
self.instanceSpecification->forAll(i |
   if i.slot->exists(b |
      b.definingFeature.name = broken and b.value = true)
   then self.reviewProcess->notEmpty()
   else true endif)
```

OCL Constraint 4 *Each role is allowed to own a task either regularly or via a break-glass override assignment. To separate regular task ownerships from break-glass task ownerships, we need to ensure that no BusinessAction is assigned to a certain role via both mappings:*

```
context Role
inv: self.businessAction->forAll(b |
       self.breakableTask->select(bbr |
         bbr.name = b.name)->isEmpty())
inv: self.businessAction->forAll(b |
       self.inheritedBreakableTask->select(bbri |
         bbri.name = b.name)->isEmpty())
inv: self.inheritedTask->forAll(bi |
       self.breakableTask->select(bbr |
         bbr.name = bi.name)->isEmpty())
inv: self.inheritedTask->forAll(bi |
       self.inheritedBreakableTask->select(bbri |
         bbri.name = bi.name)->isEmpty())
```

OCL Constraint 5 *Each subject is allowed to own a task either regularly (via its role memberships) or via a break-glass override assignment. To separate regular task ownerships from breakable task ownerships, we need to ensure that no BusinessAction is assigned to a certain subject via both mappings:*

```
context Subject
inv: self.roleToSubjectAssignment->forAll(rsa |
       rsa.role.businessAction->forAll(b |
         self.breakableTask->select(bbs |
           bbs.name = b.name)->isEmpty()))
inv: self.roleToSubjectAssignment->forAll(rsa |
       rsa.role.inheritedTask->forAll(bi |
         self.breakableTask->select(bbs |
         bbs.name = bi.name)->isEmpty()))
inv: self.inheritedRole->forAll(ri |
       ri.businessAction->forAll(b |
         self.breakableTask->select(bbs |
           bbs.name = b.name)->isEmpty()))
inv: self.inheritedRole->forAll(ri |
       ri.inheritedTask->forAll(bi |
         self.breakableTask->select(bbs |
           bbs.name = bi.name)->isEmpty()))
```

OCL Constraint 6 *Each role inherits the breakable tasks assigned to its junior-roles (i.e. breakable tasks assigned to junior-roles are indirectly/transitively assigned to the corresponding senior-roles):*

```
context Role
inv: self.seniorAssignment->forAll(sa |
       sa.juniorRole.breakableTask->forAll(bbr |
         self. inheritedBreakableTask->exists(ibbr | ibbr.name = bbr.name)) and
       sa.juniorRole.inheritedBreakableTask->forAll(jb |
         self.inheritedBreakableTask.exists->(ibbr | ibbr.name = jb.name)))
inv: self.inheritedBreakableTask->forAll(ibbr |
       self.seniorAssignment->exists(sa |
         sa.juniorRole.breakableTask->exists(bbr | bbr.name = ibbr.name) or
         sa.juniorRole.inheritedBreakableTask->exists(jb | jb.name = ibbr.name)))
```

OCL Constraint 7 *For all broken BusinessAction instances, the executing subjects of corresponding SME tasks do not have to be different:*

```
context BusinessAction inv:
self.instanceSpecification->forAll(b |
  b.slot->select(s |
    s.definingFeature.name=broken
    if (s.value = true) then
      self.staticExclusion->forAll(sme |
        sme.instanceSpecification->forAll(i |
          b.slot->forAll(bs |
            i.slot->forAll(is |
              if bs.definingFeature.name=executingSubject
                and is.definingFeature.name=executingSubject
                then (bs.value = is.value) or not (bs.value = is.value)
              else true endif ))))
     else true endif ))
```

OCL Constraint 8 *For all broken BusinessAction instances, the executing subjects of DME tasks do not have to be different:*

```
context BusinessAction inv:
self.instanceSpecification->forAll(b |
  b.slot->select(s |
    s.definingFeature.name=broken
    if (s.value = true) then
      self.dynamicExclusion->forAll(dme |
        dme.instanceSpecification->forAll(i |
          b.slot->forAll(bs |
            i.slot->forAll(is |
              if bs.definingFeature.name=executingSubject
                and is.definingFeature.name=executingSubject
                then (bs.value = is.value) or not (bs.value = is.value)
              else true endif ))))
     else true endif ))
```

OCL Constraint 9 *For all broken BusinessAction instances, the executing role of role-bound tasks does not have to be the same:*

```
context BusinessAction inv:
self.instanceSpecification->forAll(b |
  b.slot->select(s |
    s.definingFeature.name=broken
    if (s.value = true) then
      self.roleBinding->forAll(rbt |
        rbt.instanceSpecification->forAll(i |
          b.slot->forAll(bs |
            i.slot->forAll(is |
              if bs.definingFeature.name=executingSubject
                and is.definingFeature.name=executingSubject
                then (bs.value = is.value) or not (bs.value = is.value)
              else true endif ))))
     else true endif ))
```

OCL Constraint 10 *For all broken BusinessAction instances, the executing subject of subject-bound tasks does not have to be the same:*

```
context BusinessAction inv:
self.instanceSpecification->forAll(b |
  b.slot->select(s |
    s.definingFeature.name=broken
    if (s.value = true) then
      self.subjectBinding->forAll(sbt |
        sbt.instanceSpecification->forAll(i |
          b.slot->forAll(bs |
            i.slot->forAll(is |
              if bs.definingFeature.name=executingSubject
                and is.definingFeature.name=executingSubject
                then (bs.value = is.value) or not (bs.value = is.value)
              else true endif ))))
     else true endif ))
```

Moreover, the following two constraints must be satisfied which cannot be expressed in OCL (see [OMG11b]):

Constraint 1 *For all broken BusinessAction instances, context constraints do not have to be fulfilled. Therefore, the fulfilled$_{CC}$ Operations do not have to evaluate to true.*

Constraint 2 *For all broken BusinessAction instances, context conditions do not have to be fulfilled. Therefore, the fulfilled$_{CD}$ Operations do not have to evaluate to true.*

4 Perspectives for UML Break-Glass Models

We suggest to use three complementary perspectives to model process-related break-glass RBAC models. This is because capturing all aspects within the process model will presumably overload it. Figure 4a shows the *process perspective* of the medical examination process (see Section 2).

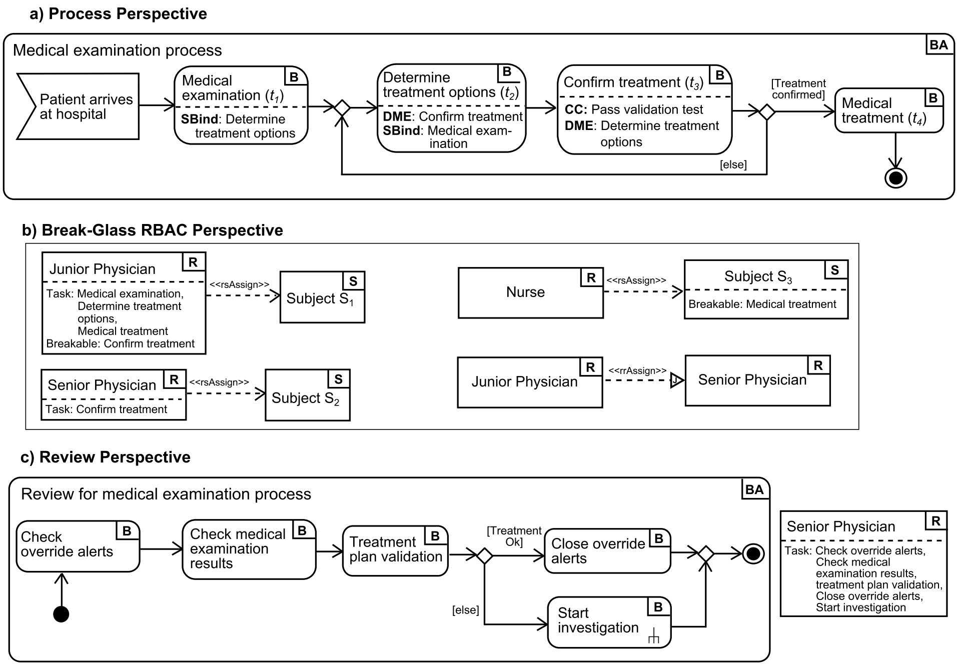


Figure 4: Example for process-related break-glass RBAC models

The *Break-Glass RBAC perspective* is exemplified in Figure 4b illustrating task-to-role, role-to-subject, and role-to-role assignments. For example, subject s_1 is assigned to the Junior Physician role. Corresponding notation symbols are described in detail in [SM11]. Moreover, this perspective provides a detailed view on which role or subject is allowed to perform a break-glass override. For example, we define a breakable-by-role override relation between the Junior Physician role and the “Confirm treatment” task in Figure 4b. Thus, in a break-glass scenario, members of the junior physician role are able to perform the “Confirm treatment” task. Moreover, a breakable-by-subject override is defined on subject s_3 and the “Medical treatment” task, because nurse s_3 has all necessary skills to perform the medical treatment in an emergency case.

Finally, the *review perspective* illustrates the review process which is triggered each time after a break-glass override is executed (see Section 3). An example review process for the medical examination process is shown in Figure 4c. In particular, a physician (who was not involved in the medical examination process) is appointed to perform the following tasks: After checking the override alerts for a particular process, the physician checks the medical examination results and validates the medical treatment plan. If the treatment plan is successfully validated, the override alerts are closed. Otherwise, an investigation process is started.

5 Discussion and Related Work

Standard process modeling languages, such as BPMN [OMG11a] or UML Activity diagrams [OMG11b], do not provide native language constructs to model break-glass policies. Previous work by van der Aalst et al. [vdARD07] has identified a general lack of process modeling language capabilities to adequately model emergency scenarios. Due to missing modeling support, organizations need to specify break-glass policies via informal textual descriptions, or the modeling of various break-glass scenarios results in complex and confusing diagrams. Such work-arounds, however, easily result in consistency, maintainability, and communication problems. The separation between the regular process model perspective and the break-glass view has been suggested in [vdARD07] in order to clearly distinguish between both scenarios. Our approach provides integrated modeling support for break-glass policies in a process-related RBAC context. Moreover, we propose three different modeling perspectives, where each of these perspectives focuses on different aspects of integrated break-glass models. By defining an extension to the UML2 standard and specifying OCL constraints for our newly introduced modeling elements, our extension can also be integrated with UML-based software tools.

In recent years, there has been much work on various aspects of (process-related) break-glass policies (see, e.g., [Pov00, FCCA+06, AdVF+10, MCMD11, FCF+09]). In [BP09], a break-glass extension for SecureUML is provided. The resulting SecureUML break-glass policies can then be transformed into XACML. However, this approach does not consider break-glass decisions in connection with dynamic mutual exclusion or binding constraints. In [WBK03], Wainer et al. present an RBAC model for workflow systems that allows the controlled overriding of entailment constraints in case of emergency. To achieve this, each constraint is associated with a certain level of priority. On the other hand, roles hold override privileges according to their level of responsibility. A comprehensive overview of exception handling patterns – including resource reallocation – is provided in [RvdAH06].

To the best of our knowledge, this work represents the first attempt to address break-glass policies from a process modeling perspective. However, several other approaches exist that deal with process adaptations and process evolutions in order to flexibly handle different types of exceptions in process-aware information systems. For example, [RD98] provides a formal model to support dynamic structural changes of process instances. A set of change operations is defined that can be applied by users in order to modify a process instance execution path. In [WRR07], change patterns and change support features are identified and several process management systems are evaluated regarding their ability to support process changes. Exception handling via structural adaptations of process models are also considered in [RRMD09]. In particular, several correctness criteria and their application to specific process meta models are discussed. In comparison to our work, all of these approaches have in common that processes must be changed in order to handle exceptional situations. The main goal of our approach is to maintain the designed process flow, while ensuring that only authorized subjects are allowed to participate in a workflow.

6 Conclusion

The need for integrated modeling of business processes and certain kinds of exception handling mechanisms has been repeatedly identified in research and practice. In this paper, we focus on the integrated modeling of business processes and break-glass policies to define who is allowed to execute particular critical tasks in case of emergency. This is necessary because standard process modeling languages do not provide native language support for break-glass policies. However, providing suitable modeling primitives for break-glass policies is especially important to support the controlled overriding of access rights in information systems.

In particular, we presented a domain-specific modeling extension for the UML2. We also used the Object Constraint Language to formally define break-glass specific knowledge which cannot be captured via the metamodel extension. Thus, our approach can be applied to supplement other UML-based approaches and can be integrated into UML-based software tools. Moreover, we implemented our approach as a break-glass extension for the BusinessActivity library and runtime engine.

References

[AdVF+10] C. A. Ardagna, S. D. C. di Vimercati, S. Foresti, T. W. Grandison, S. Jajodia, and P. Samarati. Access control for smarter healthcare using policy spaces. *Computers & Security*, 29(8), 2010.

[BBF01] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3), 2001.

[BP09] Achim D. Brucker and Helmut Petritsch. Extending Access Control Models with Break-Glass. In *Proceedings of the 14th ACM symposium on Access control models and technologies (SACMAT)*, 2009.

[BPW10] Achim D. Brucker, Helmut Petritsch, and Stefan G. Weber. Attribute-Based Encryption with Break-glass. In *Proc. of the Workshop In Information Security Theory And Practice (WISTP)*, 2010.

[CCB08] Frederic Cuppens and Nora Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security*, 7(4), July 2008.

[FCCA+06] A Ferreira, R Cruz-Correia, L Antunes, P Farinha, E Oliveira-Palhares, D W. Chadwick, and A Costa-Pereira. How to Break Access Control in a Controlled Manner. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, 2006.

[FCF+09] Ana Ferreira, David Chadwick, Pedro Farinha, Ricardo Correia, Gansen Zao, Rui Chilro, and Luis Antunes. How to Securely Break into RBAC: The BTG-RBAC Model. In *Proceedings of the 2009 Annual Computer Security Applications Conference*, December 2009.

[FKC07] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House, second edition edition, 2007.

[GMPT01] Christos K. Georgiadis, Ioannis Mavridis, George Pangalos, and Roshan K. Thomas. Flexible Team-Based Access Control Using Contexts. In *Proceedings of the sixth ACM symposium on Access control models and technologies (SACMAT)*, May 2001.

[MCMD11] Srdjan Marinovic, Robert Craven, Jiefei Ma, and Naranker Dulay. Rumpole: A Flexible Break-Glass Access Control Model. In *Proceedings of the 16th ACM symposium on Access control models and technologies (SACMAT)*, June 2011.

[MJ10] Haramlambos Mouratidis and Jan Jürjens. From Goal-Driven Security Requirements Engineering to Secure Design. *International Journal of Intelligent Systems*, 25(8), 2010.

[OMG11a] OMG. OMG Business Process Modeling Notation. available at: `http://www.omg.org/spec/BPMN/2.0/`, January 2011. Version 2.0, formal/2011-01-03, The Object Management Group.

[OMG11b] OMG. Unified Modeling Language (OMG UML): Superstructure. available at: `http://www.omg.org/spec/UML/2.4.1/`, August 2011. Version 2.4.1, formal/2011-08-05, The Object Management Group.

[OMG12] OMG. Object Constraint Language Specification. available at: `http://www.omg.org/spec/OCL/2.3.1/`, January 2012. Version 2.3.1, formal/2012-01-01, The Object Management Group.

[OP03] Sejong Oh and Seog Park. Task-Role-Based Access Control Model. *Information Systems*, 28(6), 2003.

[Pov00] Dean Povey. Optimistic Security: A New Access Control Paradigm. In *Proceedings of the 1999 workshop on New security paradigms (NSPW)*, 2000.

[RD98] Manfred Reichert and Peter Dadam. Adept_flex-Supporting Dynamic Changes of Workflows Without Losing Control. *J. Intell. Inf. Syst.*, 10(2), 1998.

[RHE05] Nick Russell, Arthur H. M. Ter Hofstede, and David Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE)*. Springer, 2005.

[RRMD09] Manfred Reichert, Stefanie Rinderle-Ma, and Peter Dadam. Flexibility in Process-Aware Information Systems. In Kurt Jensen and Wil M. Aalst, editors, *Transactions on Petri Nets and Other Models of Concurrency II*, pages 115–135. Springer, Berlin, Heidelberg, 2009.

[RvdAH06] Nick Russell, Wil M.P. van der Aalst, and Arthur H. M. Ter Hofstede. Exception Handling Patterns in Process-Aware Information Systems. In *International Conference on Advanced Information Systems Engineering (CAiSE)*, 2006.

[SCFY96] Ravi Sandhu, Edward Coyne, Hal Feinstein, and Charles Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2), 1996.

[SM10] Mark Strembeck and Jan Mendling. Generic Algorithms for Consistency Checking of Mutual-Exclusion and Binding Constraints in a Business Process Context. In *Proc. of the 18th International Conference on Cooperative Information Systems (CoopIS)*. Springer, October 2010.

[SM11] Mark Strembeck and Jan Mendling. Modeling Process-related RBAC Models with Extended UML Activity Models. *Information and Software Technology*, 53(5), 2011.

[SN04] Mark Strembeck and Gustaf Neumann. An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments. *ACM Transactions on Information and System Security (TISSEC)*, 7(3), August 2004.

[SSMB11] Sigrid Schefer, Mark Strembeck, Jan Mendling, and Anne Baumgrass. Detecting and Resolving Conflicts of Mutual-Exclusion and Binding Constraints in a Business Process Context. In *OTM Conferences (1) 2011, Proc. of the 19th International Conference on Cooperative Information Systems (CoopIS)*. Springer, October 2011.

[Str10] Mark Strembeck. Scenario-Driven Role Engineering. *IEEE Security & Privacy*, 8(1), 2010.

[SWS12] Sigrid Schefer-Wenzl and Mark Strembeck. Modeling Context-Aware RBAC Models for Business Processes in Ubiquitous Computing Environments. In *Proc. of the 3rd International Conference on Mobile, Ubiquitous and Intelligent Computing (MUSIC), IEEE*, June 2012.

[TCG04] Kaijun Tan, Jason Crampton, and Carl A. Gunter. The Consistency of Task-Based Authorization Constraints in Workflow Systems. In *Proceedings of the 17th IEEE workshop on Computer Security Foundations*, June 2004.

[vdARD07] Wil M. P. van der Aalst, Michael Rosemann, and Marlon Dumas. Deadline-based Escalation in Process-Aware Information Systems. *Decis. Support Syst.*, 43(2), 2007.

[WA06] Janice Warner and Vijayalakshmi Atluri. Inter-instance authorization constraints for secure workflow management. In *Proceedings of the eleventh ACM symposium on Access control models and technologies (SACMAT)*, June 2006.

[WBK03] Jacques Wainer, Paulo Barthelmess, and Akhil Kumar. W-RBAC - A Workflow Security Model Incorporating Controlled Overriding of Constraints. *International Journal of Cooperative Information Systems (IJCIS)*, 12(4), 2003.

[WRR07] Barbara Weber, Stefanie Rinderle, and Manfred Reichert. Change Patterns and Change Support Features in Process-Aware Information Systems. In *International Conference on Advanced Information Systems Engineering (CAiSE)*, 2007.

[WSM08] Christian Wolter, Andreas Schaad, and Christoph Meinel. Task-Based Entailment Constraints for Basic Workflow Patterns. In *Proceedings of the 13th ACM symposium on Access control models and technologies (SACMAT)*, 2008.

Selecting Event Monitoring Points for Optimal Prediction Quality

Andreas Rogge-Solti, Nico Herzberg, and Luise Pufahl

Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam
{firstname.lastname}@hpi.uni-potsdam.de

Abstract: Organizations strive to optimize their business processes in order to satisfy customer requirements and internal goals. A basic necessity in order to meet time and quality objectives is to monitor an organization's business processes. Process monitoring makes their execution more transparent and allows to react to observed deviations with corrective actions. This paper focuses on monitoring processes in manual or semi-automatic environments, where the installation of each monitoring point is costly, as it requires effort to measure and record observed progress. During process execution, the allocation of event monitoring points (EMPs) is restricted to certain positions, e.g., the termination of activities. We propose an approach for optimizing the allocation model of EMPs in order to improve the estimation quality. We implemented this approach and show its applicability in a case study of a Dutch hospital for its surgical care process.

1 Introduction

Modern companies and organizations face a challenging and ever changing market environment today. Thus, managing their business processes effectively and efficiently is essential to be competitive. A cornerstone of business process management is the monitoring of process instances. *Business process monitoring* assists in performance estimation, e.g., prediction of time until completion of a process instance or duration of certain activities [ASS11]. This technique enables the detection of performance issues, e.g., being behind schedule, so that corrective actions can be taken to finish the affected instances according to the plan and avoid deviations from planned goals and service level agreements.

Process execution ranges from completely manual, over semi-automated to fully automated enactment; the latter is a process execution according to specified models controlled by a central process engine. In automated environments, the central process engine provides execution data of the performed activities out-of-the-box usable for process monitoring. In contrast, non-automated process execution requires a separate implementation for capturing activity execution data. In several domains, the majority of process execution is still manual, such as in the healthcare domain, where the treatment processes require high flexibility and individual reactions to each patient [LR07].

In [HKRS12] an architecture is presented to make use of the sparse data that is generated while manually executing processes for monitoring purposes. In that work, the concept of event monitoring points (EMP) is introduced as well-defined places in the process model, to which process execution information can be correlated, e.g., the termination of activities. However, the information that is already available might not suffice to meet certain process monitoring requirements, e.g., prediction quality of time until completion of a process instance. In that case, additional installation of event capturing has to be considered to provide more transparency and improve prediction quality. Options for event capturing include simple stopwatches, bar-code scanners, RFID readers or other devices, but installing and running such equipment is expensive.

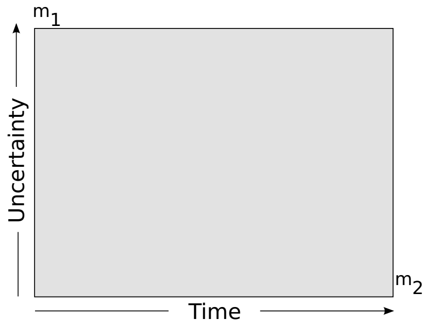
In this context process managers face two major questions: (1) How many EMPs have to be installed in order to reach a certain level of monitoring quality? (2) Where to optimally position the given number of EMPs to achieve best prediction quality of time until completion of a process instance? In this paper, we address the latter question and present an optimization algorithm for sequential processes. The applicability of our approach is discussed in the context of a surgical care process of a Dutch hospital. Since the operating room of a hospital is its most costly asset [MVDM95], hospitals try to maximize utilization and avoid idle times. High prediction quality for the end of the surgeries is crucial to allow rescheduling of surgeries and resources in case of deviations from the surgery schedule.

The paper is structured as follows. In Section 2 we introduce the problem and show how selection of EMPs influences the overall prediction quality of process duration. Related work is discussed in Section 3. Section 4 provides a formal description of the approach indicating where EMPs should be optimally allocated and describes the implementation of the proposed algorithm. In Section 5, we evaluate the applicability of the approach based on the use case data mentioned above. Finally, we conclude the paper and look on future work in Section 6.

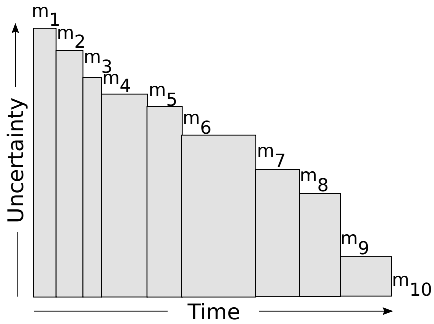
2 Problem

In a manual process execution environment, a trade-off between monitoring effort and prediction quality has to be made. In order to quantify the quality of the prediction regarding the process completion, uncertainty is considered. Uncertainty is the expected deviation of the actual process end to the estimated process end, e.g., estimated process completion will be at 6 p.m. with an expected deviation of +/-1hr. In this case, the uncertainty is 1 hour. Uncertainty can be measured for example based on the observed variance in historical data or by quantifying the error of future predictions, e.g., with measuring the mean square error.

For the prediction of the process completion time, at least two monitoring points are necessary, at the beginning and at end of a process. Figure 1 illustrates the uncertainty over time when utilizing two EMPs m_1, m_2 in Fig. 1(a), resp. ten EMPs $m_1, \ldots, m_{10}$ in Fig. 1(b). The distance between the EMPs is the mean duration between them. In the basic scenario with two EMPs at start and end, the monitoring effort is minimal for estimating process duration, but it gives only a very rough prediction regarding process completion time. The uncertainty in this scenario stays over the whole process execution time equal to the initial uncertainty of the mean process duration. Assuming the uncertainty is scaled, the initial maximum value is always 1 and the value at completion of a process instance is always 0.



(a) Minimal monitoring effort



(b) Optimal Quality

Figure 1: Distribution of uncertainty in dependence of the number of EMPs. The width of a column is the mean duration between the EMPs, the height is a measure of uncertainty, e.g., mean square error. In (a), only start and end of a process instance is measured. In (b), all possible EMPs are set up to reach the highest possible prediction quality.

Further EMPs can be installed in the process model at well-defined places. For simplicity reasons, we assume in this paper that it is possible to measure at the start of a process instance as well as at the termination of each activity. In a sequential process, the uncertainty over time regarding the process completion time is decreased by each additional EMP. Thereby, it is assumed that the activities' duration do not correlate with each other. In Figure 1(b) *all ten possible* EMPs for the process model depicted in Figure 2 are considered, i.e., EMPs at the start of a process as well as at the end of each activity. Here, the optimal prediction quality with the lowest possible uncertainty over time is achieved for this process. However, this setup will produce the highest effort of monitoring.

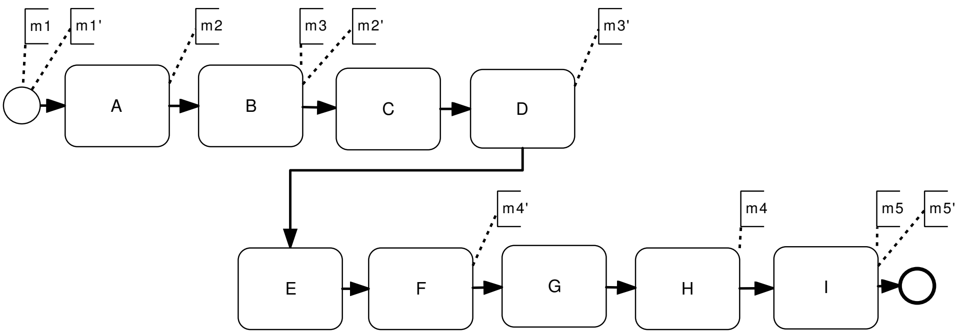
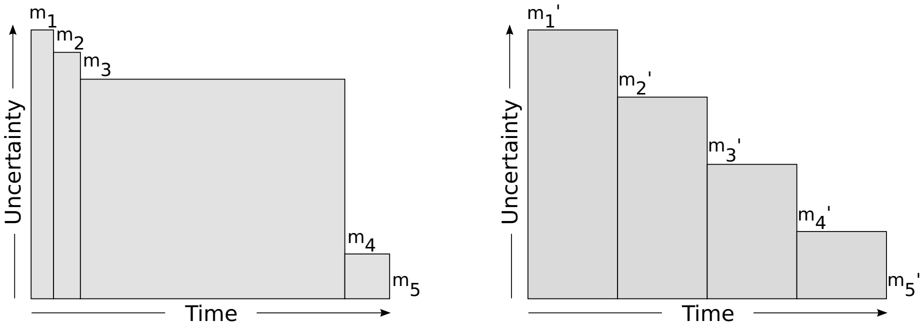


Figure 2: Process Model with differently allocated EMPs $m_1, \ldots, m_5$ (scenario 1) and $m'_1, \ldots, m'_5$ (scenario 2)

Beyond the number of EMPs, the allocation of them has also an important influence on the prediction quality. Figure 2 represents a process model where five EMPs are distributed differently. In the first scenario, the EMPs accumulate at the beginning and the end of the process ($m_1, \ldots, m_5$), whereas in the second one the EMPs are placed more homogenously over the process ($m'_1, \ldots, m'_5$). Figure 3 depicts the decreasing uncertainties with five implemented EMPs for the two mentioned scenarios in the diagrams. While comparing the overall uncertainties of the two configurations (i.e., the sum of the areas of the uncertainty bars for each EMP), it can be observed that the overall uncertainty over time is much higher in the first scenario, see Figure 3(a). There is a long period between m_3 and m_4 with high uncertainty which makes the overall uncertainty in the process higher than in Figure 3(b).



(a) Inhomogeneous monitoring (b) Homogeneous monitoring (optimal)

Figure 3: The overall uncertainty of the process, depicted in Figure 2, with five differently allocated EMPs $m_1, \ldots, m_5$ (scenario 1) resp. $m_1', \ldots, m_5'$ (scenario 2). The four columns indicate the uncertainty of the prediction for the remaining process execution time.

Before presenting an algorithm for sequential processes which assist process managers to find an optimal allocation for a given number of EMPs to keep the overall uncertainty minimal, we present related work upon which we build.

3 Related Work

In a complementary work [RSW12], the authors described how process progress can be estimated in process models with sparse EMPs. The authors assume known probability density functions for activity durations and answer the question, to which state a process instance progressed over time on a probabilistic basis. Our approach can be used to improve the estimation quality in [RSW12] by an optimal placement of EMPs. The underlying architecture feeding the EMPs in the process models with data from scattered information throughout the IT system landscape was presented in [HKRS12]. The same architecture is assumed as the basis for the presented approach in the paper at hand.

The question, when a running process instance will be finished, is addressed by van der Aalst et al. in [ASS11] with the concept of process mining. The authors propose to use historic process execution data captured in the event logs of supporting IT systems for predicting the completion time of a running instance. Their approach starts with building up a state transition system for the respective process based on the event log. In a next step, each state of the transition system is annotated with statistical information, e.g., mean, and variance, about the remaining time until completion from this state. This information is learned from historic instances which have visited the state. The annotated transition system can then be used for making predictions for running instances. Hereby, a fully or partly automated process environment is assumed, where historic data can be simply derived from the event logs. In contrast, in this paper, we do not intend to predict completion time of running instances, we rather seek to minimize the uncertainty of the prediction under given budget constraints.

A related problem statement can be found in the research domain of project management discussing the optimal timing of control points within a project. Control points are moments in a project where control activities are conducted for measuring the actual project state against the project plan. On the one hand, control activities are important, because they allow the detection of project deviations from the planned schedule and the implementation of corrective actions. On the other hand, they produce direct costs, are time-consuming and bind resources. Hence, similar decisions have to be made: the number of control points during a project and their allocation have to be planned. Partovi and Burton [PB93] evaluated in a simulation study the effectiveness of five different control timing policies: equal intervals, front loading, end loading, random and no control. Due to the heterogeneity of projects, no clear policy could be identified to be superior. De Falco and Macchiaroli argued that individual allocation of monitoring activities is required [FM98]. Therefore, they provided a method to determine quantitatively the optimal control point distribution by defining at first the effort function of a project based on activity intensity and slack time. Then, the control points are placed around the concentration of effort.

The concept of activity intensities was also used by Raz and Erel [RE00]. They determined the optimal timing of control activities by maximizing the amount of information generated by the control points. The amount of gathered information is calculated based on the intensity of activities carried out since the last control point and the time elapsed since their execution. The authors utilized dynamic programming in order to solve their optimizing problem. It seems promising to apply that approach to the optimal allocation of EMPs in a process model as well. The difference to control points being distributed over uniform intervals, e.g, days, is that the EMPs can only be positioned at well-defined places in a process model, i.e., at the end of activities.

Another application area for optimal allocation of control points is the diagnosability of systems. A diagnosis process of a system (e.g., a microprocessor) aims at the identification of reasons for unexpected behavior [CCGDV06]. A set of sensors (observations) and a model of the system is needed to detect system components being responsible for incorrect behavior. Existing sensors divide the system into clusters, whereby the connections between components inside of a cluster are not monitored, but the connections with components outside of a cluster. Ceballos et al. [CCGDV06] present in their research work a concept to allocate a set of new sensors in order to improve the system diagnosability. Their goal is to maximize the number of monitored clusters with a given number of additional sensors. Due to the exponential complexity of this maximization problem, the authors developed a greedy algorithm. This algorithm identifies bottlenecks of the system as the best candidates for allocating new sensors. This approach was transferred to business processes in the work of Borrego et al. [BGLGC10]. Installed control points within a process can help to identify which activities are responsible for deviating behavior from the process model. For their allocation, the authors refer to the proposed algorithm in [CCGDV06]. This algorithm was already used for business processes and focuses on increasing the number of monitored activity clusters. However, a maximum number of activity clusters does not necessarily yield an optimal prediction quality. In the next section we address this issue.

4 Approach

In order to ensure high prediction quality in a certain effort frame we developed and implemented an approach including an algorithm that allocates EMPs in a process model according to the given input. The algorithm requires as input (i) a process model, (ii) data about the execution time of that model (historical records or simulated), (iii) the number of EMPs that should be allocated, and (iv) the uncertainty function that should be used for calculation. In order to describe the approach formally, we first introduce in Section 4.1 some preliminary notions. Afterwards, we describe the algorithm in Section 4.2 and present in Section 4.3 the implementation of our approach.

4.1 Preliminaries

In this paper we define the process model as a connected graph consisting of a set of activities A and control flow edges F beginning with a start event e_s and terminating with an end event e_e.

Definition 1 (Sequential Process Model) *A process model is a tuple $P = (A, F, e_s, e_e)$, where A is a set of activities, e_s is the start event and e_e is the end event of P. The flow relation is defined as $F \subseteq (\{e_s\} \cup A) \times (\{e_e\} \cup A)$ and captures the ordering constraints of the activity execution.*

In a sequential process model it holds that each node can have at most one incoming and one outgoing control flow edge, i.e.,
$(x, y) \in F \wedge (x, y') \in F \Rightarrow y = y'$ *and*
$(x, y) \in F \wedge (x', y) \in F \Rightarrow x = x'$
Hence, the flow relation yields an ordering of activities A in the process model, where $(x, y) \in F \Leftrightarrow x < y$, i.e., y can only be executed, when x has been terminated.

In a process model P event monitoring points (EMPs) can be allocated for correlating process execution information at well-defined places. In this paper, we assume that the start and end of a process are known, i.e., an EMP exists at the start of a process as well as at the end of the last activity a_n. Further EMPs can be allocated to the termination state change of all other activities of the process model.

Definition 2 (Event Monitoring Point) *Let P be a process model with the set of activities $A = \{a_1, \ldots, a_n\}$. The set of possible EMPs M is defined as the union of the EMP at the start of the process m_1, and the EMPs $m_2, \ldots, m_{(n+1)}$ capturing the termination of each individual activity $a_1, \ldots, a_n$.*

Thus, we define for a sequential process with n activities $n + 1$ possible EMPs. At these EMPs information about process execution is gained. When we observe the occurence of an event at the resp. EMP, the previously uncertain activity durations become certain at this point. Thus, the overall uncertainty is reduced from this EMP on for the remaining process duration. We want to be able to quantify this and define the mean of the remaining process duration and the uncertainty of these durations respectively.

Definition 3 (Mean of the Remaining Duration) *The remaining mean duration is a function $mean_{\mathrm{dur}}\colon M \to \mathbb{R}_0^+$ assigning to each EMP $m_i \in M$ the arithmetic mean of the durations from the time at m_i until the termination time of the process which is captured by the termination of the last activity in EMP $m_{(n+1)}$.*

When the remaining mean duration is calculated on a sample of observed values, it is usually subject to bias. This bias becomes less prominent with growing sample size due to the law of large numbers. We assume a large sample size of the historical execution data and do not consider therefore bias and statistical confidences of the observed mean. In this research work, we want to focus on the uncertainty of the predicted mean duration from the EMPs until the process end.

Definition 4 (Uncertainty of the Remaining Duration) *Let $u_{\mathrm{dur}}\colon M \to \mathbb{R}_0^+$ be the uncertainty function assigning a non-negative value to an EMP capturing an uncertainty measure of the remaining process duration.*

This definition does not limit to a specific uncertainty function, as there are many potential ways to measure and calculate the uncertainty of the remaining duration, e.g., by the variance (VAR). Estimation and prediction is a broad field of operational research and many measures have been introduced, e.g., Mean Square Error (MSE), Root Mean Square Error (RMSE). An overview can be found in the research work by Hyndman and Koehler [HK06].

Mean duration as well as the uncertainty is relatively scaled, so that the maximum is 1, in order to ensure comparability. With these notions, we can quantify the relative overall uncertainty in the process.

Definition 5 (Overall Uncertainty) *Let P be a process model with the set of activities $A = \{a_1, \ldots, a_n\}$ and the corresponding set of possible EMPs $M = \{m_1, \ldots, m_{n+1}\}$. The overall uncertainty U in the process is defined as:*

$$U(m_1, \ldots, m_{n+1}) = \sum_{i=1}^{n} u_{\mathrm{dur}}(m_i) \cdot (mean_{\mathrm{dur}}(m_i) - mean_{\mathrm{dur}}(m_{i+1}))$$

Note that by this definition, we interpret the overall uncertainty as the area under the stair-shaped uncertainty figures, cf. Figure 1 and Figure 3.

4.2 Algorithm for optimal placement of EMPs

In the following, the algorithm for the optimal placement of a number of EMPs is presented, such that the resulting overall uncertainty is minimal. The algorithm will closely follow the proposed approach in [RE00] for the optimal placement of control points in projects. In contrast to [RE00] in which the given control points are set at arbitrary intervals (e.g., on a per day basis), EMPs in processes can only be set at well defined positions, i.e., at the termination of an activity. Hence, the number of activities limit the maximum number of EMPs. When all EMPs are installed in a process, i.e., monitoring the start of the process and each activity's termination, the highest possible prediction quality can be achieved. The presented algorithm indicates where to implement a given number of EMPs in order to decrease the overall uncertainty of the prediction.

The problem of selecting k EMPs optimally out of n potential EMPs for overall maximal certainty of the prediction of process completion is computationally complex. There are $\binom{n}{k}$ solution candidates. The problem can be divided into computing local optimal solutions recursively. The local optimal solution only depends on the previously allocated EMP. Thus, we can store intermediate optimal solutions and skip calculating these again for the other combinations with the same previously allocated EMP. This makes a dynamic programming approach, as also proposed in [RE00], feasible and the problem can be described as follows.

The idea is to minimize the overall uncertainty U that depends on where the EMPs are installed. We do this by looking at the complementary problem of maximizing the reduction of the uncertainty. Note that the allocation of an additional EMP m_j reduces the uncertainty of the prediction by the uncertainty portion which lies between m_j and the previous EMP m_i. This decrease of uncertainty applies for the remaining mean duration $mean_{\text{dur}}(m_j)$. The reduction of uncertainty $\bar{U}$ can be interpreted as the white area that complements the overall uncertainty to 1, i.e., $U + \bar{U} = 1$. Thus, the overall reduction of the uncertainty $\bar{U}$ of all allocated EMPs can be defined as:

$$\bar{U}(m_1, \ldots, m_{n+1}) = \sum_{i=2}^{n} (u_{\text{dur}}(m_{i-1}) - u_{\text{dur}}(m_i)) \cdot mean_{\text{dur}}(m_i) \tag{1}$$

For the basic setup we always need two initial EMPs for capturing the start and the end of a process, therefore i starts with 2 in Equation (1). Thus, we formulate the problem as maximizing the reduced uncertainty $\bar{U}$ for a given number k of EMPs $(2 < k \leq n$. Let m_i denote the previous EMP. Let $\bar{U}(m_i, m_j)$ be the uncertainty removed by introducing the next EMP m_j:

$$\bar{U}(m_i, m_j) = (u_{\text{dur}}(m_i) - u_{\text{dur}}(m_j)) \cdot mean_{\text{dur}}(m_j) \tag{2}$$

We define the maximal reduced uncertainty by allocating one additional EMP, when the last EMP is m_i, as $\bar{U}_1^*(m_i)$:

$$\bar{U}_1^*(m_i) = \operatorname*{Max}_{i<j\leq n} \{\bar{U}(m_i, m_j)\} \tag{3}$$

We are interested in finding the particular EMP that maximizes the reduction of uncertainty of the prediction when m_i is the previous EMP, i.e., we want to find the argument that is responsible for the maximum in Equation (3):

$$m_{j_1}^* | m_i = \operatorname*{arg\,max}_{i<j\leq n} (\bar{U}(m_i, m_j)) \tag{4}$$

At this stage, we can describe the optimal solution for implementing one additional EMP. However, the problem is more complex, as we are also interested in the optimal placement of two or more EMPs, given the last EMP is already set. We denote this multiplicity with an index, i.e., $\bar{U}_2^*(m_i)$ for the maximum gained certainty with two additional EMPs after the EMP m_i.

$$\bar{U}_2^*(m_i) = \underset{i<j\leq n-1}{\text{Max}} (\bar{U}(m_i, m_j) + \bar{U}_1^*(m_i)) \tag{5}$$

Further, we define the maximum reduced uncertainty $\bar{U}_k^*(i)$ for a number k of additional EMPs recursively:

$$\bar{U}_k^*(m_i) = \underset{i<j\leq n-(k-1)}{\text{Max}} (\bar{U}(m_i, m_j) + \bar{U}_{k-1}^*(m_j)) \tag{6}$$

We are further interested in the position of the next EMP in the sequence that yields the maximum decrease in uncertainty, given that the previous EMP is m_i.

$$m_{j_k}^*|m_i = \underset{i<k\leq n-(k-1)}{\arg\max} \; (\bar{U}(m_i, m_j) + \bar{U}_{k-1}^*(m_j)) \tag{7}$$

With this notation, we can formulate the problem introduced in Section 2, as to compute $\bar{U}_{(k-2)}^*(m_1)$ (cf. Equation 6), i.e., the maximum reduction of uncertainty gained by k EMPs, given that two of them measure the start and the end of the process, and return the arguments $m_{j_{(k-2)}}^*, m_{j_{(k-3)}}^*, \ldots, m_{j_1}^*$, cf. Equation (7). In order to solve this problem, the algorithm pursues the following steps:

1. Determine the set M of potential EMPs in the given process model.
2. For each $m_1, \ldots, m_{n+1} \in M$ calculate the remaining mean duration $mean_{\text{dur}}(m_i)$ until process termination based on given historical execution data.
3. Calculate uncertainty $u_{\text{dur}}(m_i)$ of remaining durations for each identified potential EMPs based on historical execution data according to the given uncertainty function.
4. Compute $\bar{U}_{(k-2)}^*(m_1)$ for the given number of requested EMPs k by using dynamic programming for searching through the $\binom{n}{k}$ solution combinations. Thereby, intermediate computed optima are stored to save time by not recomputing such solutions.

Note that the presented algorithm can be also utilized to determine the required EMPs to meet a given uncertainty threshold. Therefore, the algorithm has to be executed iteratively by incrementing k starting with 2 until the threshold is met. This is shown exemplarily Section 5.

The described algorithm is used in the implementation which is presented in the next section.

4.3 Implementation

We implemented the approach for sequential processes in ProM [DMV+05]. Our developed ProM plug-in needs as input an event log of the respective process, which provides the historical execution data, consisting of the case start time and activities termination time for a set of process instances. Usually, event logs are provided in an automated process execution environment by information systems. In a manual process environment, an event log can be created by two different possibilities:

- Recording execution data over a certain period of time in the real process environment
- Performing a simulation based on an annotated process model including performance models for activities, e.g., probability density functions for durations, and collecting the simulated execution data, e.g. with CPN tools [JKW07]

No process model is needed by the implementation as it uses the ProM functionality and derives the model from the event log as proposed in [ASS11]. Furthermore, the developed plug-in has two user parameters: (1) selected uncertainty function and (2) the number of desired EMPs.

As soon as the user has picked one of the provided *uncertainty functions* $u_{\mathrm{dur}}(M)$ (e.g., VAR, MSE, RMSE), the overall uncertainty is shown for all possible EMPs in M of the process model as the optimal stair-shaped uncertainty graph to the user, as depicted in Figure 6.

Secondly, the user can select *a number of EMPs*, which should be distributed optimally between a range from two (only start and end are monitored) to the size of possible EMPs in the process. The implementation uses dynamic programming for searching the optimal solution of the $\binom{n}{k}$ combinations as described by the algorithm in the Section 4. The resulting optimal EMPs are highlighted at the x-axis to the user and the resulting uncertainty graph based on the selection is laid behind the optimal graph in gray color for comparison. Additionally, the plug-in provides numeric values for the overall uncertainty U of the respective graphs in order to assist the user in reasoning on the required number of EMPs.

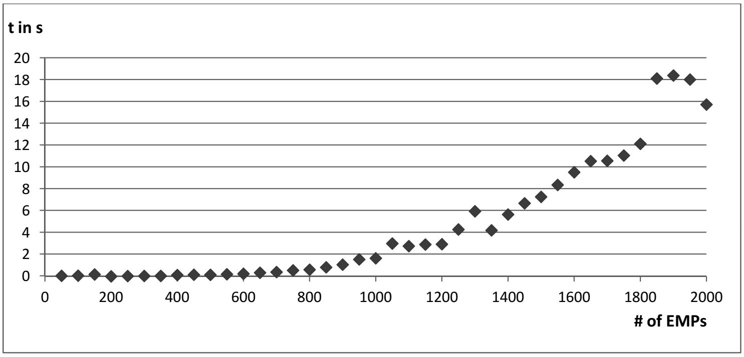


Figure 4: Algorithm runtime performance with sequential models with an increasing number of EMPs

In order to test the performance of the algorithm, we generated random stair shaped uncertainty graphs with $n = 50, 100, \ldots, 2000$ nodes and calculated the optimal position of $k = \frac{n}{10}$ EMPs on a typical laptop computer. The optimization algorithm can deal with model sizes up to 2000 nodes, but runs into memory shortages above. This is no practical issue, as process models with such dimensions should be hard to find. As shown in Fig. 4, optimal positioning in models with a size of less than 1000 possible EMPs can be computed in under two seconds.

5 Case Study

In this section, the applicability of the developed optimization algorithm is shown with a use case of the surgical care process of a Dutch hospital.

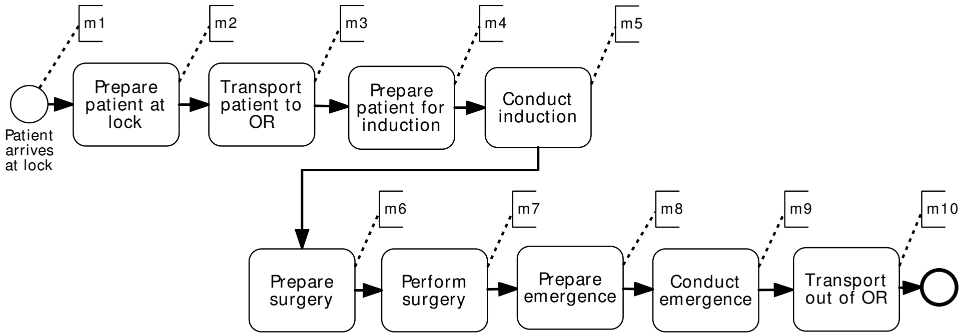


Figure 5: Surgical care process represented as BPMN diagram with annotated EMPs $m_1..m_{10}$

The process is modeled in a BPMN diagram, see Figure 5, and the currently existing EMPs, which are allocated at the start of the process and the termination of every activity, are annotated. The surgical care process starts with the arrival of the patient in the lock measured by the first EMP m_1. After the patient was prepared in the lock (m_2), he or she is transported into the operating room (m_3). The activities in the operating room start with preparing the induction of the anesthesia (m_4) followed by its conduction (m_5). As soon as all preparations for the surgery are finished (m_6), the surgical procedure is performed (m_7). Afterwards, the emergence (m_8) is prepared and the patient is emerged from his/her anesthesia (m_9). Once the patient is transported out of the operating room (m_{10}), the process ends.

All annotated EMPs are captured manually by process participants in an IT system. However, some execution data records of EMPs were empty due to human failure. Therefore, we conducted a preprocessing step in which we filtered out traces having no start or end time, because the calculation of the process mean duration and its uncertainty is not possible for such cases. In a next step, we imported the historical execution data of 935 instances for the described process into the ProM tool and applied our developed plug-in to the data. The result is shown in Figure 6. The overall uncertainty U—based on the variance here—of the prediction regarding the completion time of a surgical case for this process is 0.7583.

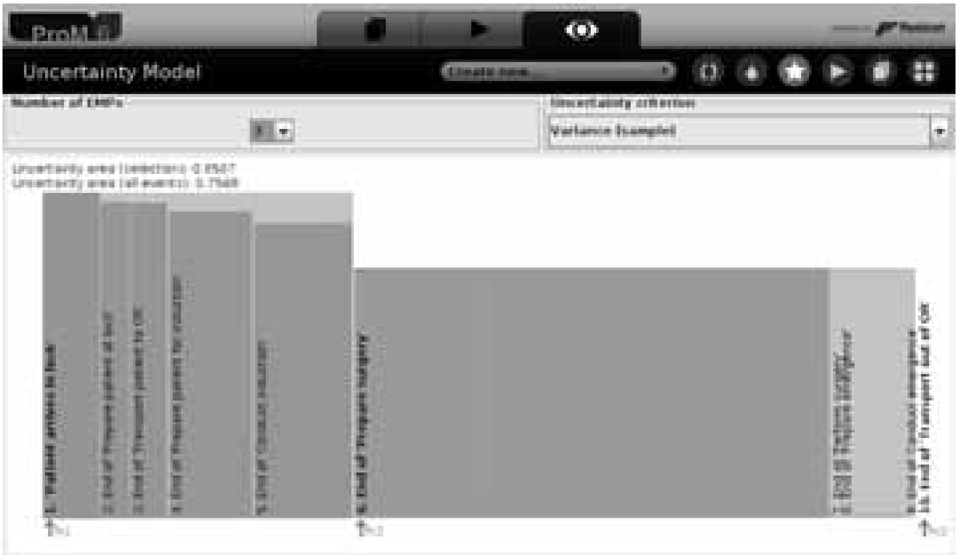


Figure 6: Screenshot of the results by optimization algorithm implemented in ProM

The graph makes explicit that the activity with the highest uncertainty over time is the *performance of the surgery*. This can be explained by having a closer look on the execution data. The data set of the Dutch hospital contains heterogeneous types of surgeries, e.g., excision of the tonsils, breast neoplasm surgery, which show different operating times.

In a next step, the algorithm was run for the optimal allocation of three EMPs, whereby the first two have their standard position at the beginning and end of the process. The third EMP was allocated optimally by the algorithm at the end of the *surgery preparation* which is shown in Figure 6. The reduction from ten to three EMPs leads to an increase of the overall uncertainty to 0.8509. Hence, if the management of the surgery department accepts roughly 0.1 more overall uncertainty, seven monitoring points could be saved. Thus, saving efforts for manually capturing those EMPs results in slightly worse estimation quality. The participants—mainly nurses, which are in charge of recording the times—would be able to focus more on their main task and on the patients.

The following graph in Figure 7 represents the decrease of uncertainty over time by adding additional EMPs starting with two, i.e., monitoring only the start and end of a process. Note that, while every additional EMP can add to the reduction of the overall uncertainty, the first few do so with bigger impact. The highest decrease in uncertainty for this surgical care process can be gained with the third (0.15) and the fourth EMP (0.07). With every additional EMP, the decrease of uncertainty converges to the maximal value of 0.24.

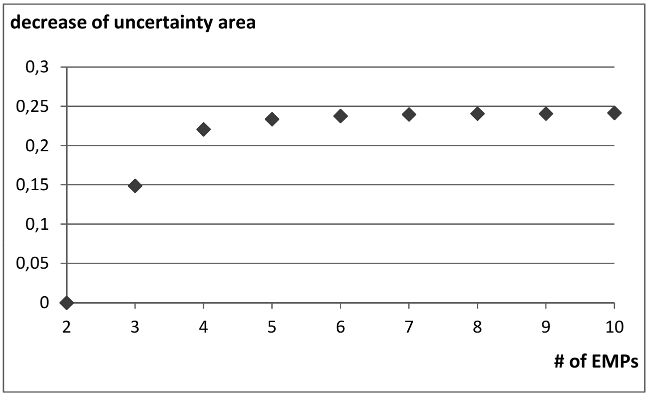


Figure 7: Decrease of uncertainty with increasing number of EMPs

This use case shows that a reduction by six to seven EMPs would lead only to a small increase of uncertainty regarding the remaining mean duration, but it would help to save efforts recording them manually. Furthermore, the algorithm supports the surgery management in deciding where a number of EMPs should be installed.

6 Conclusion and Outlook

A cornerstone of today's business process management is *business process monitoring* ensuring the quality of process execution and preventing deviations from the organization's goals and service level agreements. Especially in rather manual process enacting domains, e.g., healthcare, monitoring of processes challenges the organizations in balancing the efforts for monitoring and the quality of the predictions based on monitoring. In this paper, we presented an approach to find the optimal allocation of a given number of EMPs in sequential process models for minimizing the uncertainty of the predicted time until completion, thus gaining the best prediction quality according to a flexible uncertainty measure. The number of EMPs varies between two (i.e., start of process and end of the last activity) and all possible measurable points in the process model (i.e., start of process and termination of all activities), depending on the efforts the organization is willing to spend for process monitoring.

Our proposed optimization algorithm was implemented in the ProM tool. We showed that the runtime complexity is well manageable and works on a laptop computer for up to 2000 possible EMPs with a selection of 200 EMPs in a sequential process model that should suffice for almost any process in practice. A more important limitation of this approach is the support of sequential processes only. This is due to the fact that there exist complex relations between EMPs on parallel branches in execution that influence the uncertainty about the state of a process, as shown in [RSW12]. In this paper, we focused first on the sequential case, but we plan to lift this limitation in future work, and allow also control flow structures, e.g., exclusive and parallel gateways, or loops.

We assumed in this research work that there exists no correlation between single activities in order to assure a decreasing uncertainty over time with each additional EMP. However, positive as well as negative correlation between the duration of activities can be observed in practice. An example for a positive correlation is the surgery preparation activity which usually needs more time, if the surgery takes longer and is more complicated. A negative correlation can be observed for instance in processes with a certain deadline; activities are processed faster when previous activities took longer and deadlines get closer. Considering correlation of activities would be very valuable for predictions and is thus worth to look at in future research work.

References

[ASS11] W.M.P. van der Aalst, M.H. Schonenberg, and M. Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.

[BGLGC10] D. Borrego, M.T. Gomez-Lopez, R.M. Gasea, and R. Ceballos. Determination of an optimal test points allocation for business process analysis. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 159–160. IEEE, 2010.

[CCGDV06] R. Ceballos, V. Cejudo, R. Gasca, and C. Del Valle. A topological-based method for allocating sensors by using CSP techniques. *Current Topics in Artificial Intelligence*, pages 62–68, 2006.

[DMV$^+$05] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A new era in process mining tool support. *Applications and Theory of Petri Nets 2005*, pages 1105–1116, 2005.

[FM98] M. de Falco and R. Macchiaroli. Timing of control activities in project planning. *International Journal of Project Management*, 16(1):51–58, 1998.

[HK06] R.J. Hyndman and A.B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[HKRS12] N. Herzberg, M. Kunze, and A. Rogge-Solti. Towards Process Evaluation in Non-automated Process Execution Environments. In *Proceedings of the 4th Central-European Workshop on Services and their Composition, ZEUS*, 2012.

[JKW07] K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(3):213–254, 2007.

[LR07] R. Lenz and M. Reichert. IT Support for Healthcare Processes – Premises, Challenges, Perspectives. *Data & Knowledge Engineering*, 61(1):39–58, April 2007.

[MVDM95] A. Macario, T.S. Vitez, B. Dunn, and T. McDonald. Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. *Anesthesiology*, 83(6):1138, 1995.

[PB93] F.Y. Partovi and J. Burton. Timing of monitoring and control of CPM projects. *Engineering Management, IEEE Transactions on*, 40(1):68–75, 1993.

[RE00] T. Raz and E. Erel. Optimal timing of project control points. *European Journal of Operational Research*, 127(2):252–261, 2000.

[RSW12] A. Rogge-Solti and M. Weske. Enabling Probabilistic Process Monitoring in Non-automated Environments. In *Enterprise, Business-Process and Information Systems Modeling*, volume 113 of *LNBIP*, pages 226–240. Springer, 2012.

Grundlegende Überlegungen zum adäquaten Detaillierungsgrad von Geschäftsprozessmodellen

Frank Termer, Volker Nissen, Sven Wessels

Fachgebiet Wirtschaftsinformatik für Dienstleistungen
Technische Universität Ilmenau
Postfach 10 05 65
98684 Ilmenau
frank.termer@tu-ilmenau.de
volker.nissen@tu-ilmenau.de
sven.wessels@tu-ilmenau.de

Abstract: Dieser Beitrag befasst sich mit einigen grundsätzlichen Überlegungen im Vorfeld der Definition adäquater Detaillierungsgrade von Geschäftsprozessmodellen für unterschiedliche Modellierungszwecke. Dabei stehen Fragen zum Verständnis des Detaillierungsbegriffes, Einflussfaktoren auf den zweckmäßigen Detaillierungsgrad sowie ein mögliches Bewertungsschema der Modelldetaillierung im Fokus. Diese Grundlagen sollen helfen, im nächsten Schritt adäquate Detaillierungsgrade in Abhängigkeit vom jeweiligen Zweck der Prozessmodellierung zu finden.

1 Ausgangslage und Motivation

Nach ROSEMANN [Ro96, 9] stellt ein Prozess die inhaltlich abgeschlossene, zeitliche und sachlogische Abfolge von Funktionen dar, die zur Bearbeitung eines betriebswirtschaftlich relevanten Objekts ausgeführt werden. Im Rahmen des Geschäftsprozessmanagements kommt der Prozessmodellierung eine zentrale Bedeutung zu. Ein Prozessmodell dient der Abstraktion des realen Sachverhalts zu bestimmten Modellierungszwecken [Ro96, 17]. Mit Hilfe von Prozessmodellen kann ein Geschäftsprozess dokumentiert und ein Teil des in ihm enthaltenen Wissens expliziert werden [Al07, 23]. Für die Modellierung von Prozessen stehen mehrere grafische Modellierungstechniken zur Verfügung, die jeweils ein fest definiertes Regelwerk für die Ausgestaltung der Modelle besitzen. Die wichtigsten Techniken zur Prozessmodellierung sind die Ereignisgesteuerte Prozesskette (EPK) und ihre Weiterentwicklung, die erweiterte Ereignisgesteuerte Prozesskette (eEPK), die Business Process Modeling Notation (BPMN), die Aktivitätsdiagramme der Unified Modeling Language (UML), Petri-Netze und die PICTURE-Methode [Ga10, 71].

In diesem Zusammenhang stellt sich die Frage, wie detailliert ein Prozess modelliert werden sollte. Die Frage leitet sich aus den von BECKER ET AL. [BRS95, 435-445] definierten Grundsätzen ordnungsmäßiger Modellierung (GoM) ab, die sich im Kontext der Prozessmodellierung als De-facto-Standard für die Qualitätssicherung von

Prozessmodellen etabliert haben. Durch den Grundsatz der Relevanz wird die Frage aufgeworfen, wie umfangreich die Modellierung in Abhängigkeit vom verfolgten Modellierungszweck zu erfolgen hat. Dem steht der Grundsatz der Wirtschaftlichkeit begrenzend entgegen. Dieser besagt, dass der Aufwand für die Erstellung eines Modells in einem angemessenen Verhältnis zum Modellnutzen stehen muss. BECKER ET AL. [BRS95, 444] stellen klar, dass die Grundsätze ordnungsmäßiger Modellierung sehr allgemein gehalten sind und durch konkrete Gestaltungsempfehlungen näher bestimmt werden müssen.

Die Auswahl der adäquaten Modelldetaillierung ist heute eines der zentralen Probleme der Prozessmodellierung, wie eine Studie über die derzeitigen Probleme und zukünftigen Herausforderungen im Geschäftsprozessmanagement ergeben hat [IRR09, 9]. Auch GADATSCH [Ga10, 211-212] identifiziert den adäquaten Detaillierungsgrad als aktuelles Modellierungsproblem in der Praxis. Einerseits können Prozessmodelle sehr grob modelliert werden, etwa um als Basis für strategische Überlegungen einen Überblick über die derzeitige Prozesslandschaft eines Unternehmens zu erhalten. Wird das Prozessmodell hingegen zum Zwecke einer Simulation oder für die Automatisierung erstellt, muss es sehr detailliert ausfallen. In der Literatur finden sich zum Problem der adäquaten Detaillierung allerdings meist nur Aussagen zur hierarchischen Untergliederung von Prozessen in verfeinerte Teilprozesse. Die Hierarchisierung von Prozessmodellen wird häufig synonym mit dem Begriff des Detaillierungsgrades verwendet. Andere mögliche Aspekte der Detaillierung, wie etwa ergänzende Informationsobjekte oder die Anreicherung der Prozesselemente mit Attributinformationen werden selten thematisiert. Eine strukturierte Bearbeitung des Detaillierungsproblems findet nach unserer Erfahrung bisher nicht statt.

BECKER ET AL. [BRS95, 440] gehen davon aus, dass aufgrund der umfangreichen Verwendungsmöglichkeiten von Prozessmodellen die Festlegung des Detaillierungsgrades nur in Abhängigkeit des jeweiligen Verwendungszweckes möglich ist. Da ein einzelnes Prozessmodell häufig verschiedenen Modellierungszwecken gerecht werden soll, stellt sich die Wahl des adäquaten Detaillierungsgrades als sehr komplexes Problem dar [Ro08, 57]. Zusätzlich sorgt die Beteiligung heterogener Anwenderkreise an der Modellierung für unterschiedliche Anforderungen an den Detaillierungsgrad [Ro96, 2]. Es ergibt sich der Bedarf nach einer strukturierten Untersuchung der Detaillierungsproblematik in Abhängigkeit vom Modellierungszweck und etwaiger weiterer Einflussfaktoren. Es ist die Forschungsfrage zu klären, welche Prozessmodelldetaillierung in Abhängigkeit von verschiedenen Modellierungszwecken angemessen ist. Dieser Beitrag widmet sich grundlegenden Vorüberlegungen mit dem Ziel, den Detaillierungsbegriff zu klären, Einflussfaktoren auf den zweckmäßigen Detaillierungsgrad zu identifizieren sowie ein mögliches Bewertungsschema der Modelldetaillierung zu entwerfen.

2 Analogie zur Kartografie

Um die Problematik der adäquaten Abstraktion von Prozessmodellen zu erklären, lässt sich die Analogie zur Kartografie nutzen. Prozessmodelle können als Landkarten interpretiert werden, die die Prozesse eines Unternehmens beschreiben [vA11, 321-335].

Kartografen versuchen in einer Karte, die für ihren Einsatzzweck relevanten Informationen auszudrücken. Nicht benötigte Informationen gilt es hingegen zu eliminieren und insgesamt die Komplexität zu reduzieren, während die Verständlichkeit verbessert werden soll.

Zur Reduktion der Komplexität und Erhöhung der Übersichtlichkeit existieren in der Kartografie mit dem hierarchischen Strukturieren und der Sichtenbildung zwei wichtige Prinzipien. Beim hierarchischen Strukturieren entstehen durch Abstraktionen verschiedene Ebenen des abzubildenden Systems. Jede dieser Ebenen besitzt einen bestimmten Detaillierungsgrad und enthält nur die Informationen, die für die Zwecke dieser Ebene sinnvoll sind. In der Kartografie reichen die Beispiele von Weltkarten, welche die höchste Abstraktionsstufe darstellen, über Kontinentkarten, Länderkarten, Regionenkarten bis hin zur Ebene der Stadtpläne. Die Sichtenbildung dient ebenfalls der Komplexitätsreduktion. Sie setzt den Fokus auf bestimmte Teilaspekte des Gesamtsystems, um diese beispielswiese in einer besonderen Form darzustellen oder um bestimmte Sachverhalte auszublenden und so nur die für die Sicht relevanten Inhalte darzustellen [Br05, 19-22].

Prozessmodelle müssen, ebenso wie Karten, von weniger wichtigen Dingen abstrahieren. Wenn eine Aktivität in einem Prozessmodell nur selten zur Ausführung kommt, liegt die Überlegung nahe, sie aufgrund geringer Relevanz zu entfernen. Karten abstrahieren jedoch nicht nur von weniger wichtigen Details, sie aggregieren auch Details zu übergeordneten Einheiten. Je nach Verwendungszweck können die Karten zudem um ergänzende Informationen angereichert werden. Beispielsweise sind Angaben über das Streckennetz von öffentlichen Verkehrsmitteln denkbar oder auch Standorte von Restaurants, inklusive ihrer Speisekarte, Öffnungszeiten, Bewertung, etc. Die zusätzlich in der Karte darstellbaren Details sind nahezu unbegrenzt. Diese Detaillierungsmöglichkeiten ergeben sich in ähnlicher Weise auch für die Prozessmodellierung. Hier können zusätzlich zu den einzelnen Prozessschritten weitere Informationen in den Modellen festgehalten werden, wie beispielswiese die beteiligten Organisationseinheiten oder die Daten, die für die Bearbeitung des jeweiligen Schrittes benötigt werden. Analog zur Kartografie stellt sich hier abhängig vom konkreten Verwendungszweck die Frage, welche Informationsobjekte relevant sind und wie detailliert diese Objekte ausgestaltet werden müssen.

3 Methodische Vorgehensweise

Methodische Basis der Arbeit ist eine Literaturrecherche nach FETTKE [Fe06, 260-261]. Als Grundlage dienten hierfür wissenschaftliche Publikationen zum Thema der Prozessmodellierung der vergangenen Jahre. Zu diesem Zweck wurden thematisch relevante Beiträge der Fachzeitschriften HMD - Praxis der Wirtschaftsinformatik und WIRTSCHAFTSINFORMATIK sowie Beiträge der Konferenz EPK – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten und der Konferenz Modellierung betrieblicher Informationssysteme (MobIS) sowie weitere Basisliteratur im Kontext der Grundsätze ordnungsmäßiger Modellierung untersucht. Als Ausgangsliteratur wurden die Arbeiten von ALLWEYER [Al05], BECKER ET AL.

[BMW09], BECKER ET AL. [BKR08], DAVIS [Da05], FREUND ET AL. [FRH10], GADATSCH [Ga10], GAITANIDES [Ga83], KUGELER [Ku00] und ROSEMANN [Ro96] verwendet. Eine vorwärts und rückwärts gerichtete Literatursuche diente der Identifikation der potentiell relevanten Literatur für die Beantwortung der Forschungsfrage. Durch eine Grobsichtung der Literatur hinsichtlich Aussagen zum Detaillierungsgrad von Prozessmodellen wurde im Anschluss die Eignung der Quellen beurteilt. Bei entsprechender Eignung wurden in wiederholten Iterationsschritten der vorwärts- und rückwärtsgerichteten Suche weitere relevante Quellen identifiziert. Um der Gefahr zu begegnen, einen isolierten Autoren- und Meinungskreis zu erhalten, sind in Ergänzung auch Schlagwortsuchen in einschlägigen Datenbanken und Suchmaschinen zur Auffindung wissenschaftlicher Literatur mit den Begriffen „Detaillierungsgrad" und „Abstraktionsgrad" in Verbindung mit „Prozessmodellierung" durchgeführt worden. Bei entsprechender Eignung der Quellen wurde hier ebenso eine vorwärts und rückwärts gerichtete Suche nach weiteren Publikationen durchgeführt.

4 Zwecke der Prozessmodellierung

Prozessmodelle werden in Unternehmen für unterschiedliche Zwecke erstellt und genutzt. Für die Erarbeitung von Gestaltungsempfehlungen zum adäquaten Detaillierungsgrad ist es wichtig, diese Zwecke zu ermitteln und im Anschluss zu untersuchen. Als Ausgangspunkt dienen die bei ROSEMANN [Ro08, 51-58] identifizierten elf allgemeinen Modellierungszwecke, die in die beiden Kategorien Organisationsgestaltung und Anwendungssystemgestaltung eingeteilt werden. Zur Organisationsgestaltung zählen hierbei: Organisationsdokumentation, prozessorientierte Reorganisation, kontinuierliches Prozessmanagement, Zertifizierung, Benchmarking und Wissensmanagement. Der Anwendungssystemgestaltung werden zugerechnet die Modellierungszwecke Auswahl von ERP-Software, modellbasiertes Customizing, Softwareentwicklung, Workflowmanagement und Simulation.

In Ergänzung dazu können die bei ALLWEYER [Al05, 28-33] präsentierten verschiedenen Aufgaben von Geschäftsprozessen und ihre Beziehungen zu anderen Themenstellungen herangezogen werden (siehe Abbildung 1). Zwar stehen hier Geschäftsprozesse und nicht Geschäftsprozessmodelle im Zentrum, dennoch können Rückschlüsse auf mögliche Einsatzzwecke der Prozessmodellierung gezogen werden.

Abb. 1 Geschäftsprozesse im Betrieb [Al05, 28]

Mithilfe dieser Ausgangsbasis sind die in Tabelle 1 dargestellten Haupteinsatzzwecke von Prozessmodellen ermittelt worden. Gemäß dem Vorschlag von ROSEMANN [RO08, 57] erfolgt die Einteilung der Zwecke in die Kategorien Organisationsgestaltung und Anwendungssystemgestaltung. Abweichend von der ursprünglichen Sichtweise wird allerdings der Einsatzzweck Simulation in die Kategorie der Organisationsgestaltung eingeordnet, da diese nicht primär im Rahmen der Gestaltung von Anwendungssystemen zum Einsatz kommt, sondern verstärkt zur Beantwortung von organisatorisch-ablauftechnischen Fragestellungen genutzt wird.

Zusätzlich in diese Liste aufgenommen worden ist die Prozessmodellierung zum Zwecke der Personalbedarfsplanung und zur Prozesskostenrechnung. Die Hinzunahme dieser Modellierungszwecke scheint notwendig, um dem Stellenwert dieser Anwendungsbereiche in der Praxis gerecht zu werden und insbesondere diese Modellierungszwecke gegenüber der Simulation abzugrenzen. Die Simulation wird in unserem Verständnis vornehmlich dazu verwendet, um Schwachstellen im Prozessverlauf zu erkennen oder verschiedene inhaltliche Ablaufvarianten in Form von Prozessalternativen zu bewerten. Die Personalbedarfsplanung hingegen hat zum Ziel, für eine konkrete Prozessausgestaltung die zur Erbringung definierter Fallzahlen von Prozessen benötigten Arbeitskräfte zu ermitteln. Hierbei geht es auch um Sensitivitätsanalysen, bei denen ermittelt wird, wie sich unterschiedliche Mengengerüste der Prozesse auf den Personalbedarf und auch auf die Auslastung des Personals auswirken [Hü06, 123-131]. Die Prozesskostenrechnung hat zum Ziel, die Kosten des Durchlaufs einzelner Prozesse zu ermitteln, um im Rahmen der Kostenrechnung insbesondere die Gemeinkosten verursachungsgerecht zuordnen zu können. Sowohl die Personalbedarfsermittlung als auch die Prozesskostenrechnung unterscheiden sich also hinsichtlich Zielsetzung und Datengrundlage von der Simulation und sollen daher als eigenständige Modellierungszwecke explizit genannt und weitergehend betrachtet werden.

Die Auflistung in Tabelle 1 ist nicht als abgeschlossen zu betrachten. Prozessmodelle können auch in weiteren Szenarien zum Einsatz kommen, wie zum Beispiel als Planungsinstrument im Projektmanagement [Ro96, 45]. Tendenziell sind diese weiteren Anwendungszwecke allerdings von geringerer Bedeutung. Sie werden seltener in der Literatur erwähnt und in der Praxis genutzt oder die Prozessmodellierung lässt sich durch andere adäquate Beschreibungstechniken leicht substituieren.

Tab. 1 Einsatzzwecke von Prozessmodellen

Schwerpunkt Organisationsgestaltung	**Schwerpunkt Anwendungssystemgestaltung**
Organisationsdokumentation	Auswahl von Standardsoftware
Prozessorientierte Reorganisation	Modellbasiertes Customizing
Kontinuierliches Prozessmanagement	Softwareentwicklung
Simulation	Workflowmanagement
Personalbedarfsplanung	
Prozesskostenrechnung	
Benchmarking	
Wissensmanagement	
Zertifizierung	

Auf eine detaillierte Darstellung der Modellierungszwecke wird an dieser Stelle verzichtet, da im Folgenden die Frage im Vordergrund steht, wie der Detaillierungsgrad von Prozessmodellen geeignet ermittelt und beschrieben werden kann. In einem späteren Beitrag soll dann der Frage nachgegangen werden, welcher Detaillierungszweck für welche Modellierungsfragestellung am zweckmäßigsten ist.

5 Detaillierungsgrad von Prozessmodellen

5.1 Detaillierungsbegriff

GAITANIDES [Ga83, 77-83] wertet den Detaillierungsgrad von Prozessen bereits als eine der zentralen Problemstellungen bei der Prozessanalyse. Der Detaillierungsgrad bezieht sich hier auf die hierarchische Zerlegung von Prozessen in detaillierter beschriebene Teilprozesse, um so ein tieferes Verständnis des Prozessablaufs zu erhalten. Weitere mögliche Detaillierungen eines Prozesses, wie Informationsobjekte und Attribuierungen, sind in diesem Verständnis nicht enthalten. Synonym zum Detaillierungsgrad verwendet GAITANIDES auch den Begriff Auflösungsniveau sowie den Aggregationsgrad, um die der Detaillierung entgegengesetzte Dimension zu verdeutlichen.

SCHEER [Sc01, 24-25] unterscheidet nach der gewählten Detaillierung zwischen Funktionsbündeln, Funktionen, Teilfunktionen bis hin zu Elementarfunktionen. Dabei weisen Funktionsbündel keine Detaillierung und Elementarfunktionen die stärkste Detaillierung auf. Das Verständnis des Detaillierungsgrades ist hier ebenfalls hierarchischer Natur. ALLWEYER [Al05, 55-57] stellt den Detaillierungsgrad als eine zentrale Eigenschaft von Prozessen heraus. Er betrachtet die Aktivitäten, aus denen sich ein Prozess zusammensetzt. Diese können auf unterschiedlichen Detaillierungsebenen beschrieben werden, sodass sich eine Aktivität durch einen detaillierteren Prozess darstellen lässt. Das Ergebnis stellt eine Prozesshierarchie dar. ALLWEYER schließt bei seinem Verständnis des Detaillierungsgrades zusätzlich zur Verfeinerung von Funktionen auch die Verfeinerung weiterer in Modellen enthaltener Informationen mit ein.

ROSEMANN [RO08, 79-80] gehen auf den Detaillierungsgrad im Rahmen der Spezifikation von Modellierungskonventionen ein. Der Detaillierungsgrad wird hier als eine der Kategorien (neben Namenskonventionen, Darstellungsregeln, etc.) vorgestellt, die vor der Durchführung der Prozessmodellierung geregelt und festgehalten werden sollten. Mittels der Modellierungskonventionen kann so eine einheitlichere Modellierung gewährleistet werden. Was der Detaillierungsgrad bei ihrem Begriffsverständnis konkret umfasst, bleibt jedoch offen. Der Begriff des Detaillierungsgrads findet sich auch in weiteren Quellen zur Prozessmodellierung, etwa bei LEHMANN [Le08, 84-85], FISCHER ET AL. [FFO06, 9], GADATSCH [Ga10, 211-212], BRUGGER [Br05, 19] oder STAUD [St06, 231-232]. Wie oben beschrieben, ist dabei das Verständnis des Begriffs nicht einheitlich. Auch eine explizite Definition des Begriffes oder eine konkrete Beschreibung der möglichen zu detaillierenden Inhalte eines Prozessmodells findet sich nur selten.

Ausnahmen sind hier insbesondere ROSEMANN und KUGELER. ROSEMANN [RO96, 71-82] sieht den Detaillierungsgrad als zu messende Eigenschaft von Prozessmodellen, mit deren Hilfe eine einheitliche Modellierung gewährleistet werden kann, in dem Modelle auf hierarchischen Detaillierungsstufen mit einem festen Detaillierungsgrad erstellt werden. Synonym verwendet ROSEMANN [Ro96, 133] auch den Begriff *Prozesstiefe*, erweitert diese Definition jedoch kurz darauf, indem er die Prozesstiefe als lediglich einen Bestandteil des Detaillierungsgrades sieht. Der Detaillierungsgrad umfasst demnach zusätzlich die Prozessbreite, Prozesslänge, Informationsobjekte und die Intensität der Attribuierung. Die *Prozessbreite* beschreibt, in welchem Umfang die Gesamtheit der möglichen Systemzustände durch das vorliegende Modell abgebildet wird und steigt, je mehr Sonderfälle berücksichtigt werden. Die *Prozesslänge* ist umso größer, je umfangreicher das Prozessmodell, bei gleichbleibender Prozesstiefe und -breite, ist. Über den Prozessablauf hinaus, kann das Prozessmodell durch zusätzliche *Informationsobjekte* ergänzt werden. Dies können beispielsweise im Modell ergänzte Organisationseinheiten, Informationssysteme oder Ressourcen sein. *Attribute* können einen Prozess sowie die im Prozess enthaltenen Informationsobjekte näher spezifizieren. Nach dem Verständnis von KUGELER [KU00, 116-117] umfasst der Detaillierungsgrad die Detaillierungstiefe und die Detaillierungsbreite. Die *Detaillierungstiefe* zeigt an, wie granular die in einem Modell enthaltenen Informationsobjekte beschrieben werden. Die *Detaillierungsbreite* gibt an, wie vollständig ein Modell die Gesamtheit der in der Realität vorhandenen Elemente abbildet.

Aufbauend auf den Definitionen von ROSEMANN und KUGELER soll der Detaillierungsgrad hier wie folgt definiert werden: Der *Detaillierungsgrad* von Prozessmodellen beschreibt den Umfang, in dem Prozesse modelliert werden. Die Detaillierung ist in Abhängigkeit vom angestrebten Modellierungszweck auszuwählen und zu begrenzen. Die für die Detaillierungsproblematik zu entscheidenden Aspekte der Prozessmodellierung lassen sich unterscheiden in die Bereiche Prozessdarstellung, Informationsobjekte und Attribute. Als *adäquater Detaillierungsgrad* soll das für eine bestimmte Problemstellung zweckmäßige Verfeinerungsniveau eines Prozessmodells bezeichnet werden.

5.2 Einflussfaktoren auf den Detaillierungsgrad

Um den adäquaten Detaillierungsgrad von Modellen bestimmen zu können, muss untersucht werden, welche Faktoren den Detaillierungsgrad beeinflussen können und wie sie sich auf ihn auswirken. Einen ersten Hinweis liefern die Grundsätze ordnungsmäßiger Modellierung [BRS95, 435-445]. Die dort beschriebenen Grundsätze der Wirtschaftlichkeit und der Relevanz haben Auswirkung auf den Detaillierungsgrad. Damit ein Prozessmodell dem Grundsatz der Wirtschaftlichkeit genügt, darf seine Erstellung nicht mehr Aufwand verursachen als nötig und muss dabei in einem angemessenen Verhältnis zum Modellnutzen stehen. Um dem Grundsatz der Relevanz zu entsprechen, müssen Prozessmodelle die Bestandteile enthalten, die für die Erfüllung des ausgegebenen Einsatzzweckes benötigt werden. Es muss also der richtige Gegenstand auf dem richtigen Abstraktionsniveau modelliert sein. Die Relevanz kann daran festgemacht werden, dass der Nutzen des Modells sinken würde, wenn die betreffende Information weggelassen werden würde bzw. das Modell aufgrund fehlender Inhalte für den vorgesehenen Zweck unbrauchbar ist.

Ideal für die Prozessmodellierung wäre es, wenn das Modell alle für den Modellierungszweck relevanten Inhalte enthält und dabei gleichzeitig dem Grundsatz der Wirtschaftlichkeit gerecht werden kann. Der adäquate Detaillierungsgrad steht also häufig im Konflikt zwischen den Grundsätzen der Relevanz und der Wirtschaftlichkeit, so dass entschieden werden muss, wie ein adäquater Kompromiss zu erreichen ist. Ergänzend können eine ganze Reihe weiterer Einflussfaktoren ausgemacht werden. Diese lassen sich meist den beiden erwähnten GoM-Grundsätzen zuordnen. Wie in Kapitel 3 beschrieben, wurden auf Basis einer umfassenden Literaturrecherche die in Tabelle 2 dargestellten Einflussfaktoren ermittelt, die bei der Erarbeitung eines adäquaten Detaillierungsgrades relevant erscheinen. Es ist nicht auszuschließen, dass weitere Einflussfaktoren auf den Detaillierungsgrad existieren, so dass diese Liste zunächst lediglich einen nicht abschließenden Überblick gibt.

Tab. 2 Beispiele für Einflussfaktoren mit Bezug zu den Grundsätzen der Wirtschaftlichkeit und Relevanz

Einflussfaktoren: Wirtschaftlichkeit	**Auswirkung auf den Detaillierungsgrad**
Flexibilität	Mit steigenden Anforderungen an die Modellflexibilität sinkt der adäquate Detaillierungsgrad, damit Prozessmodelle sich schneller und unter geringem Auswand anpassen lassen
Dynamik, Änderungshäufigkeit	Je dynamischer und häufiger sich Prozesse verändern können, desto niedriger ist der adäquate Detaillierungsgrad, da detaillierte Modelle mehr Anpassungsaufwand erfordern als weniger detaillierte Modelle.
Strukturiertheit	Unstrukturierte Prozesse erfordern mehr Aufwand bei der Modellierung als strukturierte Prozesse. Die adäquate Detaillierung ist bei unstrukturierten Prozessen tendenziell geringer, um den Modellierungsaufwand zu begrenzen
Ausführungshäufigkeit	Der adäquate Detaillierungsgrad bei der Modellierung von Prozessen mit geringer Ausführungshäufigkeit ist tendenziell niedriger als bei Prozessen, die häufiger ausgeführt werden. Der Modellierungsaufwand steht hier einem geringeren Nutzen gegenüber.
Einflussfaktoren: Relevanz	**Auswirkung auf den Detaillierungsgrad**
Automatisierbarkeit	Soll eine Automatisierung von Prozessen erfolgen, sind ausführliche Modelldetaillierungen nur für die Prozesse nötig, die auch automatisiert werden können.
Wettbewerbsrelevanz	Prozesse, die für das Unternehmen strategisch wichtig sind, müssen tendenziell genauer modelliert werden.
Wissens- und Datenintensität	Ist ein Prozess besonders von implizitem Wissen abhängig, ist die Detaillierung schwieriger, da implizites Wissen sich nur begrenzt explizieren lässt und somit der Detailgrad eingeschränkt wird [Al09, 65].
Domäne	Abhängig von Eigenschaften der Domäne (beispielsweise ein sicherheitskritisches Geschäftsfeld), wo der zu modellierende Prozess angesiedelt ist, wird eine detailliertere oder wenig detaillierte Modellierung erforderlich [SS08, 189].
Perspektiven	Es sollten in einem Prozessmodell nur die Details enthalten sein, die für die jeweilige Perspektive (können sich zum Beispiel in Abhängigkeit vom Modellierungszweck oder vom Nutzer des Modells unterscheiden) relevant sind. Zusätzliche Details erhöhen Aufwand und Komplexität [BDK02, 25-43]

5.3 Ein mögliches Bewertungsschema für den Detaillierungsgrad

Um eine möglichst objektive Aussage über die adäquate Detaillierung von Prozessmodellen bei gegebenen Modellierungszwecken treffen zu können, wird im Folgenden ein Bewertungsschema für den Detaillierungsgrad erarbeitet. Dieses soll für jeden oben genannten Aspekt der Detaillierung eine Einordnung und Empfehlung ermöglichen. Durch die Nutzung eines einheitlichen Schemas können außerdem später Zusammenhänge und Unterschiede in den Anforderungen an den Detaillierungsgrad zwischen verschiedenen Modellierungszwecken ersichtlich werden.

Der Detaillierungsaspekt der *Prozesstiefe* lässt sich hinsichtlich der verschiedenen Hierarchieebenen differenzieren, auf der ein Prozessmodell modelliert werden kann. Bezüglich der Unterteilung der verschiedenen Ebenen existieren in der Literatur Vorschläge, die meistens zwischen vier bis sechs Ebenen unterscheiden. Die Prozesstiefe soll in dieser Arbeit auf einer fünfstufigen Skala bewertet werden, wobei jeweils eine Stufe für eine Hierarchieebene steht. Die Unterteilung der Prozesstiefe richtet sich nach dem Hierarchisierungskonzept bei HÜSSELMANN [Hü06, 125-126] und unterscheidet die Stufen 1. Hauptprozesse, 2. Geschäftsprozesse, 3. Arbeitsvorgänge, 4. Arbeitsschritte und 5. Elementartätigkeiten. Abbildung 2 zeigt die verschiedenen Stufen und die ihnen zugeordneten Ebenen und Bezeichnungen am Beispiel des Hauptprozesses Auftragsabwicklung.

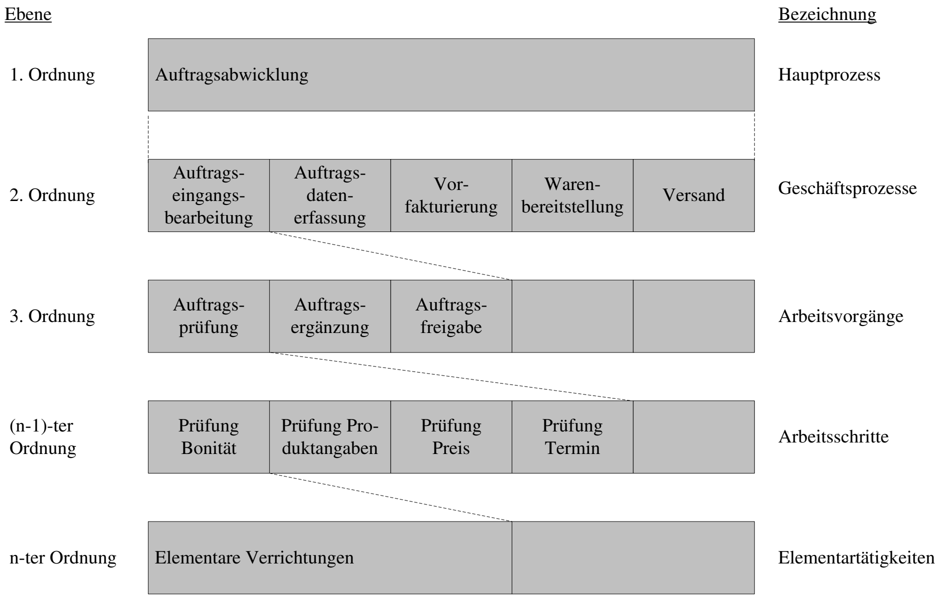


Abb. 2: Auflösung von Prozessen nach Prozesstiefe am Beispiel „Auftragsabwicklung“
(In Anlehnung an [Ga83, 80]; [Hü06, 125-126])

Für die Bewertung der *Prozessbreite* existieren in der Literatur hingegen keine Vorschläge zur Abstufung des Detaillierungsgrades. Es liegt nahe, die Unterteilung der Prozessbreite einerseits von der Durchführungshäufigkeit der jeweiligen Prozessalternative sowie andererseits von der Anzahl theoretisch möglicher Ablaufvarianten insgesamt abhängig zu machen. Durch die sinnvolle Kombination dieser Kriterien ergibt sich eine vierstufige Differenzierung für die Modellierung von Prozessen in die Breite (siehe Tabelle 3). Der Umfang der Prozessbreite nimmt von Stufe eins bis zu Stufe vier zu.

Die erste Stufe bildet der „Optimalfall" des Prozessdurchlaufs oder auch „Happy Path" [FRH10, 131]. Auf dieser Stufe werden nur diejenigen alternativen Prozesswege erfasst, die zur Bewältigung des Optimalfalls benötigt werden. Die Durchführungshäufigkeit des Optimalfalls sollte in der Regel höher sein als die der übrigen Ablaufalternativen. Auf der zweiten Stufe kommen alle Ablaufvarianten hinzu, die zusammengefasst den höchsten Anteil an der Gesamtmenge der Prozessdurchläufe ausmachen. Die Anteilsgrenze soll, in Anlehnung an die ABC-Analyse, bei etwa 75% der Fälle festgelegt werden. Es handelt sich somit um vergleichsweise wenige Varianten, die jedoch eine überproportional große Anzahl an Fällen ausmachen. Die Varianten, die mit dieser Stufe umfasst werden, sollen im weiteren Verlauf auch als „Standardfälle" bezeichnet werden. Auf der dritten Stufe der Prozessbreite werden alle Varianten modelliert, die zusammen für 95% der Fallzahlen verantwortlich sind. Diese Stufe soll als „Umfangreich" bezeichnet werden. Als „Vollständig" wird schließlich die letzte Stufe bezeichnet, die bis zu 100% der Fälle und somit sämtliche Ablaufalternativen einschließt.

Tab. 3 Differenzierung der Prozessbreite von Prozessmodellen

Stufe	Bezeichnung	Berücksichtigte Ablaufvarianten	Durchführungshäufigkeit
1	Happy Path	Optimalfall	Etwa 50% der Fälle
2	Standardfälle	Wenige	Bis etwa 75% der Fälle
3	Umfangreich	Viele	Bis etwa 95% der Fälle
4	Vollständig	Alle	Bis 100% der Fälle

Hinsichtlich des Detaillierungsaspektes der *Informationsobjekte* ist eine Skalierung nicht ratsam, da hier Objekte betrachtet werden, die nicht aufeinander aufbauen. Stattdessen soll eine einfache Auflistung der für die Prozessmodelle zum jeweiligen Modellierungszweck relevanten Informationsobjekte genügen. Hieraus wird bereits hinreichend ersichtlich, welche Modellierungszwecke welche Informationsobjekte benötigen und somit eventuell Parallelen zu anderen Modellzwecken aufweisen. Auch bezüglich des Detaillierungsaspektes der *Attribute* wird aus gleichem Grund auf eine Auflistung der jeweils zweckrelevanten Attribute zurückgegriffen.

Die Ausprägungen (später: Empfehlungen) zum Detaillierungsgrad lassen sich pro Modellierungszweck nun tabellarisch zusammenfassen. Abbildung 3 zeigt exemplarisch den Aufbau der Tabelle. Die einzelnen Aspekte des Detaillierungsgrades sind zur besseren Übersicht farblich voneinander abgegrenzt. Bei der Prozesstiefe und der Prozessbreite kennzeichnen die Kästen mit einer dunklen Farbgebung, welche Stufe mindestens für den betreffenden Modellierungszweck zu wählen ist. Die hellere Farbe signalisiert entsprechend, bis zu welcher Stufe die Detaillierung maximal ratsam ist. Im Beispiel ist also die Prozesstiefe mindestens auf Stufe 3 und somit auf der Ebene der Arbeitsvorgänge zu detaillieren, maximal ist eine Detaillierung sogar bis auf Stufe 5 und somit auf der Ebene der Elementartätigkeiten sinnvoll. In der Spalte Informationsobjekte findet sich in den Kästen aus Platzgründen die Kurzbezeichnung der im Modell zu detaillierenden Informationsobjekte. Die Bezeichnung ORG im Beispiel bezeichnet Organisationseinheiten. In der Attributspalte stehen wiederum stellvertretende Bezeichner für Attributkategorien (hier beispielhaft Kennzahlen), die mehrere Attributfelder und ihre Ausprägungen zusammenfassen. Die Attribute und Informationsobjekte können, wie im Beispiel ersichtlich, auch eingeklammert dargestellt werden, um zu kennzeichnen, dass die Modellierung dieser Details als optional bewertet wird.

Zwecke	Prozesstiefe					Prozessbreite				Informationsobjekte				Attribute
	1	2	3	4	5	1	2	3	4					
Modellzweck 1										ORG				(Kennzahlen)

Abb. 3: Exemplarische Ergebnistabelle

6 Ausblick

Auf Basis der in diesem Beitrag gelegten Grundlagen sollen im nächsten Schritt für unterschiedliche Modellierungszwecke die Anforderungen an den Detaillierungsgrad strukturiert ermittelt und beschrieben werden. Zunächst sind dazu die sich aus dem Modellierungszweck ergebenden allgemeinen fachlichen Anforderungen an die Modelldetaillierung zu ermitteln. Aus diesen werden anschließend die Anforderungen an die Detaillierungsaspekte Prozessbreite, Prozesstiefe, Informationsobjekte und Attribute abgeleitet und in das Bewertungsschema überführt.

Anschließend können dann auch Rückschlüsse gezogen werden, welche Modellierungszwecke enge Beziehungen mit Blick auf den richtigen Detaillierungsgrad aufweisen. Daraus würde deutlich, inwiefern es möglich ist, einzelne Modellierungszwecke miteinander zu kombinieren, um so Prozessmodelle besser wiederverwenden zu können und den Modellierungsaufwand insgesamt zu reduzieren.

Literaturverzeichnis

[Al05] Allweyer, T.: Geschäftsprozessmanagement – Strategie, Entwurf, Implementierung, Controlling. W3L, Herdecke, 2005.

[Al07] Allweyer, T.: Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen. In (Nüttgens, M.; Rump, F. J.; Gadatsch, A., Hrsg.): Proceedings der EPK 2007. GI e.V., Bonn, 2007, S. 23-38.

[Al09] Allweyer, T.: BPMN 2.0 – Einführung in den Standard für die Geschäftsprozessmodellierung. 2. Aufl., Books on Demand, Norderstedt, 2009.
[BDK02] Becker, J.; Delfmann, P.; Knackstedt, R.; Kuropka, D.: Konfigurative Referenzmodellierung. In (Becker, J.; Knackstedt, R., Hrsg.) Wissensmanagement mit Referenzmodellen. Physica, Heidelberg, 2002, S. 25-43.
[BKR08] Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.): Prozessmanagement – Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 6. Aufl. Springer, Berlin, 2008.
[BMW09] Becker, J.; Mathas, C.; Winkelmann, A.: Geschäftsprozessmanagement. Springer, Berlin, 2009.
[BRS95] Becker, J.; Rosemann, M.; Schütte, R.: Grundsätze ordnungsmäßiger Modellierung. Wirtschaftsinformatik, 37 (5), 1995, S. 435-445.
[Br05] Brugger, R.: IT-Projekte strukturiert realisieren. 2. Aufl. Vieweg, Wiesbaden, 2005.
[Da05] Davis, R.: Business Process Modelling with ARIS: A Practical Guide. Springer, London, 2005.
[Fe06] Fettke, P.: State-of-the-Art des State-of-the-Art. Wirtschaftsinformatik, 48 (5), 2006, S. 257–266.
[FFO06] Fischer, H.; Fleischmann, A.; Obermeier, S.: Geschäftsprozesse realisieren – ein praxisorientierter Leitfaden von der Strategie bis zur Implementierung. Vieweg, Wiesbaden, 2006.
[FRH10] Freund, J.; Rücker, B.; Henninger, T.: Praxishandbuch BPMN. Hanser, München, 2010.
[Ga10] Gadatsch, A.: Grundkurs Geschäftsprozess-Management : Methoden und Werkzeuge für die IT-Praxis: eine Einführung für Studenten und Praktiker. 6., aktualisierte Aufl., Vieweg+Teubner Verlag / GWV Fachverlage, Wiesbaden, 2010.
[Ga83] Gaitanides, M.: Prozeßorganisation – Entwicklung, Ansätze und Programme prozeßorientierter Organisationsgestaltung. Vahlen, München, 1983.
[Hü06] Hüsselmann, C.: Personalbedarfsplanung als Produkt werkzeuggestützter Geschäftsprozessoptimierung – Stellenbewertung und -bemessung am Beispiel Public Sector. In (Kruppke, H.; Otto, M.; Gontard, M., Hrsg.): Human Capital Management: Personalprozesse erfolgreich managen. Springer, Berlin, 2006, S. 115-140.
[IRR09] Indulska, M.; Recker, J. C.; Rosemann, M.; Green, P.: Business Process Modelling : Current Issues and Future Challenges. In (van Eck, P.; Gordijn, J.; Wieringa, R., Hrsg.) Proc. 21st Int. Conf. on Advanced Information Systems - CAiSE. Springer, Amsterdam, 2009, S. 501-514.
[Ku00] Kugeler, M.: Informationsmodellbasierte Organisationsgestaltung – Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation. Dissertation, Westfälische-Wilhelms-Universität. Münster, 2000.
[Le08] Lehmann, F. R.: Integrierte Prozessmodellierung mit ARIS. dpunkt, Heidelberg, 2008.
[Ro96] Rosemann, M.: Komplexitätsmanagement in Prozessmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Dissertation, Westfälische-Wilhelms-Universität. Münster, 1996.
[Ro08] Rosemann, M.: Vorbereitung der Prozessmodellierung. In (Becker, J.; Kugeler, M., Rosemann, M., Hrsg.) Prozessmanagement – Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 6. Aufl. Springer, Berlin, 2008, S. 45-104.
[Sc01] Scheer, A. W.: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4. Aufl. Springer, Berlin , 2001.
[SS08] Speck, M.; Schnetgöke, N.: Sollmodellierung und Prozessoptimierung. In (Becker, J.; Kugeler, M.; Rosemann, M., Hrsg.): Prozessmanagement – Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 6. Aufl. Springer, Berlin, 2008, S. 185-220.
[St06] Staud, J.: Geschäftsprozessanalyse. 3. Aufl. Springer, Berlin, 2006.
[vA11] van der Aalst, W. M. P.: Process Mining – Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg, 2011.

Optional Activities in Process Flows

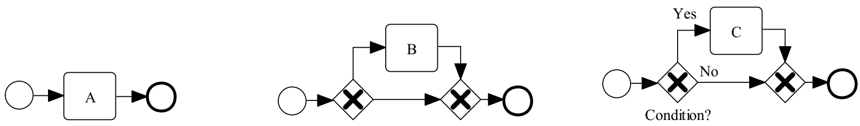
Christine Natschläger, Verena Geist, Felix Kossak, Bernhard Freudenthaler

Software Competence Center Hagenberg GmbH
Softwarepark 21
A-4232 Hagenberg
{firstname.secondname}@scch.at

Abstract: A drawback of many business process modelling languages (BPMLs) is that modalities are implicitly expressed through the structure of the process flow. All activities are implicitly mandatory and whenever something should be optional, the process flow is split to offer the possibility to execute the activity or to do nothing. This implies that the decision whether to execute one or more activities is described within another element, such as a gateway. The separation of decision and execution requires additional modelling elements and a comprehensive understanding of the entire process to identify mandatory and optional activities. In this paper, we address the problem and present an approach to highlight optionality in BPMLs based on the Control-Flow Patterns. Furthermore, we study the semantics of explicitly optional activities and show how to apply the general approach to a concrete BPML like the *Business Process Model and Notation* (BPMN). An explicitly optional activity is deemed to be more intuitive and also positively affects the structural complexity and the understandability of the process flow as shown by a graph transformation approach and a case study.

1 Introduction

Many business process modelling languages (BPMLs) provide a solely implicit expression of modality through the structure of the process flow. All activities are implicitly mandatory and whenever something should be optional, the process flow is split to offer the possibility to execute the activity or to do nothing. This implies that the decision whether to execute an activity is described within another element, e.g. a gateway. The separation of decision and execution requires additional modelling elements to split and merge the process flow and a comprehensive understanding of the entire process to identify mandatory and optional activities. For example, three process flows are shown in Fig. 1. In the first case, activity A must be executed in every process instance and is, thus, mandatory (see Fig. 1(a)). In the second case, the process flow is split to express that activity B is optional (see Fig. 1(b)). In addition, the modality also depends on the type of the split (user vs. conditional choice, parallel/exclusive/inclusive split) and on alternative elements. For example, the process flow shown in Fig. 1(c) has the same structure but comprises a conditional choice. In this case, activity C is mandatory if the condition evaluates to true (conditional obligation).

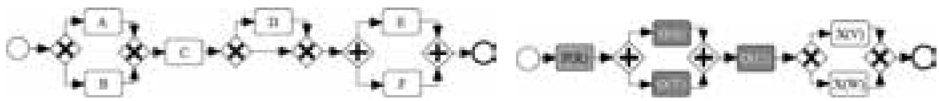


(a) Mandatory Activity (b) Optional Activity (c) Conditional Obligation

Figure 1: Implicit Expression of Modality through Process Flow Structure

This paper is also motivated by previous work (see [Nat11], [Nat12]), in which deontic logic was used to highlight modalities and applied to the *Business Process Model and Notation* (BPMN). The understandability of the approach was studied within a preliminary survey, which was answered by 22 post-graduate computer scientists. Although the number of respondents is too small for a significant survey, the preliminary survey provides some interesting results and identifies possible problems.

One representative example of the survey is shown in Fig. 2 and the explicit expression of modality is based on the deontic concepts of permission P (optional), obligation O (mandatory), and alternative X. The respondents then answered six questions for each model type as, for example, which tasks are mandatory, which tasks are optional, or what happens if task D (or R) cannot be executed. Note that in order to avoid a recognition of models expressing the same example, corresponding tasks received different names and the order of examples, elements, and questions varies.



(a) Implicit Expression of Modality (b) Explicit Expression of Modality

Figure 2: Preliminary Survey: Example

In summary, 64 mistakes emerged in the implicit and 15 mistakes in the explicit expression of modality, so the number of mistakes could be reduced by 77%. Considering the entire survey, further mistakes emerged in the deontic extension, because other examples comprised complex deontic constructs like conditional commitments and multiple deontic classifications, but still the overall number of mistakes could be reduced by 14%. In addition, all respondents were asked whether they prefer the implicit or explicit expression of modality. In most examples, the explicit expression was preferred, especially for more complex process flows. So the preliminary survey allows us to assume that a highlighting of modalities and a reduction of structural complexity increases the understandability of a process flow.

The following section provides an overview of related work and presents the expression of modality in other BPMLs like *UML Activity Diagrams* (UML ADs), *Event-Driven Process Chains* (EPCs), or *Yet Another Workflow Language* (YAWL). Subsequently, Section 3 identifies optional activities in BPMLs based on the *Control-Flow Patterns*, provides a description of the semantics based on *Abstract State Machines* (ASMs), and extends BPMN with optional activities. The approach for optional activities is then evaluated in Section 4 by a graph transformation that shows a reduction of structural complexity and a case study taken from an industrial project. This papers concludes with a summary of the results and a description of future goals.

2 Related Work

There are several scientific investigations that highlight the importance and need for distinguishing between optional and mandatory behaviour in BPMLs. For example, in their reference model, Reinhartz-Berger et al. introduce stereotypes for constraining the multiplicity of elements within activity diagrams [RBSS05]. Optional activities can be denoted as ≪0..n≫ or ≪0..1≫, indicating that the activity can be performed any number of times or at most one time respectively. In [MA02], Muthig and Atkinson introduce a model-driven approach for product line engineering and define optional activities in a platform-independent way for capturing variabilities. They therefore apply the ”VariantAssetElement” pattern and represent an optional activity by setting the element's fill colour to grey. Kïnzle and Reichert describe a data-driven approach for object-aware process management [KR11] that allows for realising optional activities, which enables authorised users to access object instances asynchronously to the normal control flow, i.e. to optionally execute activities at any point in time.

However, common languages and notations, such as BPMN, UML ADs, EPCs, or YAWL, do not provide means to explicitly denote optional and mandatory activities. For this reason, Kïnzle and Reichert suggest to add conditional branches at various positions within a process model as a workaround for supporting optional activities for process-oriented approaches [KR09]. Chang and Kim [CK07] also recommend this approach to realise optional activities for defining workflows in BPMLs in order to develop adaptable services. In the same way, Oestereich advises to express optional activities in UML ADs through the structure of the process flow [Oes02]. So all approaches suggest to express modality through the structure of the process flow. As a result, users have difficulties with comprehending the resulting bulky and spaghetti-like process models as well as distinguishing between optional and mandatory activities [KR09].

Besides some specialised versions of EPCs and YAWL, e.g. C-EPCs [Ros09] or C-YAWL [tHvdAAR10], that support optionality, there also exist several recommendations and extensions for handling optional activities in UML and BPMN. Razavian and Khosravi present a method for modelling variability in business process models based on UML ADs [RK08]. They propose modelling solutions for optional activities that are classified based on the origins of variability, i.e. control flow, data flow, and actions, and realised by multiplicities and stereotypes respectively for business process elements. In [PSWW05], Puhlmann et al. give an overview of variability mechanisms in product family engineering. They suggest to realise optional activities in UML ADs using variation points represented by “Null-Activities”, i.e. a variant being an optional activity can be bound to a variation point using a dependency relation, both having assigned respective stereotypes.

Schnieders and Puhlmann [SP06] adapt the stereotype concept of UML to BPMN in order to address variability mechanisms for e-business processes families. They propose a ≪Null≫ variation point that can be resolved with specific variants to represent optional behaviour. In case that the variation point has only one optional resolution, the activity can be marked with an ≪Optional≫ stereotype and directly placed within the main process flow. A further approach for realising optional activities in BPMN suggested in [Mic10] is to apply ad-hoc sub-processes which leave the sequence and number of performances for a defined set of activities up to the performers (cf. [OMG11]). However, in that case more logic than just the concept of the completion condition is needed (see also [Sil09]). Using text annotations for displaying additional information about process elements, also suggested in [Mic10], has the drawback that annotations marking variabilities cannot be distinguished from other annotations in a business process diagram [PSWW05].

3 Optional Activities

In order to address the problems noted above, we propose to explicitly introduce optional activities in business process modelling. By optional activities we mean activities which a user can choose at runtime to either perform or not, without affecting control flow. Such activities should be graphically marked, either by a dashed line or by colour.

Activities are implicitly optional if they are, for example, defined after an exclusive choice or a multi-choice and if there is at least one alternative *Phi*-Path (empty path). With explicitly optional activities, the *Phi*-Paths and the respective choices can be removed, since the choice is directly expressed within the activity.

We first consider the Workflow Patterns introduced by research groups around Wil van der Aalst and Arthur ter Hofstede (see e.g. [vdAtHKB03], [vdAtH11]), because they are independent of concrete BPMLs and, thus, a solution for Workflow Patterns can be easily adapted for most concrete BPMLs, including BPMN, UML ADs, YAWL, and EPCs.

Next we compare the semantics of diagrams with conventional activities only and diagrams containing optional activities. Then we apply our concept to BPMN.

3.1 Control-Flow Patterns

In this section the Control-Flow Patterns are studied and optional activities are identified. An overview of all Control-Flow Patterns that express modality is given in Tab. 1. The first column provides the pattern number and the second column the pattern name. Only the exclusive and multi-choice pattern may express optionality and are, thus, studied in more detail in the following.

Table 1: Control-Flow Patterns with Modalities based on [vdAtH11]

No.	Pattern Name
Basic Control-Flow Patterns	
1	Sequence
2	Parallel Split
4	Exclusive Choice
Advanced Branching and Synchronization Patterns	
6	Multi-Choice
State-based Patterns	
16	Deferred Choice
18	Milestone
Iteration Patterns	
10	Arbitrary Cycles
21	Structured Loop
Trigger Patterns	
23	Transient Trigger
24	Persistent Trigger

3.1.1 *Exclusive Choice* Pattern

The *Exclusive Choice* pattern describes the divergence of one branch into two or more branches, but the thread of control is only passed to exactly one outgoing branch as shown in Fig. 3. The selection of the outgoing branch is based on an evaluation of the conditions defined for the outgoing paths [vdAtH11].

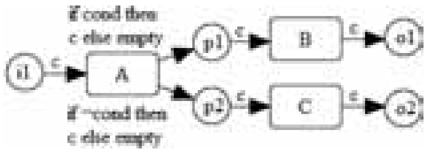


Figure 3: *Exclusive Choice* Pattern (Source: [vdAtH11])

The modality depends on the type of choice (user vs. conditional) and on a possible *Phi*-Path (i.e. an empty path that directly connects the split with the corresponding merge). If the exclusive choice provides a user decision with one or more *Phi*-Paths, then these *Phi*-Paths can be removed and all other tasks are classified as optional (marked with a dashed line) as shown in Fig. 4. If there is only one other path apart from the *Phi*-Path, then also the exclusive choice (split and merge) can be removed.

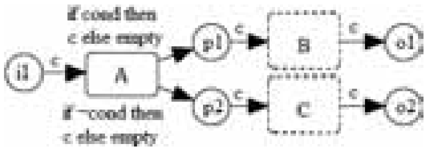
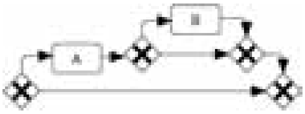
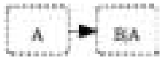


Figure 4: *Exclusive Choice* Pattern: Optional Activities

Conditional commitments (i.e. preconditions) are necessary for nested choices as shown by the BPMN example in Fig. 5(a) with the modality highlighted in Fig. 5(b). The precondition defines that task B may only be executed if task A was executed before.



(a) Example: BPMN



(b) Example: Optional Activities

Figure 5: Nested Choice with Conditional Commitment

3.1.2 *Multi-Choice* Pattern

The *Multi-Choice* pattern describes the splitting of one branch into two or more branches, where the thread of control is passed to one or more branches as shown in Fig. 6. The selection of the outgoing branches is again based on an evaluation of the conditions defined for the outgoing paths [vdAtH11].

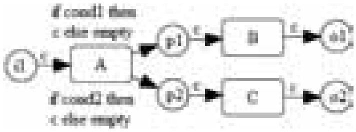


Figure 6: *Multi-Choice* Pattern (Source: [vdAtH11])

Similar to the exclusive choice pattern, the modality depends on the type of choice (user vs. conditional) and on a possible *Phi*-Path. If the multi-choice provides a user decision with one or more *Phi*-Paths, then these *Phi*-Paths can be removed and all other tasks are classified as optional (marked with a dashed line) as shown in Fig. 7. If there is only one other path apart from the *Phi*-Path, then also the multi-choice can be removed.

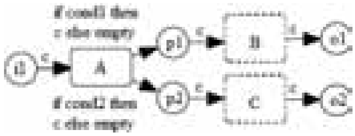


Figure 7: *Multi-Choice* Pattern: Optional Activities

3.2 Comparing the Semantics of Business Process Diagrams with and without Optional Activities

We now show that the introduction of optional activities, with the potential omission of certain choices (or gateways), is possible without altering the semantics of diagrams.

We show that a standard process diagram d on the one hand and a diagram d' with (explicitly) optional activities, which is derived from d by a respective transformation, on the other hand have the same semantics in the following sense:

1. Whenever a particular activity a would have been performed in d, also the corresponding optional activity a' in d' is performed and
2. A partial order with respect to the performance of activities in time is preserved, i.e., if it is relevant that activity a_1 is performed before a_2 in d, then also a'_1 is performed before a'_2 in d'.

We define the semantics with a token-based model. In conventional diagrams, an activity will be performed if it has been reached by a token and if certain other conditions are met (such as the availability of data or resources). We can express this algorithmically in the following way (using an *Abstract State Machine* (ASM) notation, see e.g. [BS03]):

```
WorkflowTransition(activity) =
  if controlFlowCondition(activity) and
        performanceCondition(activity) then
    ConsumeTokens(activity)
    PerformanceOperation(activity)
    wait until completed(activity) then
      ProduceTokens(activity)
```

where the `controlFlowCondition` signifies whether a token has reached the given activity and `performanceCondition` summarises all the other conditions for the activity to be able to perform (this simplification is sufficient for the given context of this paper). The three actions – `ConsumeTokens`, `PerformanceOperation`, and `ProduceTokens` – can be started in parallel (note that `ProduceTokens` is appropriately guarded; also note that we have omitted issues regarding different process instances for the sake of simplicity).

The semantics of an exclusive choice with two outgoing paths can be derived from `WorkflowTransition(activity)` by defining `ProduceTokens(activity)` as follows:

```
ProduceTokens(exclusiveChoice) =
  if choiceCondition(exclusiveChoice) then
    ProduceTokensAt(outgoingPlaces(exclusiveChoice)[0])
  else
    ProduceTokensAt(outgoingPlaces(exclusiveChoice)[1])
```

where `outgoingPlaces` is a list of places directly connected to the exclusive choice at the outgoing paths, out of which the first or, respectively, the second place is chosen. (We have renamed the argument to make it clear that this definition holds for exclusive choices only.)

The semantics of a respective exclusive merge (or ”simple merge”) can be derived by setting

```
controlFlowCondition(exclusiveMerge) =
  forsome (p in incomingPlaces(exclusiveMerge)) :
      enabled(p)
```

where `incomingPlaces` denotes the set of places relevant for the transition – i.e. (at least) one of the places before the choice must have the required number of tokens.

We now look at a subdiagram d with one incoming and one outgoing path, consisting of an exclusive choice c and an exclusive merge, joined by two paths, one of which contains one activity a, while the other path is empty (a *Phi*-Path).

The corresponding subdiagram d' derived from d by using (explicitly) optional activities will consist of a single optional activity a', with neither a choice (split) nor a merge. The semantics of such an optional activity can be modelled as follows:

```
WorkflowTransition(optionalActivity) =
  if controlFlowCondition(optionalActivity) and
        performanceCondition(optionalActivity) then
    ConsumeTokens(optionalActivity)
    if choiceCondition(optionalActivity) then
      PerformanceOperation(optionalActivity)
    if choiceCondition(optionalActivity) then
      wait until completed(optionalActivity) then
        ProduceTokens(optionalActivity)
    else
      ProduceTokens(optionalActivity)
```

Note that the definition of the semantics of a conventional activity as given further above can be derived from this definition for an optional activity by simply setting `choiceCondition(optionalActivity) := true`.

We assert (by design) that in d', `choiceCondition(a')` is identical to `choiceCondition(c)` in d (where c is the exclusive choice). Then it is easy to see that whenever a is performed in d, also a' will be performed in d'.

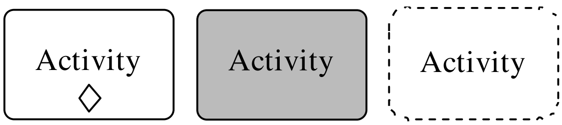
Also the control flow after the subdiagram remains unchanged, because if a' actually performs (i.e. `choiceCondition(a') = true`), then it sends a token (or tokens) on, just like a (in the latter case via the exclusive merge), and if a' does not perform, then it still sends on a token (or tokens), just like the *Phi*-Path in d would have done (again via the exclusive merge). Furthermore, if we assume that time for the splitting and merging activities of the exclusive choice and the exclusive merge can be neglected (in contrast to the performance of the main activity), then also the temporal performance of the subdiagram is unchanged. Therefore, also the sequence of activities in the wider context (i.e. including before and after the subdiagram) remains unchanged.

Within the given space, we can show the equivalence of the semantics only for such a simple case as that of d and d' discussed above. However, we think that the reader can already guess that this will also hold for more complex cases (as we have already checked for all relevant cases in yet unpublished work).

3.3 Optional Activities in BPMN

A BPMN activity can be either a *Task*, a *Sub-Process*, or a *Call Activity*. Although there is no formal basis for optional activities in BPMN (see also Section 2), process modellers should have the possibility to make use of optional activities to enhance intelligibility of their business process diagrams. This can be realised by introducing an explicit graphical representation of an optional activity, which must conform to the BPMN specification [OMG11]. According to this specification, an extended BPMN diagram may comprise new markers or indicators as well as new shapes representing a kind of artefact. In addition, graphical elements may be coloured where the colouring may have a specified semantics, or the line style of a graphical element may be changed if this change does not conflict with any other element.

Considering optional activities, we suggest the three representations shown in Fig. 8. In Fig. 8(a) an activity is marked as optional by an additional marker that corresponds to the modal operator of possibility ($\Diamond$). The advantage of this representation is that markers are a well-known concept in BPMN and easy to understand; however, since activities may comprise several markers, a further marker can result in an overcrowded representation. Thus, another possibility is to define a different background colour for an optional activity, e.g. green (see Fig. 8(b)). The advantage of this representation is that it is supported by most modelling tools; however, it requires diagrams to be coloured. The last suggestion is shown in Fig. 8(c) and uses a dashed line to highlight an optional activity. This representation is easy to understand and can be represented by several modelling tools. Note that the dashed line style does not conflict with the dotted line style of an event sub-process.



(a) Marker (b) Colour (c) Line Style

Figure 8: Possible Representations of Optional Activities in BPMN

4 Evaluation

Finally, we evaluate our approach for optional activities by a graph transformation that shows a reduction of structural complexity and a case study taken from an industrial project. The case study comprises a workflow implemented in an industrial project for the prevention domain of the *Austrian Social Insurance Company for Occupational Risks* (AUVA). Due to the fact that the original order execution process is too complex for an example in this contribution, we combined several details to sub-processes to better demonstrate the reduction of modelling elements and increased comprehensibility of diagrams. Furthermore, pools and lanes are not considered in this example. An extract of the order execution process and its transformation to an explicit representation of optionality is presented in [Nat11].

4.1 Graph Transformation

In order to study the reduction of structural complexity, we defined a graph transformation system (GTS) and specified several transformation rules to transform models with an implicit expression of modality to models with an explicit expression based on deontic logic. By adding only one binary attribute (optionality), the number of gateways and sequence flows can be reduced. Every transformation rule further highlights the original and the resulting number of gateways and sequence flows in an element called *MeasuredValues*. The GTS currently uses BPMN as a specific BPML to represent exclusive and multi-choices and is limited to structured diagrams with one task per path following a splitting gateway. In the following, only the transformation rules for exclusive choices are presented, since the multi-choice transformation rules are very similar.

ExclusiveWithPhiDualRule: The rule *ExclusiveWithPhiDualRule* takes an exclusive gateway with a task and a *Phi*-Path and transforms it to a permissible task (see Fig. 9). A negative application condition (NAC) forbids further alternative nodes. The transformation leads to a reduction of two gateways and three sequence flows.

(a) LHS (b) RHS

Figure 9: ExclusiveWithPhiDualRule

ExclusiveWithPhiRule: The next rule is called *ExclusiveWithPhiRule* and transforms a task of an exclusive gateway with a *Phi*-Path into a permissible task (see Fig. 10). The rule can be applied several times to classify an arbitrary number of tasks. A positive application condition (PAC) specifies that a further node exists, since otherwise the rule *ExclusiveWithPhiDualRule* should be applied. This transformation rule does not reduce the number of gateways and sequence flows.

(a) LHS (b) RHS

Figure 10: ExclusiveWithPhiRule

ExclusiveWithPhiRuleFinish: The rule *ExclusiveWithPhiRuleFinish* will be applied after the rule *ExclusiveWithPhiRule* and removes the *Phi*-Path as shown in Fig. 11. The left-hand side of the rule requires two permissible tasks as well as a *Phi*-Path, and a NAC forbids further not transformed tasks. The transformation leads to a reduction of one sequence flow.

(a) LHS (b) RHS

Figure 11: ExclusiveWithPhiRuleFinish

Since every transformation rule leads to equally many or fewer gateways and sequence flows, the structural complexity of the explicit representation is either the same or lower. The graph transformation system is described in more detail in [Nat11].

4.2 Case Study

In order to study the suitability of the proposed extension for the prevention domain of the AUVA, we provide a case study with a workflow taken from a business environment. The goal of the prevention domain is to suggest possibilities to improve the employees' safety and health conditions at their workplaces. For this purpose, the prevention workers receive requests to visit companies. Thus, one major workflow describes the process of an order execution. Since this workflow is typical for many companies, it is taken as a basis for this case study. The BPMN diagram of the order execution process is shown in Fig. 12. The process comprises the following activities: *Create Request* (CR), *Approve Order* (AO), *Appointment Management* (AM), *Order in Progress* (OP), *Execute Order* (EO), *Report Management* (RM), and *Close Order* (CO). The sub-processes *AM* and *RM* comprise further details.

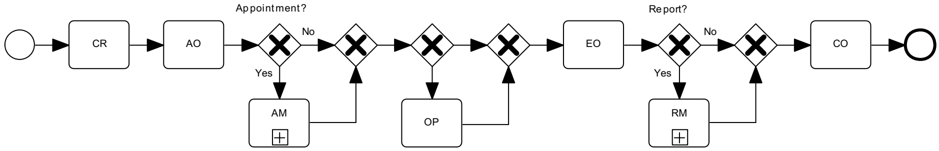


Figure 12: Order Execution Process: Implicit Expression

In a first step, the order execution process defines that a request is created (different roles are disregarded in this example). If a request was created, then this request must be approved resulting in an order. Afterwards, an appointment can be defined and it can be specified that the order is in progress. Subsequently, every order must be executed and reports can be created and modified. Finally, the order is closed. Although it is possible to discuss some aspects of the process flow, e.g., the optional creation of appointments or reports, this model describes an actual workflow taken from an industrial project based on real customer requirements (in a simplified form). For example, appointments are optional, since prevention workers may visit companies spontaneously if they are nearby. Considering the order execution process, all splitting gateways provide user choices and do not depend on any states or data. The entire diagram in this simple example consists of 6 gateways and 17 sequence flows. In the original form it is more complex (26 gateways and 64 sequence flows). Thus, it is even more difficult to identify optional and mandatory tasks. The explicit expression of the order execution process is shown in Fig. 13. Every complete order execution requires that a request is created and that the order is approved, executed, and closed, which is clearly shown in the proposed extension using general (mandatory) activities. All other activities are highlighted as optional.

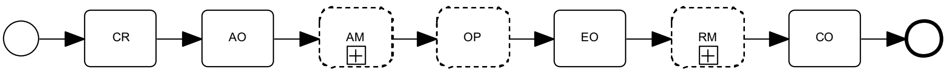


Figure 13: Order Execution Process: Explicit Expression

The explicit expression provides two advantages with respect to understandability. First, the diagram only consists of 8 sequence flows and no gateways are necessary. So it was possible to remove 6 gateways and 9 sequence flows and thereby reduce the structural complexity of the process flow. Secondly, mandatory and optional activities can be distinguished at first sight based on the highlighting (e.g., different line style). It is still necessary to decide whether an optional activity is executed or not but instead of describing this decision through separate gateways and alternative paths, the decision is described within the corresponding activity. The claim that the understandability of the order execution process was increased is described in detail in a preliminary survey (see [Nat11], [Nat12]). In summary, this case study comprises an order execution process and demonstrates the transformation from an implicit to an explicit expression of modality. The structural complexity was reduced and the understandability increased.

5 Conclusion

In this paper, we addressed a drawback of many BPMLs, i.e. to implicitly express modalities through the structure of the process flow. This leads to additional modelling elements such as gateways or sequence flows, and thus complicates the identification of mandatory and optional activities. A preliminary survey affirmed that explicitly optional activities appear to be more intuitive and also positively affect the structural complexity and the understandability of the process flow. Hence, optional activities should be supported in BPMLs like BPMN, UML ADs, YAWL, and EPCs.

We therefore proposed to explicitly support optional activities in business process modelling. Based on the Workflow Patterns we identified optional activities in BPMLs by considering the Control-Flow Patterns, studying in detail the exclusive and multi-choice pattern. We further compared the semantics of diagrams with and without optional activities using ASMs. Then we applied our concept to BPMN and suggested three possible representations for optional activities, i.e. specifying an additional marker, defining a different background colour, or using a dashed line. Finally, we evaluated the proposed approach by introducing a graph transformation system and specifying several transformation rules to transform models with an implicit expression of modality to models with an explicit representation. The transformation always leads to equally many or fewer gateways and sequence flows and, thus, confirms a reduction of structural complexity. Furthermore, we showed within a case study that the use of explicitly optional activities not only reduces the structural complexity but also increases the understandability of a process flow.

In future work, we plan to investigate a full set of deontic classifications of activities, including obligation, permission, prohibition, and conditional commitments. Another issue is related to the limited support for actor modelling in BPMLs like BPMN. In this respect, we are currently developing a new approach including deontic logic and speech act theory that will be evaluated based on the resource perspective of the Workflow Patterns.

Acknowledgement. The project *Vertical Model Integration* is supported within the program "Regionale Wettbewerbsfähigkeit OÖ 2007-2013" by the European Fund for Regional Development as well as the State of Upper Austria. This work was further supported in part by the Austrian Science Fund (FWF) under grant no. TRP 223-N23.

References

[BS03] Egon Börger and Robert Stärk. *Abstract State Machines - A Method for High-Level System Design and Analysis*. Springer Verlag, 2003.

[CK07] Soo Ho Chang and Soo Dong Kim. A Service-Oriented Analysis and Design Approach to Developing Adaptable Services. In *IEEE SCC*, pages 204–211. IEEE Computer Society, 2007.

[KR09] Vera Künzle and Manfred Reichert. Towards Object-aware Process Management Systems: Issues, Challenges, Benefits. In *Proc. 10th Int'l Workshop on Business*

Process Modeling, Development, and Support (BPMDS'09), number 29 in LNBIP, pages 197–210. Springer, 2009.

[KR11] Vera Künzle and Manfred Reichert. PHILharmonicFlows: towards a framework for object-aware process management. *Journal of Software Maintenance*, 23(4):205–244, 2011.

[MA02] Dirk Muthig and Colin Atkinson. Model-Driven Product Line Architectures. In Gary Chastek, editor, *Software Product Lines*, volume 2379 of *LNCS*, pages 79–90. Springer Berlin / Heidelberg, 2002.

[Mic10] Jutta Michely. Help with BPMN notation. `http://www.ariscommunity.com/users/johnm/2010-06-09-help-bpmn-notation`, 2010.

[Nat11] Christine Natschläger. Deontic BPMN. In A. Hameurlain, S. Liddle, K.-D. Schewe, and Xiaofang Zhou, editors, *Database and Expert Systems Applications*, volume 6861 of *LNCS*, pages 264–278. Springer Berlin / Heidelberg, 2011.

[Nat12] Christine Natschläger. *Extending BPMN with Deontic Logic*. PhD thesis, Johannes Kepler Universität Linz (JKU), 2012.

[Oes02] Bernd Oestereich. *Developing Software with UML: Object-Oriented Analysis and Design in Practice*. Pearson Education Ltd, 2 edition, 2002.

[OMG11] OMG. Business Process Model and Notation (BPMN) 2.0. `http://www.omg.org/spec/BPMN/2.0`, 2011.

[PSWW05] Frank Puhlmann, Arnd Schnieders, Jens Weiland, and Mathias Weske. Variability Mechanisms for Process Models. Technical report, 2005.

[RBSS05] Iris Reinhartz-Berger, Pnina Soffer, and Arnon Sturm. A Domain Engineering Approach to Specifying and Applying Reference Models. In *Proc. of the Work. Enterprise Modelling and Information Systems Architectures*, volume 75 of *Lecture Notes in Informatics*, pages 50–63. German Informatics Society, 2005.

[RK08] Maryam Razavian and Ramtin Khosravi. Modeling Variability in Business Process Models Using UML. In *Proc. of the 5^{th} Int. Conf. on Information Technology: New Generations*, ITNG '08, pages 82–87. IEEE Computer Society, 2008.

[Ros09] Marcello La Rosa. *Managing variability in process-aware information systems*. PhD thesis, Queensland University of Technology, 2009.

[Sil09] Bruce Silver. BPMN and Case Management. `http://www.bpmncase.com`, 2009.

[SP06] Arnd Schnieders and Frank Puhlmann. Variability Mechanisms in E-Business Process Families. In *Proc. Int. Conf. on Business Information Systems (BIS 2006)*, pages 583–601, 2006.

[tHvdAAR10] Arthur H.M. ter Hofstede, Wil M.P. van der Aalst, Michael Adamns, and Nick Russell, editors. *Modern Business Process Automation: YAWL and its Support Environment*. Springer, 2010.

[vdAtH11] Wil M.P. van der Aalst and Arthur H.M. ter Hofstede. Workflow Patterns Homepage. `http://www.workflowpatterns.com`, 2011.

[vdAtHKB03] Wil M.P. van der Aalst, Arthur H.M. ter Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14:5–51, 2003.

Collaborative Business Process Modeling

Simon Forster

Simon.Forster@uibk.ac.at

Jakob Pinggera

Jakob.Pinggera@uibk.ac.at

Barbara Weber

Barbara.Weber@uibk.ac.at

Abstract: Research on quality issues of business process models has recently begun to explore the process of creating process models. With growing complexity, the creation of business process models requires the presence of several, potentially spatially distributed, stakeholders. As a consequence, the question arises how this affects the process of process modeling. In this paper, we present an extension to Cheetah Experimental Platform, specifically designed for investigating how process models are collaboratively created. Furthermore, we illustrate how various levels of interactions are supported by our modeling editor and outline its usage for analyzing the process of collaboratively creating process models. We believe that this extension is a first step toward understanding the process of process modeling in a collaborative setting which will ultimately help us to develop better process modeling environments.

1 Introduction

"*Business process modeling is the task of creating an explicit, graphical model of a business process from internalized knowledge on that process*" [IRRG09]. The resulting business process models play an important role for the management of business processes [BRU00], depicting how "*various tasks are coordinated to achieve specific organizational goals*" [MDS$^+$10]. Such process models are used to build a consensus among stakeholders involved within the business process [MDS$^+$10]. Therefore, the quality of business process models is essential [KSJ06] as it constitutes a measure of the fulfillment of its purpose (e.g., to serve as a basis for a system development project) [Rit09b]. However, industrial process model collections suffer from a range of quality problems. Understandability of process models suffers from poor quality which subsequently hampers the models' maintainability [Men08, WR08]. Examples for typical quality problems are non-intention-revealing or inconsistent naming [MRR10], redundant process fragments [HBR10] or overly large and unnecessarily complex models [SOM08].

To address these quality problems significant research has been conducted in recent years on factors that impact process model understandability and maintainability [Men08, WR08, KSJ06]. Focus of these works is on the *outcome of the modeling process* [AH00, GL06], i.e., the resulting process model. In turn, relatively little emphasis has been put on the fact that model quality presumably depends upon the modeling process that was followed to create it, i.e., the *process of process modeling* (PPM) [PZW$^+$12]. For example, [PSZ$^+$12] aims at a better understanding of the PPM, i.e., the formalization of a process model from an informal requirements specification. Thereby, [PSZ+12] assumes a modeling setting where a single model engineer is creating a process model and where the communication between model engineers and domain experts is mediated via an informal requirements specification [PZW+12]. However, when looking at the complexity of real life projects it is often not possible to have only a single model engineer creating the corresponding business process model, since knowledge of the business process might be distributed over a number of domain experts [HPvdW05]. Similarly, the corresponding knowledge to create the process model has to be distributed among model engineers. As a consequence, various domain experts and model engineers are involved in the development cycle, who collaboratively create a process model [RKdV08]. By this close collaboration the border between requirements elicitation and formalization becomes blurred. In fact, the distinction between those two phases disappears and is replaced by an iterative process performing them repeatedly.

Even though collaborative process modeling settings are increasingly found in practice [Rit09a, MRW12] and results in software engineering have shown that collaboration can increase quality and efficiency significantly [WKCJ00], the way how process models are collaboratively created is hardly understood [Rit09a]. We want to extend existing work on the PPM, which focuses on single model engineer settings, toward a collaborative setting where multiple stakeholders (e.g., domain experts and model engineers) collaboratively create a process model. Therefore, we developed a modeling tool that enables researchers to record the PPM within collaborative environments as well as the analysis of the data gathered during this process. Our tool not only features a collaborative modeling editor but specifically aims at investigating the PPM to gain an in-depth understanding of the PPM involving multiple stakeholders. Ultimately, we are trying to evaluate whether and to what extent improvements in model quality as well as efficiency can be attributed to collaboration. Therefore, the tool and evaluation techniques presented in this paper serve as starting point for further research in the area of collaborative process modeling. We are planning to conduct case studies and experiments using this tool as the underlying basis.

The remainder of this paper is structured as follows: Sect. 2 presents backgrounds on the PPM and introduces the Cheetah Experimental Platform (CEP) for single model engineer settings, Sect. 3 then details the extensions made to CEP in order to support collaborative process modeling. Related work is discussed in Sect. 4. Finally, Sect. 5 concludes the paper with a summary and outlook on future work.

2 Background

This section provides background information on the PPM and enumerates the individual processes involved (cf. Sect. 2.1). Furthermore, we introduce CEP for single modeler settings (cf. Sect. 2.2).

2.1 The Process of Process Modeling

During the formalization phase process modelers are working on creating syntactically correct process models reflecting a given domain description by interacting with the process modeling tool [HPvdW05]. This modeling process can be described as an iterative and highly flexible process [CWWW00, Mor67], dependent on the individual modeler and the modeling task at hand [Wil95]. At an operational level, the modelers interactions with the tool would typically consist of a cycle of the three successive phases of (1) comprehension (i.e., the modeler forms a mental model of domain behavior), (2) modeling (i.e., the modeler maps the mental model to modeling constructs), and (3) reconciliation (i.e., the modeler reorganizes the process model) [PZW+12, SKW11].

Comprehension. Research on human cognition and problem solving has shed light on comprehension. According to [NS72], when facing a task, the problem solver first formulates a mental representation of the problem, and then uses it for reasoning about the solution and which methods to apply for solving the problem. In process modeling, the task is to create a model which represents the behavior of a domain. The process of forming mental models and applying methods for achieving the task is not done in one step applied to the entire problem. Rather, due to the limited capacity of working memory, the problem is broken down to pieces that are addressed sequentially, chunk by chunk [SKW11, PZW+12].

Modeling. The modeler uses the problem and solution developed in working memory during the previous comprehension phase to materialize the solution in a process model (by creating or changing it) [SKW11, PZW+12]. The modelers utilization of working memory influences the number of modeling steps executed during the modeling phase before forcing the modeler to revisit the problem for acquiring more information [PZW+12].

Reconciliation. After modeling, modelers typically reorganize the process model (e.g., renaming of activities) and utilize the process model's secondary notation (e.g., notation of layout, typographic cues) to enhance the process model's understandability [Pet95, MRC07]. However, the number of reconciliation phases in the PPM is influenced by a modeler's ability of placing elements correctly when creating them, alleviating the need for additional layouting [PZW+12].

2.2 Cheetah Experimental Platform

In order to get a detailed picture of how process models are created, we use Cheetah Experimental Platform (CEP). CEP has been specifically designed for investigating the PPM in a systematic manner [PZW10]. When considering a single modeler interacting with a process modeling environment, the development of process models consists of adding nodes and edges to the process model, naming or renaming these activities, and adding conditions to edges. In addition to these interactions a modeler can influence the process model's secondary notation, e.g., by laying out the process model using move operations for nodes or by utilizing bendpoints to influence the routing of edges (cf. Table 1).

CEP instruments a basic process modeling editor to record each user's interactions together with the corresponding time stamp in an event log, describing the creation of the

User Interaction	Description	User Interaction	Description
CREATE NODE	Create activity or gateway	RENAME	Rename an activity
DELETE NODE	Delete activity or gateway	UPDATE CONDITION	Update an edge's condition
CREATE EDGE	Create an edge connecting two nodes	MOVE NODE	Move a node
DELETE EDGE	Delete edge	MOVE EDGE LABEL	Move the label of an edge
CREATE CONDITION	Create an edge condition	CREATE/DELETE/MOVE EDGE BENDPOINT	Update the routing of an edge
DELETE CONDITION	Delete an edge condition		
RECONNECT EDGE	Reconnect edge from one node to another		

Table 1: User Interactions with CEP

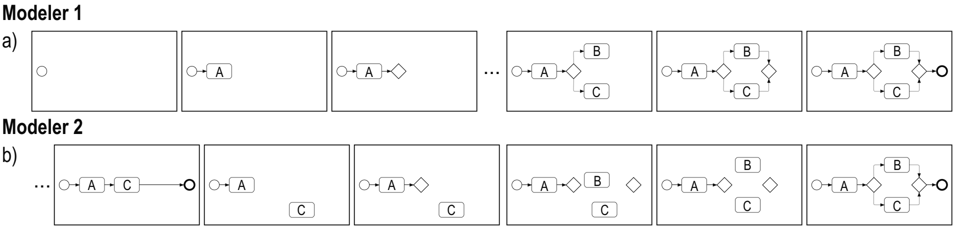


Figure 1: Two Different Processes of Process Modeling to Create the Same Process Model

process model step by step. By capturing all interactions with the modeling tool, we are able to *replay* a recorded modeling process at any point in time without interfering with the modeler or her problem solving efforts. This allows for observing how the process model unfolds on the modeling canvas[1]. Fig. 1 illustrates the basic idea of replaying the creation of a process model. Fig. 1a shows several states of a typical modeling process as it can be observed during replay. Fig. 1b shows the states of a different PPM instance that nonetheless results in the *same* model. This replay functionality of CEP allows to observe in detail *how* modelers create the model on the canvas. Beside the ability of replaying the modeling process, the captured interactions with the modeling tool can be used for evaluation of the PPM. The Modeling Phase Diagram (MPD) is an example of a visualization of the PPM that can be generated using CEP [PZW$^+$12]. This technique classifies the recorded modeling activities according to the cognitive phases introduced in Section 2.1 and provides a graphical presentation for further analysis. A PPMChart is another method illustrating the data captured during the modeling process [CVP$^+$12]. Additionally, CEP provides a calculation extension for various types of metrics [PSZ$^+$12].

3 Collaborative Modeling using CEP

CEP (cf. Sect. 2.2) aims at investigating the process of process modeling within single modeling settings. Here we introduce a tool to analyze the PPM within collaborative modeling settings (cf. Sect. 3.1). Moreover, this section provides a detailed view on extensions necessary to support collaborative business process modeling (cf. Sect. 3.2) as well as extensions required for analyzing the PPM within collaborative modeling settings (cf. Sect. 3.3).

[1] A demonstration of CEP's replay function is available at http://cheetahplatform.org

3.1 Collaborative Processes

When process models are created collaboratively, the individual processes of process modeling (*comprehension*, *modeling* and *reconciliation*) as introduced in Sect. 2 are not sufficient. In addition, team processes take place during which teams exchange information, create solution options, exchange knowledge, evaluate and negotiate alternatives and assess their own processes [FSJS+10]. As a result, the team is building further knowledge and a shared understanding of the process model [FSJS+10, Rit12a]. In order to be able to analyse these processes we extend CEP with support for collaborative modeling. Furthermore, we extend the replay functionality to replay the data retrieved from collaborative features. This, in turn, provides the ability to analyze team processes in detail in combination with the individual processes of single team members.

3.2 CEP Modeler Extension

Extensions to the modeling editor were necessary to enable users to collaboratively and concurrently edit a business process model. In addition, there are a number of levels of social interaction that have to be considered when developing a collaborative modeling tool. According to [MRW12] those levels are *awareness*, *communication*, *coordination*, *group decision making*, and *team-building*.

Awareness. The ability of the participants seeing the same objects as well as the changes done by others. Another example would be a list displaying the names of all participants making participants aware of each other.

Communication. Sending or receiving messages is crucial in case of spatially separated participant. Hence, this level of interaction aims at exchanging messages and establishing a common language [MRW12].

Coordination. As soon as there is more than one person involved in a task, splitting and distributing those tasks is a crucial aspect in collaborative modeling and requires coordination support.

Group decision making. Again, when multiple people are working together, they need a mechanism to propose their solutions to problems. Those solutions can then be evaluated and selected. Meaning the participants are negotiating about the models [Rit07].

Team-building. As a result of collaboratively working together, the team is building further knowledge [FSJS+10] as the participants exchange information between each other.

Fulfilling those levels of interaction is essential for effectively supporting collaborative process modeling.

3.2.1 Collaborative Modeling Support

The foremost level of interaction is the *awareness* level as it is crucial for spatially separated domain experts and model engineers to be able to graphically create process models in a collaborative manner [MRW12]. Participants are able to see changes made to the process model immediately. As an example, Fig. 2 illustrates participants using the collaborative modeling editor. Participants *Alice* and *Bob* are working on the same model.

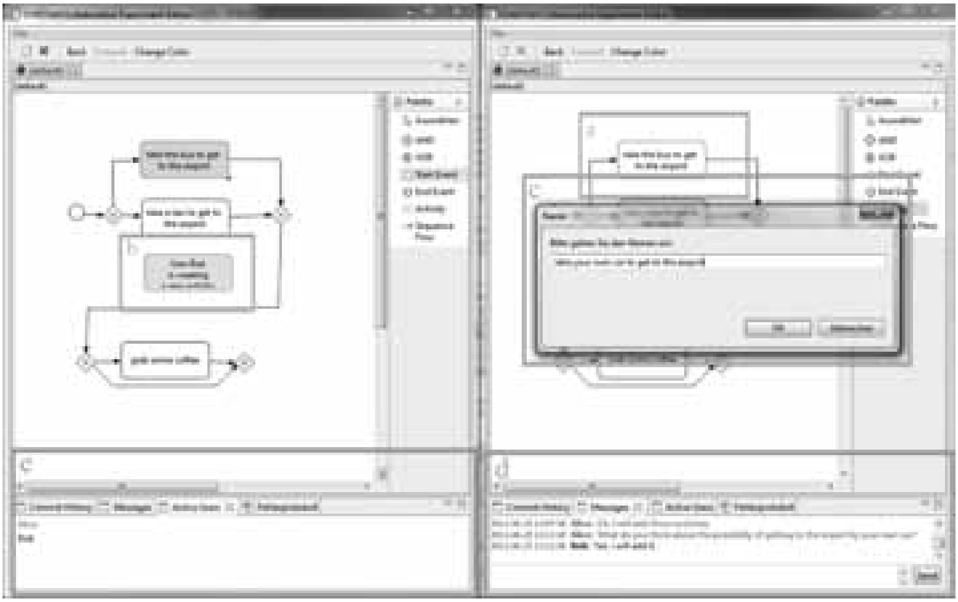


Figure 2: Participants using the Collaborative Process Modeling Editor for Working on the Same Model

Hence, they can see the same process model on their respective canvases. To further increase change awareness participants are supported to identify on which parts of the model other participants are currently working. For this purpose, activities currently selected or moved by other participants are highlighted (cf. Fig. 2a) using colors. Therefore, a unique color is linked to each participant. Meaning, at any time during the modeling process, participants are not only able to see which activities are edited by other participants, but they can also see which participants are working on specific activities. In addition, the tool is visualizing *if* and *who* is currently creating a new activity using colors. Creating an activity is a task of several seconds as the participant (*Bob*) has to enter a name for the activity (Fig. 2c). In order to increase change awareness when creating a new activity the tool already displays the new activity (cf. Fig. 2b) on the canvases of the other participants (*Alice*). Until the participant has entered the name of the activity, the other participants can see the activity displaying the name of the participant creating that activity. This way, the coordination of creating new activities can be eased.

3.2.2 Communication Support

As already mentioned, within a collaborative modeling environment participants are spatially separated from each other. Since *communication* is another important level of interaction we integrated a communication mechanism into CEP. More precisely, we integrated a "chat window" into CEP. With the help of this feature the different participants taking part in the modeling process can exchange knowledge regarding different aspects of the model, create solution options, evaluate and negotiate alternatives [Rit07], and assess their

User Interaction	Description
Join	Join a modeling session
Leave	Leave a modeling session
Message	Send a message
Change	Send a modeling interaction (e.g., CREATE NODE, CREATE EDGE, RENAME)
Conflict	Indicate a conflicting model interaction (e.g., CREATE NODE, CREATE EDGE, RENAME)

Table 2: User Interactions with Collaborative CEP

own process [FSJS$^+$10]. Hence, this feature additionally aims at supporting *coordination* and *group decision making*.

When developing this feature we also considered awareness support. For this, messages exchanged between the participants are highlighted in the colors linked to participants within the chat window (cf. Fig. 2d). In addition, to facilitate communication and to foster awareness between modeling participants, we added another window which displays all currently connected participants (cf. Fig. 2e).

Using those features our tool aims at the fulfillment of each level of interaction (i.e., *awareness*, *communication*, *coordination* and *group decision making*) according to [MRW12]. As a result, our tool supports the *team-building* process within the collaborative PPM as it provides a formal (i.e., modeling editor) as well as an informal (i.e., communication window) way interacting with other participants [MRW12].

3.2.3 Logging Extensions

Each interaction with the modeling editor is automatically tracked and logged by our tool resulting in revision control of the commands sent. Using this revision control the modeling editor also provides conflict management preventing race conditions. In case, two conflicting commands (e.g., deleting and moving the same activity) are sent to the server at the same time, the first client trying to execute the conflicting command (moving the activity) recognizes the problem and marks it being a conflict. As a result, the other clients will not execute the conflicting command. Like the actual modeling process (e.g., creation/deletion of activities or edges) also the messages sent (cf. Sect. 3.2.1) are tracked by our prototype. Moreover, the messages are automatically linked to the activities selected during creation of the message. This information, including timestamp and information regarding the user, is important for later analysis of the modeling process (cf. Sect. 3.3.2). Additionally, the modeling events logged within CEP are extended. In order to be able to retrieve the user's data performing modeling commands, we extend the data model of CEP. Therefore, we wrap the additional information captured with our tool around the modeling commands already logged by CEP (cf. Tab. 1). Tab. 2 lists the commands created and exchanged by collaborative CEP.

The modeling commands of CEP (cf. Tab. 1) are wrapped into the *Change* command of collaborative CEP (cf. Tab. 2) and stored within a data model (cf. Tab. 3).

Hence, after the modeling session ended, it can be identified *who* changed *which* elements

Attribute	Description
Revision-ID	Unique identifier of the command
User	User executing the command
Time	Timestamp of the command
Type	Type of the command (i.e., *Join*, *Leave*, *Message*, *Change*, *Conflict*).
Command	The user interaction

Table 3: Data Model of Collaborative CEP

as well as *when* these elements were changed. In addition, it can be analyzed which messages were exchanged.

3.3 Analysis Extension

Beside the extensions of the model editor (cf. Sect. 3.2) also the analysis capabilities of CEP have to be extended (cf. Sect. 3.3.1) in order to track and evaluate the team processes during a collaborative modeling session and to enable an integrated analysis with the individual processes (cf. Sect. 3.3.2).

3.3.1 Replay Functionality

Using CEP as a grounding makes it possible to use the built in “replay functionality” (cf. Sect. 2.2) which allows replaying process models created with the CEP modeler. In order to support the integrated replay of how the model was created, including the communication which took place, we extend the replay functionality of CEP. Our model editor not only records the modeling steps, but also communication between participants which can later be chronologically reviewed and evaluated.

Again, having usability and the corresponding *awareness* level in mind, we made this feature available for participants. This way, participants joining the modeling session at a later point in time have the opportunity to chronologically recap the evolution of the business process model in terms of activities created as well as messages exchanged yielding a deeper understanding of the process and increasing the awareness of which changes have occurred since they left the modeling session. This feature cannot only be used by participants joining a session later, but at all stages during the modeling process in case participants want to recap why a specific element was modeled this way.

This functionality is illustrated in Fig. 2. The right instance depicts the latest version of the model whereas the left instance shows the model at an earlier point in time. Unless participants are not on the latest version, the modeling window is locked, meaning the participants are not able to interact with the model (e.g., create new activities) or other participants using the chat window. Interacting with the model again is only possible as soon as the participants are at the latest version of the model.

3.3.2 Metrics

After capturing the PPM within a collaborative environment the next step is the evaluation of the retrieved data. More precisely, team processes can be analyzed in addition to the

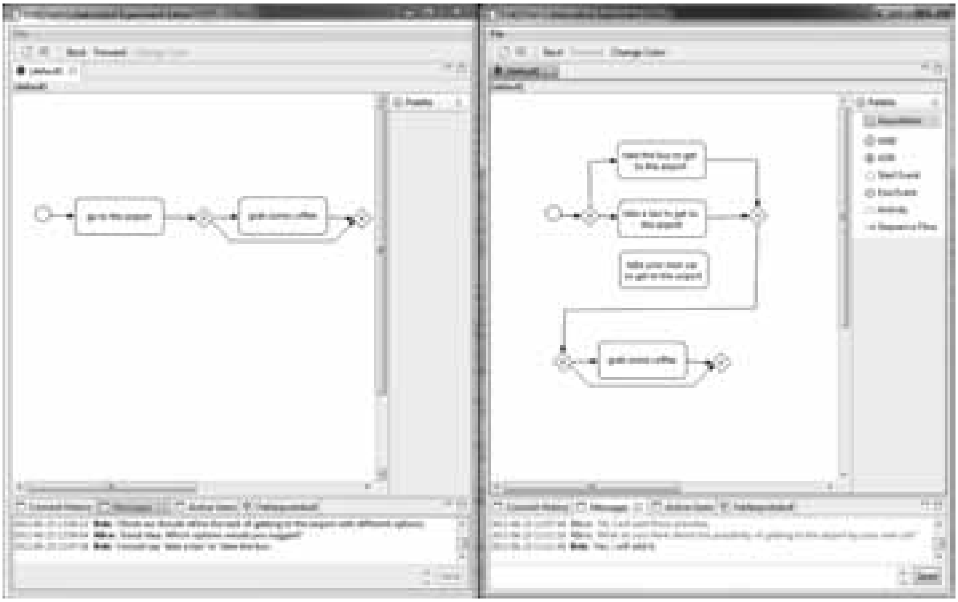


Figure 3: Two Instances Working on the Same Model at Different Points in Time

individual processes [PZW+12]. In order to make qualified assertions, we will develop visualizations, algorithms and metrics to analyze the collaborative modeling process. In particular, we present metrics for different perspectives. Here, the *model perspective* examines the modeling aspects themselves. For example, this perspective comprises measures like number of changes per activity potentially indicating modeling elements which caused difficulties during the modeling process or which caused lots of controversy. Whereas, investigating the participation and collaboration of the team members is in the focus of the *team perspective*. Combining those two perspectives results in the *integrated perspective*. This perspective investigates the interaction of the participants with the model.

"Number of changes per activity" (cf. Tab. 4) can be a measure for the difficulty of a single model element whereas the two team perspective metrics can be a measure for the participants providing the most domain knowledge. In order to be able to filter out valuable comments, we will be utilizing the CoPrA method for analyzing conversation protocols [SWM12]. Moreover, the metrics for the integrated perspective are a measure for the involvement of the participants as well as the importance of single activities. Here, "number of changes per activity by different participants" and "number of comments per modeling element" could indicate activities where the participants had to find a consensus during the modeling phase. Knowledge transfer as well as improved quality are positive results of this negotiation phase. "Number of activities created per participant" and "number of changes per participant" could expose participants claiming the leading role during the modeling process.

For the purpose of visualization, heat maps will be utilized. For example, related to metric

Perspective	Possible metrics
Model perspective	Number of changes per activity
Team perspective	Number of comments made per participant Number of times a critique is offered for a particular modeling element
Integrated perspective	Number of activities created per participant Number of changes per participant Number of changes per activity by different participants Number of comments per modeling element

Table 4: Metrics for the evaluation of the modeling process

"number of comments per modeling element" (cf. Tab. 4) modeling elements could be highlighted in different colors depending on the number of comments. Elements discussed a lot would then appear darker than less discussed ones.

The data needed for those metrics can be generated automatically from the information logged by CEP (cf. Sect. 3.2.3) because the name of the participant is assigned to each modeling command performed within CEP. Also, the name of the participant is linked to the message. A semiautomatic algorithm for linking comments with the corresponding model elements is under development, supporting the subsequent coding of conversation protocols (e.g., using the negotiation patterns [Rit07]). Still, there are metrics (e.g., number of times a critique is offered for a particular modeling element) where it is not possible to generate the needed data automatically out of the information logged. As already mentioned, evaluating those data will be done using the CoPrA method [SWM12]. The resulting data can then be utilized for statistical analysis.

3.3.3 Experimental Workflow

Ultimately, the goal will be performing case studies as well as conducting controlled experiments using this tool. Such experiments consist of a series of tasks that have to be executed by the experiment's subjects, referred to as *Experimental Workflow* [PZW10]. As the collaborative modeling editor builds upon CEP which includes the ability of incorporating components commonly used in experiments (e.g., questionnaires, surveys, tutorials) into controlled experiments, we can make use of this infrastructure. This way we can combine the automatically collected data regarding the process of process modeling with information provided by the modeler before or after the modeling session (e.g., demographic information, domain knowledge, process modeling knowledge, perceived quality, perceived ease of use, perceived usefulness, mental effort).

4 Related Work

This section presents related work in the area of collaborative process modeling.

Research on the Process of Process Modeling. The PPM is concerned with the interaction of the participants (e.g., domain experts and model engineers) during modeling. How important the modeling process itself really is beside the actual outcome is stated in [HPvdW05]. There has already been some research on the PPM [PZW+12, PSZ+12]. However these works focus on modeling settings where a single model engineer creates the process model, whereas with our tool it is possible to investigate how collaborative process modeling impacts the process of creating process models. As an exception, [Rit07] also investigates on collaborative modeling settings concentrating on the negotiation phase of this process. In addition, the team processes (e.g., combination of best performing teams) are investigated in [Rit12a] and evaluated in respect to model quality. Again, with our tool it is possible to analyze the *process of process modeling* and investigate the team processes in addition to the individual processes.

Alternative Process Modeling Tools. There already exist some environments fostering collaboration between different stakeholders. One example of such an environment is Collaborative Modeling Architecture [Rit10]. The COMA Tool provides process model collaboration by means of negotiation on proposed models by the participants.

In contrast to this collaboration methodology, where the participants work in an asynchronous way there also exists another one, synchronous collaboration or concurrent modeling where the participants are working synchronously together on the same model. The advantage of this approach is the fact that participants are able to track model changes immediately. Examples are the Signavio Process Editor[2] and the Software AG's ARIS[3] collaborative tool where it is possible to work simultaneously together on one model using a web browser. CoMoMod [DHFL11] is another example of a collaborative modeling tool. Beside the lack of supporting the BPMN process modeling notation it aims at the modeling outcome rather than the modeling process itself. The same holds for the Signavio Process Editor as well as the Software AG's ARIS collaborative tool.

Here, we do not want to create an alternative but the opportunity to analyze collaboration in a controlled manner. With the possibility of tracking the modeling process, CEP is an ideal platform for this purpose.

Research on Collaborative Process Modeling. There has already been some research in the area of collaborative process modeling [Rit12b, Rit12a]. The team-building processes when creating a model collaboratively using a proposal based tool (COMA) and allowing face to face communication are investigated in [Rit12b] and evaluated in respect to model quality. Again, using our tool it is possible to analyze the PPM. Furthermore, our tool provides the possibility of synchronously working on the same model while being spatial separated using an integrated communication channel.

Research on Process Model Quality. Different research has already been done in the area of business process model quality [Rit09b]. Additionally, there exist guidelines describing quality considerations for business process models [BRU00], the Seven Process Modeling Guidelines (7PMG) defining desirable characteristics of a business process model [MRvdA10] or identifying various aspects of process models' quality [KSJ06]. [MRC07] investigated the influence of model complexity on process model understandability. Prediction models for usability and maintainability for process models are provided by [RSG+09]. The impact of different quality metrics on error probability was

[2] http://www.signavio.com/
[3] http://www.softwareag.com/

investigated in [MVvD+08, Men08]. The role visual notations are playing for cognitive effectiveness is discussed in [Moo09]. The commonality of those works is the focus on the resulting process model, whereas only little attention is payed on the process of modeling itself.

5 Summary and Outlook

This paper introduced a tool to support both, collaboratively creating a business process model as well as analyzing the *process of process modeling* within a collaborative modeling setting. Therefore, we extended Cheetah Experimental Platform with collaboration features like concurrent modeling and a communication channel. Furthermore, we introduced metrics for the evaluation of the data captured by the tool. Therefore, the collaborative modeling editor logs each user interaction for later analysis. The integrated replay functionality not only fosters the evaluation of the PPM but also yields a deeper understanding of the process and increases the awareness of which changes occurred since participants left the modeling session.

After extending CEP to support collaborative process modeling settings we will next perform an exploratory study. The information retrieved in this study will be used for further improving the collaborative modeling editor. Afterwards, we will conduct controlled experiments using this tool. The data obtained in those experiments will then be evaluated using the visualizations and metrics introduced in Sect. 3.3.2.

Further, we plan to extend our prototype with additional features to increase the usability of the tool. An example would be the integration of speech. This would complement the chat window with a convenient way of communicating with other participants and approaches a face-to-face interaction.

Acknowledgements. This research was funded by the Austrian Science Fund (FWF): P23699-N23. Particularly, we would like to thank Peter Zangerl for his developing efforts.

References

[AH00] W.M.P. van der Aalst and A.H.M. ter Hofstede. Verification of Workflow Task Structures: A Petri-net-based Approach. *Information Systems*, 25(1):43–69, 2000.

[BRU00] Jörg Becker, Michael Rosemann, and Christoph Uthmann. Guidelines of Business Process Modeling. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 30–49, London, UK, 2000. Springer-Verlag.

[CVP+12] Jan Claes, Irene Vanderfeesten, J. Pinggera, H.A. Reijers, B. Weber, and G. Poels. Visualizing the Process of Process Modeling with PPMCharts. In *Proc. TAProViz '12 (accepted)*, 2012.

[CWWW00] Andrew W. Crapo, Laurie B. Waisel, William A. Wallace, and Thomas R. Willemain. Visualization and the process of modeling: a cognitive-theoretic view. In *Proc. KDD '00*, pages 218–226, 2000.

[DHFL11] Thorsten Dollmann, Constantin Houy, Peter Fettke, and Peter Loos. Collaborative Business Process Modeling with CoMoMod - A Toolkit for Model Integration in Distributed Cooperation Environments. In *Proc. WETICE '11*, pages 217–222, 2011.

[FSJS+10] Stephen M. Fiore, Kimberly A. Smith-Jentsch, Eduardo Salas, Norman Warner, and Michael Letsky. Towards an understanding of macrocognition in teams: developing and defining complex collaborative processes and products. *Theoretical Issues in Ergonomics Science*, 11(4):250–271, 2010.

[GL06] Volker Gruhn and Ralf Laue. Complexity metrics for business process models. In *Proc. BIS '06*, pages 1–12, 2006.

[HBR10] Alena Hallerbach, Thomas Bauer, and Manfred Reichert. Capturing Variability in Business Process Models: The Provop Approach. *Journal of Software Maintenance*, 22(6-7):519–546, 2010.

[HPvdW05] S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide. A Fundamental View on the Process of Conceptual Modeling. In *Proc. ER '05*, pages 128–143, 2005.

[IRRG09] Marta Indulska, Jan Recker, Michael Rosemann, and Peter F. Green. Business Process Modeling: Current Issues and Future Challenges. In *Proc. CAiSE*, pages 501–514, 2009.

[KSJ06] John Krogstie, Guttorm Sindre, and Havard Jorgensen. Process models representing knowledge for action: a revised quality framework. *EJIS*, 15:91–102, 2006.

[MDS+10] Debdoot Mukherjee, Pankaj Dhoolia, Saurabh Sinha, Aubrey J. Rembert, and Mangala Gowri Nanda. From Informal Process Diagrams to Formal Process Models. In *Proc. BPM '10*, pages 145–161. Springer, 2010.

[Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction and Guidelines for Correctness*. Springer, 2008.

[Moo09] Daniel L. Moody. The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35:756–779, 2009.

[Mor67] William T. Morris. On the Art of Modeling. *Management Science*, 13(12):B–707–B–717, 1967.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In *Proc. BPM '07*, pages 48–63, 2007.

[MRR10] Jan Mendling, Hajo A. Reijers, and Jan Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems*, 35(4):467–482, 2010.

[MRvdA10] Jan Mendling, Hajo A. Reijers, and Wil M. van der Aalst. Seven Process Modeling Guidelines (7PMG). *IST*, 52(2):127–136, 2010.

[MRW12] Jan Mendling, Jan C. Recker, and Johannes Wolf. Collaboration features in current BPM tools. *EMISA Forum*, 32(1):48–65, 2012.

[MVvD+08] Jan Mendling, H.M.W. Verbeek, Boudewijn van Dongen, W.M.P. van der Aalst, and Gustav Neumann. Detection and Prediction of Errors in EPCs of the SAP Reference Model. *DKE*, 64(1):312–329, 2008.

[NS72] A. Newell and H.A. Simon. *Human problem Solving*. Prentice Hall, 1972.

[Pet95] Marian Petre. Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. *Commun. ACM*, pages 33–44, 1995.

[PSZ+12] J. Pinggera, P. Soffer, S. Zugal, B. Weber, M. Weidlich, D. Fahland, H.A. Reijers, and J. Mendling. Modeling Styles in Business Process Modeling. In *Proc. BPMDS '12*, pages 151–166, 2012.

[PZW10] Jakob Pinggera, Stefan Zugal, and Barbara Weber. Investigating the Process of Process Modeling with Cheetah Experimental Platform. In *Proc. ER-POIS'10*, pages 13–18, 2010.

[PZW+12] Jakob Pinggera, Stefan Zugal, Matthias Weidlich, Dirk Fahland, Barbara Weber, Jan Mendling, and Hajo A. Reijers. Tracing the Process of Process Modeling with Modeling Phase Diagrams. In *Proc. ER-BPM '11*, pages 370–382, 2012.

[Rit07] Peter Rittgen. Negotiating Models. In *Proc. CAiSE '07*, pages 561–573, 2007.

[Rit09a] Peter Rittgen. Collaborative Modeling – A Design Science Approach. In *Proc. HICSS '09*, pages 1–10, 2009.

[Rit09b] Peter Rittgen. Collaborative modeling of business processes: a comparative case study. In *Proc. SAC '09*, pages 225–230, 2009.

[Rit10] Peter Rittgen. Collaborative Business Process Modeling Tool Support for Solving Typical Problems. In *Proc. Conf-IRM '10*, 2010.

[Rit12a] Peter Rittgen. End-User Involvement and Team Factors in Business Process Modeling. In *Proc. HICSS '12*, pages 180–189, 2012.

[Rit12b] Peter Rittgen. The role of editor in collaborative modeling. In *Proc. SAC '12*, pages 1674–1679, 2012.

[RKdV08] Michiel Renger, Gwendolyn L. Kolfschoten, and Gert-Jan de Vreede. Using Interactive Whiteboard Technology to Support Collaborative Modeling. In *Proc. CRIWG '08*, pages 356–363, 2008.

[RSG+09] Elvira Rolon, Laura Sanchez, Felix Garcia, Francisco Ruiz, Mario Piattini, Danilo Caivano, and Giuseppe Visaggio. Prediction Models for BPMN Usability and Maintainability. In *Proc. CEC '09*, pages 383–390, 2009.

[SKW11] Pnina Soffer, Maya Kaner, and Yair Wand. Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In *Business Process Management Workshops*, pages 357–369. Springer, 2011.

[SOM08] Martín Soto, Alexis Ocampo, and Jürgen Münch. The Secret Life of a Process Description: A Look into the Evolution of a Large Process Model. In *Proc. ICSP '08*, pages 257–268, 2008.

[SWM12] Isabella Seeber, Barbara Weber, and Ronald Maier. CoPrA: A Process Analysis Technique to Investigate Collaboration in Groups. In *Proc. HICCS '12*, pages 363–372, 2012.

[Wil95] Thomas R. Willemain. Model Formulation: What Experts Think about and When. *Operations Research*, 43(6):916–932, 1995.

[WKCJ00] Laurie Williams, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. Strengthening the Case for Pair Programming. *IEEE Software*, 17(4):19–25, 2000.

[WR08] Barbara Weber and Manfred Reichert. Refactoring Process Models in Large Process Repositories. In *Proc. CAiSE '08*, pages 124–139, 2008.

Timeline Visualization for Documenting Process Model Change

Sonja Kabicher-Fuchs[1], Simone Kriglstein[2], and Kathrin Figl[3]

[1]University of Vienna, Austria
Faculty of Computer Science, Workflow Systems and Technology Group
sonja.kabicher-fuchs@univie.ac.at
[2]SBA Research, Vienna, Austria*
SKriglstein@sba-research.at
[3]Vienna University of Economics and Business, Austria
kathrin.figl@wu.ac.at

Abstract: Organizations need to deal with change in order to remain competitive in a dynamic, global business world. When business processes need to be redesigned, it often remains unclear why a process evolved to its current form. Against this background, process change documentations can be helpful in order to illustrate and capture change information. The main purpose of this paper is to expand our understanding on how to visualize process change information and to explore the concept of timeline visualization for this application area. Using an expert inspection approach, we first identify shortcomings of existing timeline tools in meeting visualization requirements of business process change. We propose a conceptual design for change visualization using timelines that aims to support a contextual, temporal and comprehensive documentation of business process model changes.

1 Introduction

One major incentive of organizations to document processes and to record process change is to conform to internal or external requirements, such as the ISO 9000 standards and guidelines. The documentation of change may refer to various attributes of change, for example, the reasons for change, planned changes, and actual changes performed. Time is as well a representative attribute of change which specifies e.g., the point of time and time period change is planned, actually performed, applied, and the number of changes within a specific time span. As the predominant purpose of timelines is to visualize historical, current and future actions and events, and simultaneously support the representation of complementary information, e.g. in different views, we chose timelines as visualization method for process model change documentation. To provide an explanation why timelines are especially suitable to visualize the timely change of processes we can draw on research in cognitive psychology. For instance, [HDL68] demonstrated that people

*funded by COMET K1, FFG - Austrian Research Promotion Agency.

rely on internal spatial representations (mental models) when they solve logic problems in reasoning, even when those problems are nonspatial and abstract. Literature reveals that for the two relation pairs relevant for the context of process change: earlier-later and cause-effect there is a clear preference 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. This natural association between left-to-right and time is also reflected in the fact that temporal relations are usually expressed from left-to-right on the horizontal axis in graphs [TKW91]. Based on these observations, we applied timelines as visualization method to support documentation of changes in process models. In this paper we investigate timelines for process model version management. Our research was guided by the following questions: (1) Which attributes of change may be helpful for understanding changes and required for documenting process change? (2) Can timelines help to document and visualize change of process models in a structured and interactive way? We evaluated selected timeline tools according to change visualization requirements in business processes by means of expert inspection. Based on the gained insights of the inspection sessions, we developed a visualization concept in order to offer: (a) the documentation of process model evolution and detailed temporally related information about change, (b) an understanding of the change in a bigger context (possible triggers/events for change as well as temporal relationships between triggers and actions), and (c) detailed information about the change per se that is loosely coupled with the time attribute of change (e.g., what was changed and who performed the change).

2 Changes of Business Process Models

In this section we focus on change in the context of business process models and discuss change attributes (illustrated in Figure 1) valuable for documentation in detail.

Process models. As a process model can be changed several times during its lifecycle, several versions of the process model can exist. Business process models consist of a set of activities with execution constraints between them and they are performed to achieve business goals. A business process model is a blueprint for a set of business process instances, which represent concrete cases of the process [Wes07, p.7]. After changing actions are taken, the resulting process model represents the desired adjusted process schema. There exist different approaches for version control in business process change management. For example, repositories can be used to support an incremental numbering system for process model versions (see e.g., [IBM05]). A further approach is to illustrate evolvement along the two axes process improvement and temporary adaptation (represented by a three-digit string x.y.z where x denotes the major version, y denotes the minor version and z denotes the temporary variation). Another approach it to present versions as directed graphs (see e.g., [ZL07]), or as version trees (see e.g., [KG99]). Other tools, such as process mashups like Gravity [DKN$^+$10]) support the logging of conducted model changes. For the process model timeline concept presented in this paper we use the following graphs (as discussed in our previous work [KKRM11]: the initial graph A which illustrates the first or the current version of the process, the adjusted graph A' which illustrates the subsequent version

Change attributes	Description	Artifact	Examples
Process models	Change is performed on process models. The result are versions of process models.	Initial process models Changed process models	Process model *A* version 1 Process model A version 2
Actions	Change actions describe (a) what has to be changed and (b) what was changed.		
Action planned	Planned change actions before change is actually performed.	Documentation of planned change actions	parallelize (S,B,C,D) = <deleteEdge(S,B,C), deleteEdge(S,C,D), insertEdge(S,A,C), insertEdge(S,A,D)>
Evaluation	Evaluation of planned change actions before change actions are performed.	Evaluation results of planned change actions	Parallel work is possible. Tasks need not to wait for data from other tasks. Analyse results show a xy% decrease of process duration.
Action taken	Change actions performed on the process model.	Documentation of performed change actions	parallelize (S,B,C,D) = <deleteEdge(S,B,C), deleteEdge(S,C,D), insertEdge(S,A,C), insertEdge(S,A,D)>
Change tracking graph	The change tracking graph contains the graph elements of the initial graph that are affected and not affected by the change.	Change tracking graph	Change tracking graph A*_(v1-v2)
Time	Date or time frame of the change.	Documentation of the date	21st Jan 2010 - 23rd Sept 2010
Trigger	Reasons for change.	Documentation of change triggers	Legislation amendment, XY Act § 10 xy% time saving
Change actor	Person(s) who perform the change.	List of change actors, teams	Internal: Department for XY Persons: Max Mustermann, Rainer Zufall
Lessons learned	Change experiences and assessment results of the implemented changes.	Documentation of lessons learned	The questionaire and reflection meetings served as "sensing tools" for... The initial process model A seemed to be modeled in a too complex way. A remodeling may support understandability... The XY analysis of the actions planned was particularly suitable for ...

Figure 1: Attributes of process model change.

of the process (a process model may have several versions during its lifecycle), and the change tracking graph A* which is discussed in more detail in the paragraph *Actions*.

Actions. Another attribute are change actions (or change instructions) which describe (a) what has to be changed (action planning, e.g. documented in a change notice by means of change patterns or primitives [WRRM08]) and (b) what change has been performed (action taking, e.g., documented in a change record by means of a table or graph). Before change actions are performed, they need to be planned, evaluated, and documented. Change of business process models can range from evolutionary-, and ad-hoc change, right up to business process redesign. Evolutionary change is often the result of reengineering efforts [vdAB02, p.2] and concerns the structural change of the process. The new version of the process model is relevant for all (new) process instances. Ad-hoc change becomes necessary when dealing with real-world exceptions (e.g. errors, rare events or special demands) and usually concerns single cases – process instances [WMC99] – or case groups [vdAB02, RRD03, RRD04]. The challenge of business process redesign is to come up with a new process design [MR05], e.g. from scratch or based on the radical improvement of current processes. In this work we focus on evolutionary change by considering process model versions. We understand change as a predefined set of changes (e.g. expressed by means of change patterns [WRRM08]) conducted to an already existing graph during a specific time frame. We use the *change tracking graph A** to visualize and document change actions conducted on the initial or current graph A in order to receive further process versions. The change tracking graph A* contains, as presented in our previous work

[KKRM11], the graph elements of the initial graph A that are not affected by the change, deleted, and added during change. We provide a set-based definition of the change tracking graph A* in order to explain the content of the graph.

Change Tracking Graph (CTG). Let P be the set of process graphs and A, A', A* $\in$ P. A=(N,E) is a directed graph with N defining the set of nodes, and E defines the set of edges, $E \subsetneqq N \times N$. Let A'=(N',E') be the adjusted graph after changing the initial graph A. σ is a predefined sequence of change patterns to be performed on the initial graph A. A [$\sigma\rangle$ A'. We denote A* := (N*,E*) as change tracking graph where:

- N* = $N \cup N_{add}$ = $(N \cap N') \cup (N_{add} \cup N_{del})$,
- E* = $E \cup E_{add}$ = $(E \cap E') \cup (E_{add} \cup E_{del})$,
- $N_{add} = N' \setminus N$, $N_{del} = N \setminus N'$, $E_{add} = E' \setminus E$, $E_{del} = E \setminus E'$.

In [KKRM11] we had further introduced three possible change tracking states of graph elements. Graph elements that are inserted are signed as $activated$ elements, deleted elements are denoted as $deactivated$ elements, and graph elements that are not affected by the change are marked as $initial$ elements.

Change Tracking States. Let the assumptions be as in Definition 1. Let T be a set of possible change tracking states with T:={deactivated, activated, initial}, and t $\in$ T. For A*=(N*,E*) we define a function *track* that assigns the corresponding change tracking state to each node and edge.
track: (N* $\cup$ E*) $\rightarrow$ T

$$\text{track(x)} := \begin{cases} deactivated, & \text{if } x \in N_{del} \cup E_{del}, \\ activated, & \text{if } x \in N_{add} \cup E_{add}, \\ initial, & \text{else (x} \in \text{(N} \cap \text{N')} \cup \text{(E} \cap \text{E'))}. \end{cases}$$

Change primitives. We could compute A, A', A* based on the set-based definition as presented above. However, as change is typically applied in an incremental manner, and we do actually know the change, we opt for determining the change tracking graphs by exploiting the changes. The basic requirement for this approach is that we can express all kinds of change patterns by means of the following change primitives [RRJK06]: 'delete node' (removing one node), 'delete edge' (removing one edge), 'insert node' (adding one node) and 'insert edge' (adding one edge). The following rules for calculating change tracking graphs hold:
- track(x) = initial [DELETE (x)$\rangle$ track (x) = deactivated
- track(x) = [INSERT (x)$\rangle$ track (x) = activated
for x $\in$ N' $\cup$ E', $\forall$ x $\in$ N $\cup$ E: track (x)

Using $insert$ and $delete$ allows to mark the change with particular visual properties without challenging the graph user with an exaggerated number of new visual appearances of the graph elements. All relevant change patterns as presented in [WRRM08] can be separated into these four change primitives. The graph elements are marked according to their change tracking state with the selected visual properties color, brightness, or size.

Visual Properties in CTG. Let Z be a set of possible visual properties for change, with Z := {Color, Size, Brightness} and z ∈ Z. For A*=(N*,E*) we define a function $visualize_c$ which assigns for each node and edge the corresponding visual property z ∈ Z.
$visualize_c$: $(N^* \cup E^*) \rightarrow$ (Color ∪ Size ∪ Brightness)

$$visualize_c\ (\mathrm{x}) := \begin{cases} C1, & \text{if } track(x) = \text{deactivated,} \\ C2, & \text{if } track(x) = \text{activated.} \end{cases}$$

and C1 ∈ Color $\Longrightarrow$ C2 ∈ Color (holds for Size Brightness resp.)

The change tracking graph is used in our design approach to document and highlight the changes implemented in the process model. The graph can be illustrated in two different layouts [KKRM11]. One layout highlights the past by preserving the layout of the initial graph A (mental map). The other layout reflects the future by adopting the layout of the adjusted process model, graph A', which is designed according to particular aesthetic criteria. Both layouts of the change tracking graph were considered in our approach.

Time. Time is an attribute of change that specifies the date or the time frame in which the change is performed. The time period of the different process model versions should be clearly represented. Time is also relevant to understand relationships between internal or external events (in the organizations's internal and external environment) as well as planned and implemented change actions.

Trigger. Triggers (or reasons) for change range from external factors – e.g., economic, legal, and technological developments – to more specific internal factors such as design errors, missing data sets and technical problems of workflow management systems used [vdAJ00]. Triggers may also refer to specific customer demands or rare events [vdAJ00]. In this work we consider triggers for evolutionary changes of process models.

Change actor. The person(s) who performed the change should be clearly represented and may include, for example, modelers and business process redesign teams.

Lessons learned. The assessment of the outcome (implemented changes) and the documentation of generated knowledge based on the change experience should as well be considered as a valuable attribute of change.

3 Methodology

Three different timelines were prepared for exemplary process models to represent the models' evolution along a time axis. The timelines were created by using three different timeline tools OfficeTimeline2010[1], TimelineStudio[2] and TimelineMaker[3]. It was decided that the best method to adopt for analyzing the timelines was expert inspection. Inspection is one of the usability methods that aim to find usability problems in a design [Nie94, RC08, Wil09]. The inspection was guided by the visualization requirements for process change which are presented in Section 4 and was performed by three inspectors

[1]http://www.officetimeline.com/ <accessed 15.06.2012 >
[2]http://www.timelinestudio.co.uk, trial version <accessed 15.06.2012 >
[3]http://www.timelinemaker.com, trial version <accessed 15.06.2012 >

individually. The inspection results were discussed during a consensus meeting. Based on findings of the inspection sessions and the attributes of process model change, the visualization concept for a timeline was developed that particularly focused on the support of process model change documentations. During the design phase, the mock-up was evaluated by means of *walkthrough* [Nie94, RC08] in order to discuss each change requirement and to identify problems with the design early in the process.

4 Change Visualization Requirements for Business Processes

For the comparison of the timeline tools (see Section 5) and for the design of our approach (see Section 6) we adapt the process change visualization requirements described in previous work [KRM12]: *Clear change representation*, *Visibility of Relationships between Versions*, *Different Views* and *Interaction*.

The requirements arose from a systematic literature review about characteristics of changes and visualization approaches in combination with a user survey in order to identify user's experiences and expectations with regard to change visualizations [KRM12]. The presented requirements primarily describe what a visualization should provide to present change information and can be used as measurable benchmarks for the evaluation of visualization approaches. To make the different timeline tools clearly comparable for our inspections, the following criteria are specified for each requirement.

Clear Change Representation. Timelines should consider all recorded changes of the process model in order to get a fast overview what was changed in the process model and which *change actions* (e.g., if a task is deleted or added in the process model) were performed. Sources as a description of process model or a link to the original files of the process models can be helpful to understand the logic behind the process models and possible effects and consequences. Furthermore, timelines should support an easy access to *additional change attributes* (e.g., actors who performed the changes, change triggers and lessons learned).

Visibility of Relationships between Versions. It is necessary to consider when changes were performed and how changes are related to each other in order to make versions more comprehensible and comparable in *process models*. The representation of versions with the help of timelines allows users to see the interconnections of versions along the *time* axis. However, it is often useful to represent the information not only as a *timeline* but also in *textual form* in order to make the graphical representation easier to understand or to present detail information [FS97, Kri11].

Different Views. The representation of timelines in combination with their process models and change information in a single view reduces the legibility and level of detail. Therefore it is necessary to consider strategies in order to deal with large quantities of information. Using *multiple views* is a possibility to simplify the design in such a way that different information is split into different views (see e.g., [GRF08, Kri11, Nor05]). The representation of timelines and their information in multiple views allows to analyze the change information from different perspectives and to compare complementary or contrasting information. For example, one view can present the timeline in combination with the change actions as milestones while another view can directly highlight the change actions in the process model. *Overview and detail* is a strategy that uses multiple views to provide an overview about the datasets and presents the corresponding detailed information when a user selects a data item from the overview [CMS99, Kri11, Nor05]. Overview and detail view are linked together and allow users a fast access to detailed information [CMS99, Kri11]. For example, if a user clicks on a specific part in the timeline, the detail information of the selected segment (e.g., change actions and who made the changes) is shown.

Interaction. Because of limited screen space, it is necessary to find strategies which support users to interact with the visualization in order to select the data items that should be presented. *Brushing and linking* is a strategy to interact with datasets in multiple views [Kri11, Nor05]. For example, if a user selects a data item in one view (brushing), this data item is then also selected in other views (linking) [Kri11, Nor01]. *Filtering* and *searching* allows users to find particular information depending on specific conditions or ranges in order to highlight specific information of interest [Kri11, YKSJ07]. For example, change information (such as change actions, change owner, and time span) can be filtered out. *Scrolling* and *panning* are further interaction strategies for navigating through the visualized information in case the screen is too small to present the whole information at once [Kri11, Spe07]. *Zooming* is the ability to zoom in and out within a viewport in order to magnify or minimize a particular area of the screen more closely (e.g., to resize documents or images). *Content zooming* is an effective strategy in order to jump from the overview to the detail information and vice versa. For visualization of timelines, content zoom can be used to change the scale of the time axis in order to switch between year, month, week, and day representation.

5 Timelines

We explore three widely used tools for timeline visualization, which are online available for free or as a trial version: Office Timeline, Timeline Studio and Timeline Maker. Office Timeline is an add-on for Power Point, thus, provides regular Power Point functionalities. For users familiar with regular Power Point is is therefore easy to learn. It offers a wizard to add timeline information and a variety of different visual timeline templates. Timeline Studio and Timeline Maker are two tools specifically designed to support the generation of timelines and manage detailed information belonging to timeline events.

5.1 Evaluation of Timelines

To determine the fit of the timeline tools with the specific purpose of visualizing process change, the selected three timeline tools were reviewed according to a coding-schema developed on basis of the requirements which are presented in Section 4. Three experts first scored the tools independently and then discussed ratings in a face-to-face meeting to reach a final, consensual rating. Figure 2 gives an overview of the results of the comparison of the timeline tools. As all tools offered a separate presentation and editing mode, the modes were independently reviewed. If not indicated otherwise in the table, presentation and editing mode got the same evaluation. Next, we discuss results concerning each requirement, in turn:

Clear Change Representation. Representation of change actions is possible in all three tools, mostly by adding text fields, in a notes files or by adding hyperlinks. Concerning the representation of additional change attributes in general, it is not possible to enter information in a structured way, but only as unstructured text. Therefore, consistency of the type of information added (as change triggers, change actors or lessons learned) is up to the editor. In Timeline Studio, users can see additional information in the detail view of a version (e.g., in description or note). Similarly, Timeline Maker offers notes which can be made visible in the presentation mode, too.

Visibility of Relationships between Versions. The support of process model visualization was quite weak in all three tools and almost no interaction with the process models is possible. In Office Timeline processes can be added as text (e.g., source code), as picture or as video for representing the change tracking graph, sound or links to URLs. The other tools also allow to link pictures and other types of files to specific events in the timeline, however, the processes are not visible in the same view as the timeline, but only in a separate window. Investigated timeline tools as a matter of fact support timeline visualization. Office Timeline offers a function to automatically create a table of milestones and intervals in chronological order based on the timeline. In Timeline Studio a textual description of the timeline information sorted in chronological order can be exported into Microsoft tools, but is not available in the tool itself. In TimelineMaker, the event entry as well as the chronology order view represent the timeline in tabular format (both part of the editing mode).

Different Views. In all three tools it is possible to see overview and detail information of the timelines. For instance, in the tool Office Timeline hyperlinks can be used in the presentation mode for detailed information. In the editing mode thumbnails show an overview of all slides on the left. However, there are no possibilities to show large timelines adequately. Timeline Studio also offers several views in the editing mode. The home view gives an overview about all process models and versions. Views on each process model present detailed information (e.g., about change information, or a change tracking graph). In addition, it is possible in the presentation mode to click on an event to see additional information (in the form of text, pictures, or links) in a new window. TimelineMaker presents overview and detail information in both, presentation and editing mode. In the presentation mode detail information about events is available on demand. It can be linked in form of files and text (place, notes, source), which are shown in a new window. In the editing mode, there also exist several views for overview and detail, e.g., thumbnails for overview. Concerning multiple views, Office Timeline uses this concept only in the editing mode (thumbnail, slideview, notes view). In Timeline Studio different views can also be found in the editing mode, when details about an event (e.g., images, notes, links and descriptions) are edited. TimelineMaker provides different views in several subwindows

Criteria	Office Timeline	Timeline Studio	Timeline Maker
Clear change representation			
Change actions	Yes	Yes	Yes
Additional change attributes	Yes	Yes	Yes
Visibility of relationships between versions			
Process models	Yes	Yes	Yes
Textual Representation	No	Yes	Only in Editing Mode
Timeline Representation	Yes	Yes	Yes
Different views			
Overview and detail	Yes	Yes	Yes
Multiple views	Only in Editing Mode	Yes	Only in Editing Mode
Interaction			
Brushing and linking	Only in Editing Mode	Only in Editing Mode	Only in Editing Mode
Filter	No	Yes	Yes
Scrolling	Only in Editing Mode	Yes	Yes
Panning	No	No	Only in Presentation Mode
Content zoom	No	Only in Editing Mode	Only in Editing Mode
Zooming	Only in Editing Mode	No	Yes
Search	Only in Editing Mode	Only in Editing Mode	Only in Editing Mode

Figure 2: Evaluation of timeline tools.

in the editing mode. In general, it is not surprising that the tools use different views more often in the editing mode than in the presentation mode, which usually tries to direct the viewers' attention on one piece of information at a time.

Interaction. Brushing and linking cannot be found in the presentation modes of the tools, but only in the editing modes. However, this concept is not directly used for interaction with the timeline, but to correctly associate detail and overview information (e.g., highlighting thumbnail of current slide on the left in Power Point). Filtering (e.g., of different time ranges and groups of processes) could only be found in the tool Timeline Studio, but not in the other tools. Scrolling is very common and could be found in all tools, while panning was only used by Timeline Maker in the presentation mode. The results concerning content zoom are interesting in so far as they demonstrate that the two time line tools Timeline Studio and Timeline Maker are targeting at dealing with the semantic information of timelines, and thus providing this option, while Office Timeline focuses on mere visualization of timeline information. Only Timeline Studio does not provide zooming. In contrast, search as a basic tool functionality is provided by all tools in the editing mode.

6 Design Concept

Based on the insights which we gained during our inspection sessions (see Section 5), we developed a visualization concept (see Figure 3 and Figure 4) in consideration of the requirements presented in Section 4.

Different Views. Multiple views are used in order to support the presentation of change information from different perspectives: timeline view, process model view and list view. The timeline view provides an overview of all process models' timelines. The juxtaposition of the different timelines allows users to compare the timelines with each other. After

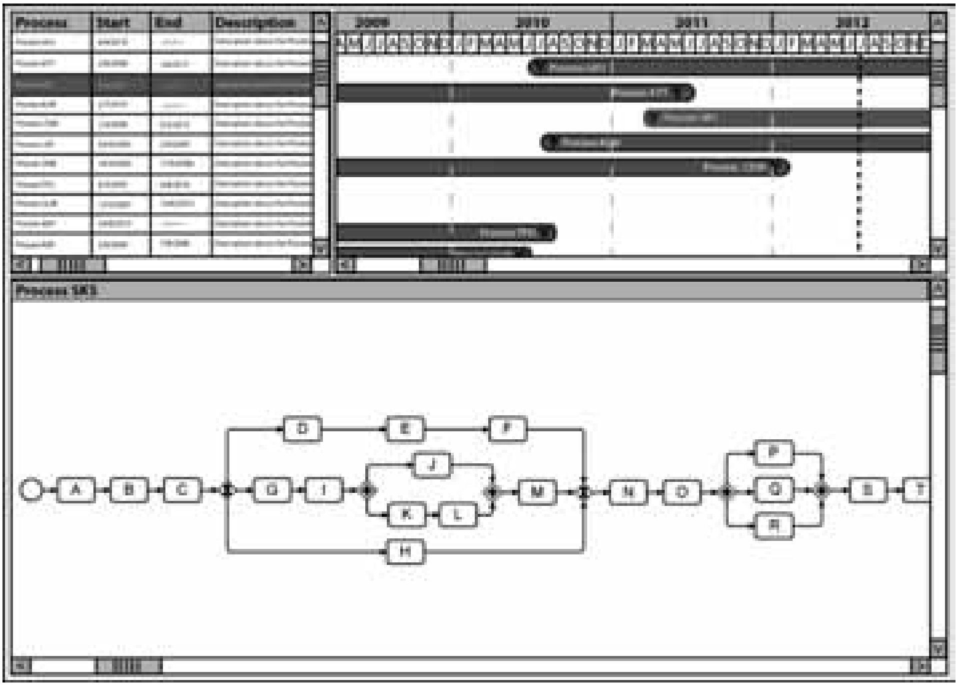


Figure 3: There exist three views: timeline view, list view and process model view. In this example the process model "Process SKS" is selected. Information (e.g., start date and end date) about "Process SKS" is presented in the list view and the corresponding process model is visualized in the process model view.

a timeline is selected, the list view presents the corresponding detailed information (e.g., start date, end date, description of the process model) and the process model is presented in the process model view (see Figure 3). The timeline is depicted as a rectangle to visualize the time span. The arrow symbol – located next to the process model name – allows users to expand and collapse change information for each process model in the timeline view (see Figure 4). In addition to expand or collapse the change information for each timeline separately, it is also possible to expand or collapse all timelines at once (e.g., via context menu). When users select the change information in the timeline, detailed information about the changes is presented in the list view and the changes are highlighted in combination with the corresponding process model in the process model view. In a previous study [KKRM11] we had evaluated two graph layouts for combining process models with change information (also called change tracking graph). As already mentioned in Section 2, one layout considered the mental map and the other layout optimized the graph according to some aesthetic criteria. The results of the evaluation had shown that users had different preferences. Thus, in our approach users have the choice how the change tracking graph is presented in the process model view. The example in Figure 4 visualizes the change tracking graph in consideration of the initial process model (mental map approach). A detailed description of the change visualization in the timeline view, list view, and process model view is presented in the subsection *Clear Change Representation*.

Interaction. The timeline view, process model view and list view are linked together via brushing and linking technique. It is possible to select the process model in the list view but also in the timeline view and the corresponding information about the selected process model is updated in all three views. For example, in Figure 3 the process model ”Process SKS” is selected in the timeline view and the corresponding entry is highlighted in the list view and the corresponding process model is presented in the process model view. Furthermore, search and filter functions are available (e.g., via context menu) in order to show only information that users want to analyze in detail. For example, users can filter out periods in timelines without any change information, to present only changes by a specific change owner or to show only the timelines within a specific time span. The search function is helpful, e.g., to find a specific process model or a specific change action within a process model. If a large amount of data has to be presented, scrolling and panning are available to navigate within the different views. Furthermore, users have the possibility to navigate between change actions with the help of two arrow symbols in the process view in order to go step by step along the timeline (see Figure 4). Moreover, a zooming function helps to magnify parts in the different views and content zooming allows to switch between month, week, and day time scale.

Clear Change Representation. In all three views, users can see which change actions were conducted. The information about change actions in the timeline view is available after the timeline has been expanded. The change actions are visualized as lines and the color reflects which kind of change actions (e.g. adding or deleting) were conducted. The line represents the point of time when the change was made and the thickness of the lines corresponds to the number of change actions for the specific time period. The example in Figure 4 shows the change actions for the process model ”Process GHJ” and ”Process SKS” in the timeline view. The time axis in this example splits the change actions into months. With content zooming it is possible to present all change actions, e.g., from a week or day for a detail analysis. Users can select the time unit (in Figure 4 months are used) or each change action separately. If users click the white space between two lines in the timeline, the process model version for this point in time is visualized in the process model view. The selected change actions are presented in the list view and are highlighted in the process model view. In all three views the same color code for the change actions is used. Although other visual properties (e.g., brightness or size) can also be used to highlight change actions in process models, one of our previous studies [KKRM11] showed that the usage of color to present changes was for most participants clear and preferred over other visual properties. Additional sources (e.g., description of a process model as separate text file) can be included in the visualization. If additional sources are available for a process model, a button in the process model view appears to open these sources in a separate window (see Figure 4). Moreover, additional change attributes (e.g., change owner) is presented in the list view. Furthermore, users have the possibility to rate and comment the conducted changes (see Figure 4).

Visibility of Relationships between Versions. The change information is available graphically in the timeline view and in the process model view as well as in textual form in the list view. The change actions are listed chronologically in the list view and in the time-

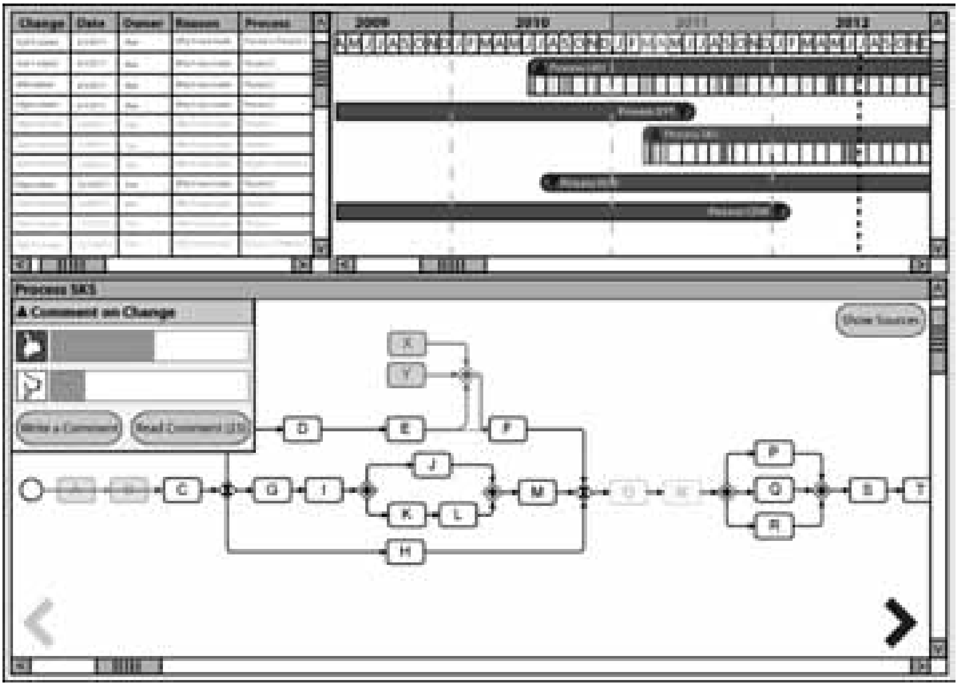


Figure 4: Change information can be expanded and collapsed for each timeline. Change information is available in all three views. Users can rate and comment changes in the process model view. In this example the like button is selected and the bar chart next to the rate buttons depicts the current voting result. Furthermore, users can read and write comments. The comment window can be expanded or collapsed on demand.

line view (see Figure 4). Furthermore, the list view also shows the dates when the changes were conducted. The actual date is presented as a line in the timeline view in order to show which process model is currently valid. Moreover, after filtering of or searching within a specific time span, the time span is additionally highlighted in the timeline view and only the timelines within the selected time span are visualized.

7 Conclusion

In this work we wanted to find out which attributes of change may be helpful for understanding and documenting changes and if timelines can help to document and visualize change of process models. We presented and discussed change attributes (such as triggers, process models, actions planned, evaluated and actually taken, time, actors and lessons learned) that support a detailed documentation of change. To find out to what extend current timelines can support the documentation of process model change, we first inspected three widely used tools for timeline visualization in regard to their fit with the change visualization requirements for business processes identified in previous work [KRM12] including the change attributes. Evaluation results showed that the requirements *interaction* and *clear change representation* offered room for improvement from the process model change documentation perspective. We further presented a first concept for visualizing and documenting process model change by means of timelines. The design concept for our approach was guided by relevant change visualization requirements and change attributes. The representation of the change performed in process models was influenced by the change tracking graph [KKRM11] as method to visualize and document change actions. We believe this work is a valuable initial step towards a holistic visualization solution for change documentations that address changes in and of process models.

References

[CMS99] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers Inc., 1999.

[DKN+10] F. Daniel, A. Koschmider, T. Nestler, M. Roy, and A. Namoun. Toward process mashups: Key ingredients and open research challenges. In *Proc. of the 3rd and 4th Int'l Workshop on Web APIs and Services Mashups*, pages 9:1–9:8. ACM Press, 2010.

[FS97] P. Faraday and A. Sutcliffe. Designing effective multimedia presentations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'97)*, pages 272–278. ACM Press, 1997.

[GRF08] T. M. Green, W. Ribarsky, and B. Fisher. Visual analytics for complex concepts using a human cognition model. In *Proc. of the IEEE Symposium on Visual Analytics Science and Technology (VAST'08)*, pages 91–98. IEEE, 2008.

[HDL68] S. Handel, C. B. DeSoto, and M. London. Reasoning and spatial representations. *Journal of Verbal Learning and Verbal Behavior*, 7(2):351–357, 1968.

[IBM05] IBM. IBM WebSphere Business Integration Handbook. Technical report, IBM, 2005.

[KG99] M. Kradolfer and A. Geppert. Dynamic workflow schema evolution based on workflow type versioning and workflow migration. In *Proc. of the IFCIS Int'l Conf on Cooperative Information Systems, 1999 (CoopIS'99)*, pages 104–114, 1999.

[KKRM11] S. Kabicher, S. Kriglstein, and S. Rinderle-Ma. Visual change tracking for business process models. In *Proc. of the 30th Int'l Conf on Conceptual Modeling (ER'11)*, pages 504–513. Springer, 2011.

[Kri11] S. Kriglstein. *Human Centered Visualization of Ontologies (Doctoral thesis).* PhD thesis, University of Vienna, 2011.

[KRM12] S. Kriglstein and S. Rinderle-Ma. Change Visualization in Business Processes - Requirements Analysis. In *Proc. of the Int'l Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications (GRAPP/IVAPP)*. SciTePress, 2012.

[MR05] S. L. Mansar and H. A. Reijers. Best practices in business process redesign: Validation of a redesign framework. *Computers in Industry*, 56(5):457 – 471, 2005.

[Nie94] J. Nielsen. *Usabiltiy Inspection Methods.* John Wiley and Sons, 1994.

[Nor01] C. North. Multiple views and tight coupling in visualization: A language, taxonomy, and system. In *Proc. of the CSREA CISST 2001 Workshop on Fundamental Issues in Visualization*, pages 626–632, 2001.

[Nor05] C. North. Information Visualization. In G. Salvendy, editor, *Handbook of Human Factors and Ergonomics*, pages 1222–1246. John Wiley & Sons, 2005.

[RC08] J. Rubin and D. Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests (2nd Edition).* Wiley Publ., 2008.

[RRD03] M. Reichert, S. Rinderle, and P. Dadam. On the Common Support of Workflow Type and Instance Changes under Correctness Constraints. In *Proc. 11th Int'l Conf. Cooperative Information Systems (CooplS '03)*, pages 407–425. Springer, 2003.

[RRD04] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systemsa survey. *Data and Knowledge Engineering*, 50(1):9 – 34, 2004.

[RRJK06] S. Rinderle, M. Reichert, M. Jurisch, and U. Kreher. On representing, purging, and utilizing change logs in process management systems. In *Proc. of the 4th Int'l Conf on Business Process Management (BPM'06)*, pages 241–256. Springer, 2006.

[Spe07] R. Spence. *Information Visualization: Design for Interaction (2nd Edition).* Prentice-Hall, Inc., 2007.

[TKW91] B. Tversky, S. Kugelmass, and A. Winter. Cross-cultural and developmental trends in graphic productions. *Cognitive Psychology*, 23(4):515–557, 1991.

[vdAB02] W.M.P van der Aalst and T. Basten. Inheritance of workflows: an approach to tackling problems related to change. *Theoretical Computer Science*, 270(12):125 – 203, 2002.

[vdAJ00] W.M.P. van der Aalst and S. Jablonski. Dealing with Workflow Change: Identification of Issues an Solutions. *Int'l Journal of Comp. Systems, Science and Engineering*, 15(5):267–276, 2000.

[Wes07] M. Weske. *Business Process Management. Concepts, Languages, Architectures.* Springer, 2007.

[Wil09] C. Wilson. *User Experience Re-Mastered: Your Guide to Getting the Right Design.* Morgan Kaufmann, 2009.

[WMC99] C. Workflow Management Coalition. Workflow Management Coalition - Terminology and Glossary, Document Number WFMC-TC-1011, Document Status - Issue 3.0. Technical report, Workflow Management Coalition, 1999.

[WRRM08] B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - Enhancing flexibility in process-aware information systems. *Data Knowl. Eng.*, 66:438–466, 2008.

[YKSJ07] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[ZL07] X. Zhao and C. Liu. Version management in the business process change context. In *Proc. of the 5th Int'l Conf on Business Process Management (BPM'07)*, pages 198–213. Springer, 2007.

How to Choose the Right BPM Tool: A Maturity-Centric Decision Framework with a Case Evaluation in the European Market

Christopher Hahn[1] Till J. Winkler[2] Fabian Friedrich[2]
Gerrit Tamm[3] Konstantin Petruch[4]

[1]Technical University of Berlin, Strasse des 17. Juni 135, 10623 Berlin, Germany
christopher.hahn@tu-berlin.de
[2]Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
till.winkler@wiwi.hu-berlin.de
fabian.friedrich@alumni.hu-berlin.de

[3]SRH University Berlin, Ernst-Reuter-Platz 10587 Berlin, Germany
gerrit.tamm@srh-uni-berlin.de

[4]Deutsche Telekom AG, T-Online-Allee 1, 64295 Darmstadt, Germany
k.petruch@telekom.de

Abstract: The enabling role of technology for effective business process management (BPM) is not being doubted. However, finding the right tool that suits a company's specific requirements is usually a challenging task. This paper presents a novel decision framework for the critical assessment of BPM tools which maps company requirements to different levels of BPM maturity and thus aims to be applicable in various organizational contexts. The framework includes emerging BPM features such as sophisticated process simulation capabilities and the support of common IT reference models and is complemented by a decision model which provides for complex preferences and uncertainty throughout the assessment process. We demonstrate the applicability of the proposed artefact by the case of a tool selection at a major telecommunications company and a survey-based analysis of 19 BPM tool vendors in the European market.

1 Introduction

Software tools are essential for effective Business Process Management (BPM), since they enable the design, enactment, management, and analysis of operational business processes [vdAHW03a]. By now, the number of different BPM tools is estimated to have grown to more than 300 products available on the market [KM05, p. 403]. As firms differ in their specific requirements, finding and choosing the right tool can become a time consuming and cumbersome procedure.

An important application area for BPM tools lies in IT management itself [Has07]. In the course of IT industrialization, IT services are increasingly commoditized, demanding a higher quality and a dynamic management of the underlying IT processes. This is also reflected in the evolution of common IT Service Management and IT Governance frameworks such as ITIL and COBIT [Ins07, CS08]. Likewise, process simulation capabilities play an increasingly important role allowing to optimize such IT production processes by providing a quantitatively supported choice of the best design [JVN06]. The rather small body of literature on BPM tool selection has largely fallen short of considering these aspects and the practical issues of choosing a BPM tool. This paper proposes a maturity-centric decision framework for the critical assessment of BPM tools, which aims to be applied in business and IT practice.

The remainder is structured as follows: Section 2 reviews related work on BPM tool selection and formalizes the foundations of a decision model. Section 3 describes the proposed framework including preference scenarios, assessment criteria and an approach for dealing with uncertain vendor information. In section 4 the proposed artifact is evaluated by the requirements of a major Telcommunications company and a market study with vendors in the European market. Section 5 concludes the evaluation and points out limitations and future work.

2 Related Work

2.1 BPM Tool Selection

Throughout this work we understand BPM tools synonymously with Business Process Management Systems (BPMS) as any packaged software which is able to support the distinct activities in the business process management life-cycle [SW08b, vdAHW03b]. Non-academic press and research institutions such as Gartner and Forrester regularly release reviews on BPM tools, e.g. in [HCKP09, Vol08, McG09, WD08, SW08c], which shows the relevance of this topic. Such studies usually evaluate a number of tools[1] for a broad range of functional and non-funtional criteria and, therefore, provide a good overview of available tools on the market. However, these evaluations often have a focus on rather technical criteria and suggest that decisions are always objective, inasmuch as they cannot take into account the individual requirements of different types of BPM initiatives [DR10].

In academic literature, four major functionality clusters for BPM tools have been emphasized to a varying extent: Design (process analysis, modelling and graphical representation), execution (implementation, enactment and processes automation), analysis (case data extraction, monitoring, mining and visualization), and simulation (what-if analyses, process comparison, optimisation and re-design). For example, Jansen-Vullers and Netjes [JVN06] perform a qualitative evaluation of six tools with a focus on simulation capabilities. Bosilj-Vuksic et al. [BVCH07] propose an extensive assessment framework with 70 criteria focusing on software packages in the context of business process change (i.e. design and execution functionality). Yet, these works do not demonstrate how to perform such assessment. The evaluation by Scheithauter and Wirtz [SW08a] covers 23 criteria

[1] The number of evaluated tools in [HCKP09, Vol08, McG09, WD08, SW08c] ranges from 7 to 22

clustered into the three layers: business, integration and execution. In [DR10] the case of a BPM tool selection in an Australian government agency is reported, where 10 products from major vendors were evaluated using a weighted scoring model with 47 criteria grouped into six main categories.

Altogether, academic literature focuses on specific functionality clusters of BPM tools and covers a comparably small fraction of tools available on the market. These frameworks use rather technical criteria and do, if at all, only implicitly take into account organisational properties such as maturity. Also, to the knowledge of the authors, there is currently no research which considers emerging BPM tool requirements such as support of common IT reference frameworks.

2.2 Model Foundations

Software tool selection can be regarded as a problem of multi-criteria decision making (MCDM). From a set of alternative choices a_i $(i = 1, \ldots, I)$ the best one is to be chosen based on a number of criteria c_j $(j = 1, \ldots, J)$. Every criterion can take different values $x_{ij} \in \mathbb{D}_j$, one for each alternative choice, which may possess nominal, ordinal and cardinal scales making them difficult to compare. Therefore, they are mapped to score values $u_{ij} \in S \subset \mathbb{R}$ by a utility function $U_j : x_{ij} \rightarrow u_{ij}$ representing the singular utility of x_{ij} for a decision maker. To come to an overall decision, each utility vector u_i is aggregated to a scalar value v_i by a value function $V : (u_{i1}, \ldots, u_{iJ}) \rightarrow v_i$. Preferences can be represented by weights w_j for each singular utility. Using a common additive value function [Ste96], the overall value for an alternative is given by Eq. 1 (left side).

To determine weights w_j, a combination of MCDM models with the Analytic Hierarchy Process (AHP) was identified to be an adequate technique. Saaty [Saa80] introduced the AHP as an integrated approach for decision making in socio-economic problems. Following the AHP, a matrix of pairwise comparisons $A = (a_{mn}) \in \mathbb{R}^{J \times J}$ is defined for the decision criteria c_j according to Eq. 2 (right side).

$$v_i = V(u_i) = \sum_{j=1}^{J} w_j U_j(x_{ij}) = \sum_{j=1}^{J} w_j u_{ij}| \tag{1}$$

$$a_{mn} \begin{cases} > 1, & \text{if } c_m \text{ more important than } c_n \\ < 1, & \text{if } c_m \text{ less important than } c_n \\ = 1, & \text{if indifference between } c_m \text{ and } c_n \end{cases} \tag{2}$$

The reciprocal values $a_{nm} = 1/a_{mn}$ can be calculated accordingly. Then, the estimated weights w can be obtained by the eigenvalue technique $(A - \lambda I) = 0$ where λ is the largest eigenvalue and I is the identity matrix [Saa80]. The advantage of this procedure is that the arbitrary element of distributing weightings is simplified to a pairwise comparison of different aspects, which reduces subjectivity.

Further, each criterion may be affected by uncertainty, particularly in the case of subjective vendor information. Thus it can be assumed that $U_j : x_{ij} \rightarrow u_{ij}$ is only valid with a certain probability p, i.e. u can be regarded as a random variable with a density $f_{ij}(u_{ij}) = p(u = u_j)$ for the discrete case. The stochastic influences in u_{ij} are also passed on to v_i. Instead of a scalar value v_i we deal with a density function $g_i(v|u_{i1}, \ldots, u_{iJ})$, $v \in V(S^J)$. In case of an additive value function and independence between values u_{ij}, g_i is the convolution of utility densities transformed by their respective weight [BP80]. An appropriate measurement to select the best alternative is the expected value for v given by $E(v|g_i) = \int vg(v|u_{i1}, \ldots, u_{iJ})\, dv$, provided the decider is risk neutral. Accordingly, for a risk avert or risk seeking decider, an expected total utility value needs to be formed, see [Fis70].

3 Decision Framework

The proposed decision framework builds on probabilistic MCDM and AHP method presented above and consists of preference scenarios, assessment criteria and an approach for evaluating uncertainty.

3.1 Preference Scenarios

As BPM initiatives may vary in their level of skills and experience [RdB05], we define six scenarios to reflect the particular preferences which firms, respectively particular stakeholders within a BPM initiative may have. Roseman and De Bruin [RdB05] introduced a BPM maturity model comprising six factors (Strategic Alignment, Governance, Method, IT/IS, People and Culture) resulting in different stages of BPM maturity. We consider such stages and propose specific scenarios for low, medium, and highly mature organizations. Further, we define three scenarios representing the preferences of decision-makers who are not aware of their current maturity level or possess different preferences such as service and support or cost. The scenarios can be briefly described as follows.

- *Low Maturity Scenario:* At this stage, the focus lies on the analysis and design of process models. Low maturity organizations will require a tool mainly for capturing processes and making them usable for the employees. Therefore, support of training or staff is important at this stage. The organization also benefits from available reference models which can be used and adapted.
- *Medium Maturity Scenario:* Based on existing process models, organizations at this stage seek a deeper understanding of the relationship between processes. Their focus shifts to monitoring and evaluation with the help of key measures which relate to performance aspects of IT Governance.
- *High Maturity Scenario:* In this scenario the handling of key measures becomes more important. High maturity organizations require monitoring of real time data,

which can be used for detailed reporting, bottleneck detection and ex-ante simulation. This enables immediate event triggering and allows an organization to instantaneously react and determine counteractions.

- *General Scenario:* This is a baseline scenario assuming an organization that has no particular preferences towards a BPM tool. Thus, in this scenario all criteria are to be weighted equally.
- *Service & Support Scenario:* Here the implementing company puts emphasis on the support and service that the vendor is able to provide, looking for strong and reliable partner. Smaller or less experienced organizations may prefer this scenario as they depend stronger from external know-how.
- *Cost Sensitive Scenario:* This scenario assumes a very cost-sensitive company. Preferences in this scenario will be distributed equally between all criteria which are not cost-related.

3.2 Categories and Criteria

Based on the preference scenarios, we introduce six categories correlating with scenario names to structure the proposed assessment criteria. Clustering the criteria this way allows the definition of preferences for each scenario on a category level and reduces effort for defining appropriate weights, provided that preferences within each category stay constant across different scenarios. The categories are indexed by letters: Low Maturity Level Requirements (L), Medium Maturity Level Requirements (M), High Maturity Level Requirements (H), General Functionality (G), Service & Support (S), and Costs (C).

In software selection, functional, non-functional as well as vendor related criteria are relevant [KB99]. We mix functional and non-functional criteria within the categories L, M, H to reflect the combined requirements on each of the maturity levels. In contrast, category G contains aspects which do not correlate with BPM maturity, for instance modelling possibilities, model reuse and multi-user characteristics. Cluster S (Service & Support) comprises criteria that provide an indicator for the reliability of a vendor, as unforeseen market disappearance may cause great financial damage. Category C (Costs) captures several cost-related aspects in the life-cycle of a BPM initiative, including hardware and software requirements. To balance the effects of recurring and one-time costs, we assumed a usage of the tool by 10 people over a duration of 5 years in the later evaluation.

Detailed criteria have been defined based on existing literature (as presented Section 2.1) and iterated in a number of expert interviews. As an expert we considered two representatives of the given case company, a university professor as well as a representative from a tool vendor who would not participate in the evaluation. Further, for each of the 58 criteria appropriate ordinal scales $\mathbb{D}_j$ have been defined and mapped to utility scores $u_j \in S = \{0, \dots, 4\}$, where zero represents the lowest and four the highest utility. Short descriptions of the criteria are listed in Tables 1 and 2, respective scales have been omitted for brevity.

Table 1: Maturity Level Criteria

Low Maturity Level Requirements	
L.1	Capability to display process models (e.g. in a web portal).
L.2	Extent of Vendor's offering for training
L.3	No. of Partner Consultants distributing the tool.
L.4	Availability of ITIL v2 reference model.
L.5	Availability of ITIL v3 reference model.
L.6	Availability of COBIT reference model.
L.7	Capability to assign Roles and responsibilities to process models.
L.8	Ability to simulate a process.
L.9	Existing project experience of the firm.
L.10	No. of employees with an IT Governance certificate.

Medium Maturity Level Requirements	
M.1	Capability to indicate process relations in a hierarchy.
M.2	Features to collaborate on process model design.
M.3	Capability to report about key measures.
M.4	No. interfaces to operational systems to extract data.
M.5	Availability of predefined ITIL key measures.
M.6	Availability of predefined COBIT key measures.
M.7	Capability to model risks (in process model).
M.8	Capability to simulate processes based on operational data.
M.9	Ability to define a distribution function for the simulation.
M.10	Activity based cost calculation capability.
M.11	Ability to define key measures.
M.12	Capability to do process mining.
M.13	No. of realized projects with an IT Governance focus.

High Maturity Level Requirements	
H.1	Ability to simulate processes in advance.
H.2	Ability to animate process simulation graphically.
H.3	Capability to estimate distributions based on certain data.
H.4	Capability to extract real time data from operational systems.
H.5	Ability to report real time data.
H.6	Key Measures can be arranged in a hierarchy.
H.7	Definition of affection between two key measures.

Table 2: General Criteria

General Functionality	
G.1	Support of the Unified Modeling Language (UML).
G.2	Support of the Business Process Modeling Notation (BPMN).
G.3	Support of other modeling notations such as EPC or the ability to extend the meta-model.
G.4	Capability to import existing models from other tools or XML (e.g. XPDL).
G.5	Capability to export existing models to other formats such as XML (e.g. XPDL).
G.6	Ability to automatically layout model elements (e.g. hierarchical or radial).
G.7	Ability to create different models, e.g. from organization or data perspective
G.8	Support of simultaneous users.
G.9	Capability to define user rights and role definition.
G.10	Support of version control system for models.
G.11	Ability to store data and information in central repository.
G.12	Ability to build and maintain a glossary or data dictionary.

Service and Support	
S.1	Offering of online support.
S.2	Offering of phone support.
S.3	Vendor or tool has won awards or obtained certifications.
S.4	Vendor provides service level agreements (SLAs).
S.5	The age of the vendor.
S.6	The age of the tool.
S.7	Number of the vendor's employees.
S.8	Total vendor's revenue in 2008.
S.9	Vendor offers customization possibilities?

Costs	
C.1	*Client Hardware Requirements:* Requirements for the client software to run.
C.2	*Server Hardware Requirements:* Required hardware for the server component.
C.3	*Tool & User License:* Acquisition cost for the tool and user license cost.
C.4	*Support Costs:* Costs that are charged for support per year.
C.5	*Training Costs:* Costs that are charged for in-house training per day.

Table 3: Uncertainty Assessment

Level	σ^2-value	Description
Low	0.2	No uncertainty at all, clear answer given consistent with prior information
Medium	1.2	Medium level of uncertainty, answer given unclearly or qualified reasons of doubt
High	2.0	High level of uncertainty, no answer given at all or the question is obviously not answered the right way.

3.3 Modeling Uncertainty

In order to deal with uncertain and potentially incomplete vendor information, the singular utility of every criterion is modeled to be normally distributed. This is a assumption regarding the underlying random variables. However, a normal distribution appears particularly suitable, because it is theoretically well understood and approximates well many real-life phenomena [LMS06, p. 961]. Given the presented additive value function (Eq. 1) and assuming stochastic independence between criteria values, we can take advantage of the resulting relationship between utility and value distributions [BP80], as displayed in Eq. 3.

$$u_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2) \Rightarrow v_i \sim \mathcal{N}(\sum_{j=1}^{J} w_j \mu_{ij}, \sum_{j=1}^{J} w_j^2 \sigma_{ij}^2) \quad (3)$$

The uncertainty connected to a value x_{ij} of a criterion is represented in the variance of its utility σ_{ij}^2. To determine an appropriate variance, three levels of uncertainty are defined depending on the quality of vendor information available, see Table 3. For example, for a singular utility distributed with $\mu_{ij} = 2$ and a *high* variance of $\sigma_{ij}^2 = 2.0$, u_{ij} falls in a confidence interval within the standard deviation of $[\mu_{ij} \pm \sigma_{ij}] = [0.6, 3.4]$ with a 68% probability, whereas for lower uncertainty levels this interval is much smaller. This way, the total variance of the value distribution $\sigma_i^2 = \sum_{j=1}^{J} w_j^2 \sigma_{ij}^2$ is a good indicator for the overall (un-)certainty in the assessment of choice a_i.

4 Case Evaluation

For the evaluation of our approach, we use a single observational case study in which we focus on the applicability and the organizational benefits of our framework.

4.1 Case Introduction

The case example refers to a BPM initiative at the department for IT production at a major telecommunications company. This department comprises about 40 employees and has the mission to develop and operate the platforms for most of the company´s end-user content offerings (such as online, mobile and TV-based entertainment portals). The department

Table 4: Weightings per Scenario (in %).

Category	General Scenario	Low Maturity Sc.	Medium Maturity Sc.	High Maturity Sc.	Service & Support Sc.	Cost Sensitive Sc.
General Functionality	16,7	17,6	16,7	13,9	18,4	18,8
Low Maturity Req.	16,7	33,5	11,4	13,9	17,1	18,8
Medium Maturity Req.	16,7	8,6	32,2	13,9	13,7	11,2
High Maturity Req.	16,7	8,6	13,7	35,6	9,0	8,4
Service & Support	16,7	19,4	7,9	7,9	27,9	10,7
Costs	16,7	12,4	14,8	14,8	13,9	32,2

acts as an internal shared service provider to internal departments and as a buyer from external parties likewise (e.g. for media content, payment services, geographical information, etc.). External providers have to fulfil quality criteria based on agreed performance indicators. The current paramount challenge is the development and usage of a governance model for the operation of both, internal and external IT services. Most of the IT service processes are related to ITIL and COBIT IT Management Frameworks [Ins07]. As a logic consequence, the company was seeking for a highly sophisticated BPM tool which integrates two aspects into one: Management of business processes and management of governance processes. The management has already put considerable effort into continually improving ITSM quality in order to achieve highest levels in common maturity frameworks. Hence, the department is aiming towards the automation of most management processes and the support of certain optimisation routines and therefore set up a BPM initiative for selecting and introducing a dedicated tool.

4.2 Preference Weighting

During the tool selection process we were able to apply and further refine the decision framework presented above. Successful introduction of a new tool demands not only the functional fit to the requirements, but also the acceptance of the tool by decision bodies and key users. Due to complex organizational structures, the requirements for a BPM tool and their importance differed considerably between the parties involved. The AHP was applied to derive the weightings w_i on a category and criteria level as described in section 2.2. In the given case, we dealt with multiple decision makers: the department head, the members of the application group as well as the IT controller. Therefore, the pairwise comparisons were performed with the former and later reviewed with all other involved parties. For example, in the cost-sensitive scenario the costs-category was considered to be 2 times as important as general functionality and 4 times as important as high maturity level requirements, resulting in its final predominance. To compute the eigenvectors of the resulting 12 pairwise comparison matrices (one for category preferences within each scenario and one for preferences within each category), a simple power iteration algorithm was applied which constantly converged after one iteration. Table 4 shows the resulting weightings of each category for each scenario.

Table 5: Short List of Vendors and Tools (* indicates participating vendors)

No.	Vendor name	Tool name	No.	Vendor name	Tool name
1	Binner IMS*	Sycat Process Designer & Analyzer	17	Lombardi*	Lombardi Teamworks
2	BOC*	ADONIS	18	MEGA	MEGA Process
3	Casewise	Casewise	19	Metastorm	Metastorm Enterprise
4	Consideo*	Consideo	20	MID*	Innovator Business
5	Cordys*	Business Operations Platform v4	21	Oracle*	Oracle BPA & BAM
6	EMC	EMC BPMS	22	Pavone*	Expresso Workflow
7	Fraunhofer IPK	Mo2GO	23	Pegasystems	Smart BPM Suite
8	Fujitsu	Interstage BPM	24	Pulinco	TopEASE
9	IBE*	Pace2008	25	Semtation*	SemTalk
10	IBM	BPMS	26	Signavio*	Signavio
11	IDS Scheer	ARIS Platform	27	Software AG*	webMethods BPMS
12	iGrafx*	iGrafx Enterprise Modeler	28	Soreco*	Xpert.ivy
13	IMG / S&T*	Promet@work	29	Synlogic*	Income Suite
14	Intalio	Intalio BPM Enterprise Edition	30	Tibco Software	Tibco iProcess
15	Intellior	AENEIS	31	Ultimus*	Adaptive BPMS
16	Inubit*	inubit BPM Suite	32	ViCon*	ViFlow

4.3 Vendor Assessment

The proposed framework was then used in course of the vendor assessment. First, we assisted in reviewing related market studies [HCKP09, Vol08, McG09, WD08, SW08c] and academic literature [DR10, vDdMV^{+}05] to identify candidate tools, which resulted in a long list of 48 vendors. Among these vendors we found both, small specialised businesses serving local customers as well as large providers which already serve the international market with a wide variety of tools and services. The proposed framework was then used to conduct a vendor assessment.

Based on the requirements of the case company, there were two important exclusion criteria for tools and vendors. Firstly, to ensure comparability of regulatory backgrounds and to reduce communication barriers, only vendors with a headquarter or subsidiary in a European country were considered. Secondly, only tools with the general ability to simulate processes were included, to ensure that at least a minimum of required functionalities are fulfilled. Table 5 gives an overview of the short-listed vendors and their offered BPM solutions.

To prepare the vendor assessment, the assessment framework was converted to a structured interview questionnaire. Each assessment criterion was turned into a concise open-ended question concealing the underlying valuation logic. By the domain knowledge of the interviewers, the answer could then be coded as an expected score value μ_{ij} with an uncertainty level σ_{ij} of the singular utility distribution. As proposed by Hunt et al. [HSW82], the questionnaire was pre-tested iteratively with the above mentioned BPM experts by the method of identifying defects in questions and rating on the comprehensibility. Questions have been logically re-ordered by topics (instead of categories) to improve understanding of each question by its context and hide the aggregation logic.

Short-listed vendors were contacted via telephone and asked to participate in the survey. 11 vendors were able to complete the survey in a telephone interview, 3 vendors gave partial information on the telephone and handed in missing information later, and 5 vendors preferred to answer via mail in a fully self-administered way. The response time differed widely from immediate interviews up to filled questionnaires after several weeks. 13 vendors did not participate or missed the deadline for handing in missing information resulting in an overall response rate of 59%. Age of the participating companies ranged from 1 year to more than 10 years (mean: 7, median: 7-10) and number of employees ranged from below 10 to more than 500 (mean: 247, median: 200-500) respectively.

Subsequently, all survey information was evaluated according to the proposed decision model. Telephone interviews were recorded which allowed for double-checking of the assessment. Utility values μ_{ij} and uncertainty levels σ_{ij} have been assigned independently by two coders and discussed in case of intercoder differences. We rated missing answers with a high uncertainty and tried to carefully draw a conclusions from the present data if no or only vague data was provided.The aggregated values for utility and variance that were allocated to each tool vendor are shown in Table 6. For reasons of confidentiality and brevity, vendors have been anonymized by alphabetical letters and rows 8 to 16 have been left out.

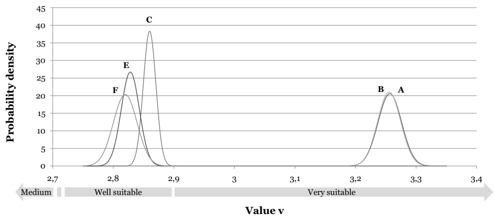
Table 6: Results of the Vendor Assessment. (Variance values to a factor 10^2)

Rank	General Scenario			Low Maturity Scenario			Medium Maturity Scenario			High Maturity Scenario			Service & Support Scenario			Cost Scenario		
	i	μ_i	σ_i^2	i	μ_i	σ_i^2	i	μ_i	σ_i^2	i	μ_i	σ_i^2	i	μ_i	σ_i^2	i	μ_i	σ_i^2
1	A	3.21	2.10	A	3.29	2.13	A	3.20	1.65	A	3.26	1.94	A	3.22	1.82	A	3.06	5.97
2	B	3.14	2.06	C	3.12	1.62	B	3.17	1.52	B	3.26	1.90	B	3.14	1.71	D	2.93	6.42
3	C	2.93	1.04	E	3.09	1.15	C	2.98	1.23	C	2.86	1.04	C	3.04	1.11	B	2.90	6.37
4	D	2.91	2.64	B	3.08	1.74	D	2.97	2.42	E	2.83	1.50	D	3.00	2.57	E	2.79	1.56
5	E	2.88	1.00	D	2.88	3.09	E	2.88	1.06	F	2.82	1.96	E	2.92	0.92	M	2.76	6.16

4.4 Case Results

The results of the assessment suggest that firm A has a leading position for all scenarios. Yet, other vendors like B, C, D and E are also often among top positions. This indicates that for the given case, one of these products is most likely to fulfill the departments BPM initiative. In order to better interpret these values, we estimated the 20%-quantiles of the resulting normal distribution of v across all tools and mapped these intervals to an ordinale 5-point scale *Very suitable, Well suitable, Medium suitable* and so on.

Table 6 also shows the variance which was factored into our model. In the high maturity scenario for example, where μ_A and μ_B only differ insignificantly, a risk averse decision-maker would opt for vendor B using the σ^2-metric an additional decision criterion. The tradeoff between expected value and information quality becomes clearer by looking at Fig. 1. Although variances of tool A and B are much higher than for tool C, it is still extremely improbable that tool C could actually be a better choice than A or B.



Quantile	Interval	Valuation
$q_{0.2}$	$[-\infty; 2.25]$	Not suitable
$q_{0.4}$	$[2.25; 2.47]$	Less suitable
$q_{0.6}$	$[2.47; 2, 67]$	Medium suitable
$q_{0.8}$	$[2.67; 2.90]$	Well suit-able
$q_{1.00}$	$[2.90; \infty]$	Very suit-able

Figure 1: Value Distributions for High Maturity Scenario and valuation Intervals.

In the given case, the utilization of the decision framework brought about several benefits. A crucial feature has been the ability to provide transparency regarding the correlation of the requirements and the derived tool recommendation, which has been very helpful in the communication with the involved parties. For example, a controller found his preferences represented in the cost-sensitive scenario, while a member of the application group primarily looked at the results in the high-maturity scenario. Thus, the decision framework helped to understand different viewpoints and dependencies between evaluation criteria so that communication was no longer focused only on group-specific requirement sets. A further advantage of this decision framework was the in-depth consideration of innovative criteria such as automation aspects and the inclusion of simulation capabilities in a differentiated manner by maturity-oriented clustering. This is of particular importance in a highly mature environment like telecommunications, where sophisticated simulation capabilities are required. Finally, the application of the decision framework supported the overall assessment of vendor and tool characteristics. As a result, only vendors of tool A and B have been chosen for further on-site evaluations. Besides, the framework and decision model has proven to be highly practicable and easy to use through implementation in a spreadsheet-like format.

4.5 Market Findings

As a byproduct of the empirical evaluation, some statements about the BPM market in general can be derived. To conduct a broader analysis, mean score values $\overline{u}_j$ were computed across different alternative tools. Those values that lie outside of a range $[1, 3.5]$ are listed in Table 7. First we find that all tools provide a way to organize processes hierarchically. Also, multiuser support is provided by almost all products. Nearly all vendors offer an online help desk including FAQ and phone support. Interestingly, online solutions are sometimes even better supported, which is why their score is slightly higher. On the downside, only few vendors make reference model support an integral part of their product. Those that integrated a reference model tend to support the second version, as version 3 has just been recently released. Another finding is that COBIT does not seem to be recognized as important as ITIL. In our study, only one vendor provides a full COBIT reference implementation. The same applies to methods of process mining which have received much attention in academia (e.g. [vdAW04]), but are hardly implemented in commercial tools yet.

Table 7: Mean Scores for Selected Criteria.

	Criterion	$\overline{u}_j$		Criterion	$\overline{u}_j$
M.1	Process Hierarchy	4.00	L.4	ITIL v2 Reference Model	1.67
G.8	Multi User Support	3.56	L.5	ITIL v3 Reference Model	0.89
S.1	Online Help Desk	3.83	L.6	COBIT Reference Model	0.33
S.2	Phone Help Desk	3.78	M.13	Process Mining	0.50

4.6 Managerial Implications

The proposed framework for BPM tool selection presents an approach that is based on widely recognized methods and easy to understand. Therefore, we consider it as a pragmatic, yet powerful tool, which, from our point of view, may assist BPM practitioners in several ways.

First, the proposed methodology including its assumptions can be used as a guidance in case of the same field of application. Second, the framework can easily be extended or adjusted if e.g. requirements are missing or weightings need to be revised. Third, our approach helps practitioners in providing a structure for various tool requirements that have to be mapped to business requirements and simultaneously considering the maturity with respect to BPM. As a consequence, time and cost for developing own methodologies can be reduced, and instead be focused on an in-depth analysis of crucial tool features. Furthermore, a transparent selection framework allows for enhanced communication on certain tool aspects and their importance, respectively. Hence, a justification for a specific vendor decision can be done credibly. At last, encompassing the uncertainty will help the assessing organization to challenge reliability and validity of given information.

5 Conclusion

5.1 Summary

In this paper we proposed a novel decision framework for the assessment of BPM tools, which incorporates different maturity scenarios and thus accounts specific clusters of requirements which are typical in a BPM initiative. The framework builds on a decision model that combines standard MCDM methods with a way to deal with uncertainty. We demonstrated the applicability of the proposed artefact based on the requirements of a BPM initiative at a major telecommunications company and a survey-based analysis of 19 BPM tool vendors in the European market. The results of the tool selection indicate that the application of a maturity-oriented and scenario-based decision framework is suitable to facilitate communication and foster transparency throughout such selection process. Although this particular framework focuses on simulation capabilities and IT governance model support, we argue that the demonstrated approach is viable to be applied in any organization facing the challenge to choose the right BPM tool.

5.2 Limitations and Future Work

We did - in most cases - not include specific implementation of functionalities which can be altered by the applying company or checked in on-site workshops. Further, we neglected the tool usability assessment and execution criteria (which could easily be included and are planned to be integrated within the next version of the decision model. An important constraint of this work is the evaluation in a single case example. By the nature of case-based research, generalizability to other organizational contexts may be limited despite the maturity-oriented approach. Thus, the evaluation performed here may rather be viewed as an indicative demonstration, rather than a rigorous evaluation. However, we are planning to apply this framework also in other, eventually less mature cases. Concerning the decision model, we made a few assumptions to increase practicability of the approach, such as constant preference weightings within a category and independent normally distributed utility scores. In a more sophisticated case, these assumptions may easily be altered increasing model complexity, yet, not changing the overall approach. Additionally, we point out some methodological drawbacks, such as the intrinsic subjectivity in utility and uncertainty coding and a moderate response rate (59%). Finally, in our evaluation we focus on the short listing phase of a tool selection process. In practice, on-site show cases and trial testing of short-listed tools are the next step to reduce the level of uncertainty before taking a final decision.

References

[BP80] D.R. Bartmann and J.A. Pope. Ein Scoring-Modell bei mehrfacher Zielsetzung mit unsicheren oder fehlenden Daten und abhängigen Zielen. *Mathematical Methods of Operations Research (ZOR)*, 24(2):29–45, 1980.

[BVCH07] V. Bosilj-Vuksic, V. Ceric, and V. Hlupic. Criteria for the Evaluation of Business Process Simulation Tools. *Interdisciplinary Journal of Information, Knowledge, and Management*, 2, 2007.

[CS08] A. Cater-Steel. *Information Technology Governance and Service Management: Frameworks and Adaptations*. Information Science Reference, 2008.

[DR10] I. Davies and M. Reeves. BPM Tool Selection: The Case of the Queensland Court of Justice. *Handbook on Business Process Management 1*, pages 339–360, 2010.

[Fis70] P.C. Fishburn. Utility theory for decision making, 1970.

[Has07] S. Hassmann. *Evaluation von BPM-Tools unter besonderer Berücksichtigung des Einsatzes im IT-Servicemanagement*. GRIN Verlag GmbH, 2007.

[HCKP09] J.B. Hill, M. Cantara, M. Kerremans, and D.C. Plummer. Magic Quadrant for Business Process Management Suites. Technical report, Gartner Research, 2009.

[HSW82] S.D. Hunt, R.D. Sparkman, and J.B. Wilcox. The Pre-Test in Survey Research: Issues and Preliminary Findings. *Journal of Marketing Research*, 19:269–73, 1982.

[Ins07] IT Governance Institute. Control Objectives for Information and Related Technology (CobiT), 4.1th Edition, 2007.

[JVN06] M. Jansen-Vullers and M. Netjes. Business process simulation–a tool survey. In *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark*, 2006.

[KB99] D. Kunda and L. Brooks. Applying social-technical approach for COTS selection. In *Proceedings of the 4th UKAIS Conference*, 1999.

[KM05] M. Kirikova and J. Makna. Renaissance of Business Process Modelling. In Olegas Vasilecas, Wita Wojtkowski, Jože Zupančič, Albertas Caplinskas, W. Gregory Wojtkowski, and Stanisław Wrycza, editors, *Information Systems Development*, chapter 35, pages 403–414. Springer-Verlag, New York, 2005.

[LMS06] R. Lahdelma, S. Makkonen, and P. Salminen. Multivariate Gaussian criteria in SMAA. *European Journal of Operational Research*, 170(3):957–970, 2006.

[McG09] M. McGregor. Mapping & Modelling Tools - Survey Report. Technical report, www.markmcgregor.com, 2009.

[RdB05] M. Rosemann and T. de Bruin. Towards a business process management maturity model. In *Proceedings of the 13th European Conference on Information Systems (ECIS 2005). Regensburg, Germany*, volume 26, 2005.

[Saa80] T.L. Saaty. The analytic hierarchy process, 1980.

[Ste96] T.J. Stewart. Robustness of additive value function methods in MCDM. *Journal of Multi-Criteria Decision Analysis*, 5(4):301–309, 1996.

[SW08a] G. Scheithauer and G. Wirtz. Applying Business Process Management Systems–a Case Study. In *The 2008 International Conference on Software Engineering and Knowledge Engineering (SEKE'08), Redwood City, California, USA*, 2008.

[SW08b] G. Scheithauer and G. Wirtz. Case Study: Applying Business Process Management Systems. In *SEKE*, pages 12–15. Knowledge Systems Institute Graduate School, 2008.

[SW08c] D. Spath and A. Weisbecker. Business Process Management Tools 2008. Technical report, Fraunhofer IAO, 2008.

[vdAHW03a] W.M.P. van der Aalst, A.H.M. Hofstede, and M. Weske. Business process management: A survey. *Lecture Notes in Computer Science*, pages 1–12, 2003.

[vdAHW03b] W.M.P. van der Aalst, A.H.M. Hofstede, and M. Weske. Business process management: A survey. *Lecture Notes in Computer Science*, pages 1–12, 2003.

[vdAW04] W.M.P. van der Aalst and A.J.M.M. Weijters. Process mining: a research agenda. *Computers in Industry*, 53(3):231 – 244, 2004. Process / Workflow Mining.

[vDdMV+05] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A. Weijters, and W.M.P. van der Aalst. The ProM framework: A new era in process mining tool support. *Application and Theory of Petri Nets*, 3536:444–454, 2005.

[Vol08] K. Vollmer. The Forrester Wave: Integration-Centric Business Process Management Suites, Q4 2008. Technical report, Forrester Research Inc., 2008.

[WD08] N. Ward-Dutton. BPM vendor capability comparison, 2H08. Technical report, MacehiterWard-Dutton (MWD), 2008.

Viewpoint-Based Modeling—Towards Defining the Viewpoint Concept and Implications for Supporting Modeling Tools

Klaus Fischer[1], Dima Panfilenko[1], Julian Krumeich[1], Marc Born[2], Philippe Desfray[3]

DFKI GmbH[1], Saarbrücken, Germany
ikv++ technologies ag[2], Berlin, Germany
SOFTEAM[3], Paris, France
{firstname.surname}@dfki.de, born@ikv.de, philippe.desfray@softeam.fr

Abstract: Viewpoint-based modeling is an important recent development in software engineering. It is likely to boost the wider use of modeling techniques because it allows to tailor existing tools with respect to the different stakeholders in software design. The paper reports on results from the ViBaM project in which viewpoint concepts are investigated. We give an overview of the most important contributions from literature regarding viewpoint concepts from which we derived the position that we take in the ViBaM project. After presenting ViBaM's position we derive features that we consider important for tools that support viewpoint features. We plan to integrate these features in the commercial modeling tools MODELIO and MEDINI ANALYZE to the end of the ViBaM project.

1 Introduction

In the course of constructing large-scale and complex systems, models are a prevalent means for gaining a better understanding of the underlying artifact that is to be engineered. Especially in traditional engineering disciplines [BKS10], graphical representations of models are widely used to design, describe and discuss various types of systems (take for instance building plans or engineering drawings). Apart from these traditional engineering disciplines, models are being increasingly used in the course of developing information systems as well as information systems architectures. In this regard, they do not only serve as a means for building systems from scratch, they are also used to capture current as-is states, e.g. for documentation purposes, with the eventual goal to derive optimized to-be states.

As a consequence, modeling techniques are commonly used nowadays—in particular standard modeling languages such as the Unified Modeling Language (UML) or the Business Process Modeling Notation (BPMN), but also proprietary and domain specific modeling languages such as those supported by Business Process Management (BPM) or Enterprise Architecture (EA) tools. However, according to our estimation, their usage has reached a significant threshold and a wider usage and diffusion needs more added value and perceived Return on Investment (ROI) by the end users. This can only be achieved if modeling techniques as well as corresponding tools put actual stakeholders more into the focus during the modeling process.

1.1 Motivation and Contribution of this Paper

According to Stachowiak [Sta73], models are a representation of a mental or physical object, in which this object can again be a model itself. Due to the reduction feature of models, they do not comprise every detail of the underlying object, since their purpose is to describe and analyze complex systems in a manageable way. However—even though models aim at reducing complexity—in large-scale systems, they are often still too complex, which is a common modeling problem [BW06]. This becomes even worse if only a single, rigid perspective is provided on models and moreover if models are used for cooperative and collaborative system design.

In such settings, many different stakeholders are working together to design a system. Each of them has very own skills, responsibilities, knowledge and expertise [FKN+92]; and thus, a very unique perspective on the system design. As a consequence, each person or group of people that is in charge with the system design would like to view and manipulate the models or model fragments according to their particular needs [DQvS08]. To foster the modeling usage considering these demands especially in the course of constructing large-scale systems, **Viewpoint-Based Modeling** (VBM) is an increasingly used technique to reduce complexity and adapt an overall model to stakeholder specific fragments in a successful manner [GBB12]. In doing so, the actual stakeholders are being put more into the focus of the modeling process, which results in a higher perceived value for them. By utilizing stakeholder-specific viewpoints on a model, its overall understanding and productivity will increase [FKN+92]. As a direct consequence, the viewpoint concept leads to better conceptual models, which is proven by several studies. For example, according to [EYA+05], applying the viewpoint concepts effectively helps to cope with the overall size of a given problem domain. Even though the usefulness of VBM is obvious and objectively proven, the concept's meaning is still quite fuzzy, since it is heavily overloaded in literature and frequently only defined in an informal manner [MP00], [GBB12].

Thus, this paper contributes towards knowledge on VBM in two respects: On the one hand, it provides a systematic clarification of the viewpoint concept on a general abstract level. In doing so, researchers will benefit from this definition and further research on VBM can be conducted on this basis. On the other hand, this paper provides various features that need to be considered in modeling tools in order to achieve a successful utilization of the viewpoint concept. In doing so, tool builders and vendors will benefit from this catalogue of features in order to validate their tools regarding VBM and to have a guideline for building new tools which support VBM.

1.2 Scientific Methodology and Organization of the Paper

The results of the paper originate from the European research project ViBaM that aims at investigating the viewpoint concept from a scientific perspective and integrate it into the commercial modeling tools MODELIO and MEDINI ANALYZE of the two involved project partners. In the first step towards analyzing and defining the viewpoint concept, a broad foundation of publications consisting of some 200 papers was built up. This was achieved by performing a search on the literature databases EBSCO Business Source Premier, Thomson Reuters Web of Knowledge and Google Scholar using search terms like ”viewpoint”, ”definition of viewpoint” or ”viewpoint-based modeling”. Afterwards, promising literature mainly focusing on viewpoint definitions have been picked up and screened to define our perspective on a general-applicable viewpoint definition. In order to assure an objective analysis, a framework consisting of eight categories has been constructed. To derive implications for modeling tools, literature mainly focusing on general and dedicated modeling features has been selected and analyzed by using two frameworks that have been created and which consist of nine and seven categories respectively. In this regard, we stress that some of these categories are taken from a number of viewpoint-related methods originated from the Software Engineering and Enterprise Architecture (EA) context.

To apply the proposed methodology, the remainder of this paper is organized as follows. Section 2 presents a discussion on the multi-domain purpose of viewpoints and outlines current limitations. In Section 3, viewpoints and adjacent concepts are defined and linked to an overarching metamodel. Furthermore, the paper's viewpoint concept is exemplified on the EA domain based on the TOGAF framework. Moreover, implications for tooling based on various features are illustrated. Finally, Section 4 summarizes the paper and provides an outlook to our future work on VBM.

2 Viewpoints—A Multi-Domain Concept and its Current Limitations

Without doubt, the concept of viewpoints is not novel. According to Lankhorst [Lan09], already in 1985, the *MultiView* approach proposed by Wood-Harper et al. [WHAA85] forms the concept's origin. *MultiView* aims at supporting the development process of (computerized) information systems by splitting its complex process into five different perspectives respective viewpoints: Human Activity System, Information Modelling, Socio-Technical System, Human-computer Interface, and Technical System. A decade later, the *MultiView* framework has been revised to *MultiView2* [AWHVW98]. Another early-published and frequently-cited work was conducted by Finkelstein et al. [FEKN92]. Even though it was published in the *International Journal of Software Engineering and Knowledge Engineering*, the focus of their work is not limited to software development, but can be applied to multiple kinds of artifacts which need to be constructed through a non-trivial engineering process. Hence, this corroborates the multi-domain spanning interest of the viewpoint concept, which will be further outlined in the remainder of this section.

With a stronger focus on software systems, Kruchten [Kru95] presents a model for describing the architecture of software-intensive systems, which it is often referred to as the *"4+1" View Model of Software Architecture*. This approach aims at the design and the implementation of the high-level structure of software. The architecture model is composed of multiple, concurrent perspectives called views. A view addresses a specific set of concerns looking on the system of the perspective of a particular stakeholder (group) [Kru95]. Also in the context of software development, viewpoints are utilized in the domain of requirements engineering. For example, Kotonya & Sommerville present a *Viewpoint-Oriented Requirements Definition* (VORD) approach in [KS96].

Another domain in which the viewpoint technique is frequently applied to is the Enterprise Architecture (EA) domain. In the course of this, the *Zachman Framework* [Zac00] (initial called as *Framework for Information Systems Architecture*), which is one of the earliest-published and most known frameworks, consists of a two dimensional classification matrix. The first dimension differentiates six different viewpoints: Planner, Owner, Designer, Builder, Subcontractor and Functioning System view. Orthogonal to this dimension, the Zachman framework differentiates between six different aspects: Data, Function, Network, People, Time and Motivation description. Another EA framework that is frequently used in practice is the *The Open Group Architecture Framework* (TOGAF) [The11]. TOGAF differentiates—without providing a concrete definition—between four viewpoints: Business, Data, Application and Technology Architecture.

While the previously mentioned concepts of viewpoints rather follow the idea of separating an artifact of interest in a vertical dimension meaning a union of all viewpoints would provide an overall vision on an artifact based on a specific level of detail, there is also a popular concept following the opposite direction [GBB12]. The *Model Driven Architecture* (MDA) [Obj03] follow the idea of introducing a strong separation of concerns regarding modeling a system at different abstraction levels. This starts from computational (CIM) and platform-independent (PIM) models and uses transformations to produce the actual code for the selected programming language and platform (PSM). In contrast to vertical-centric viewpoints, a viewpoint in MDA encompasses the whole underlying system, but on its specific level of detail regarding the distance to a concrete system implementation. Of course, even in MDA, a separation of concerns—preferable by using viewpoints—within one abstraction level seems necessary to reduce complexity.

Even though this section outlined the fact that viewpoints as a concept are very often used in multiple domains, the concept either lacks a definitional or scientific foundation (e.g. [The11]) or it strongly focuses on a specific domain. Furthermore, knowledge on requirements and implications for modeling tools aiming at realizing the viewpoint for practical usage concept is missing. One first approach towards this direction is the one recently published by Goldschmidt et al. [GBB12]. Even though their work is closely related to the one we present in this paper, it mainly focuses on a feature-based classification of view-based domain-specific modeling concepts, while in contrast we follow a more general approach; nevertheless, this recent paper shows that there is still a huge need in research particular regarding modeling tools that provide means for using the viewpoint technique.

3 The Concept of Viewpoints—Definition, Usage and Tool-Support

According to our observations, most of the definitions we found have something in common: they define the concept of ”viewpoint” as a guideline for constructing views. This can for example be observed in the *IEEE 1471:2000* standard definition [Sof07] for viewpoints, which was adopted by ISO/IEC as an *ISO/IEC 42010:2007* standard in 2007 [Hil11]. Accordingly, a viewpoint is ”a specification of the conventions for constructing and using a view”. This definition is most widely accepted in general system engineering, but also in software development. A related viewpoint definition is for example given by [ACWG94]. Consequently, the viewpoint can be seen as a pattern that defines a set of views. Another common feature of most of those definitions is that a viewpoint explicitly specifies one or more stakeholders, whose point of view it represents. Furthermore, some definitions explicitly note that a viewpoint should be as much self-contained as possible. What we are particularly interested in and will be investigating in this section is whether viewpoints are defined in terms of a metamodel, as well as whether this metamodel is separate or somehow related to other metamodels. Having relations or guidelines that drive viewpoint life-cycle and interaction as well as (de-)centralization of viewpoint underlying metamodels are further aspects in this section. Furthermore, another feature we are interested in is whether a viewpoint directly reflects the needs of a particular stakeholder.

3.1 Towards Defining the Viewpoint Concept

Due to the large amount of the related work and definitions found in the literature (more then a dozen of definitions and additionally methods for dealing with viewpoints providing their own definitions), we first clarify the terms and definitions which we take as a basis. These terms are stated in the lists below and visualized in Figure 1. We give a brief characterization of the basic terms where the first 5 below were derived and are in line with the definition given in OMG’s Model Driven Architecture (MDA) [Obj03]:

Metamodel: A metamodel defines a frame and a set of rules for creating models for a specific application domain. Metamodels typically establish possible domain objects and relationships between them, as well as constraints that should be applied to them. Metamodels serve as a basis for models instantiating them, profiles referencing them and viewpoints that choose multiple modeling concepts from various metamodels to represent as a comprehensive big picture. Related modeling concepts usually belong to a certain metamodel.

Model Concept: A model concept is an element of a metamodel. The metamodel contains a type for each relevant modeling concept and defines the relation between these types. Model concepts are part of certain metamodel and are the basis for model elements in model instances as well as for the definition of viewpoints, which might be defined across multiple metamodels.

Model: A model is an instance of a metamodel. It contains a concrete set of model elements, which adhere to the rules defined in the corresponding metamodel. Models can apply certain profiles and thus represent model elements accordingly and serve as collection items for views.

Model Element: A model element is a concrete instance of a modeling concept, and thus it represents either a domain object or a relationship between two or more objects. These elements are a part of a certain model and are being exposed in certain views belonging to certain viewpoint instances.

Profile: A profile is an extension of a metamodel, which uses the metamodel as a reference for redefining existing modeling concepts and thus targeting specific domains. It serves a basis for viewpoints, refers to a certain metamodel and can be applied to various models for domain alignment.

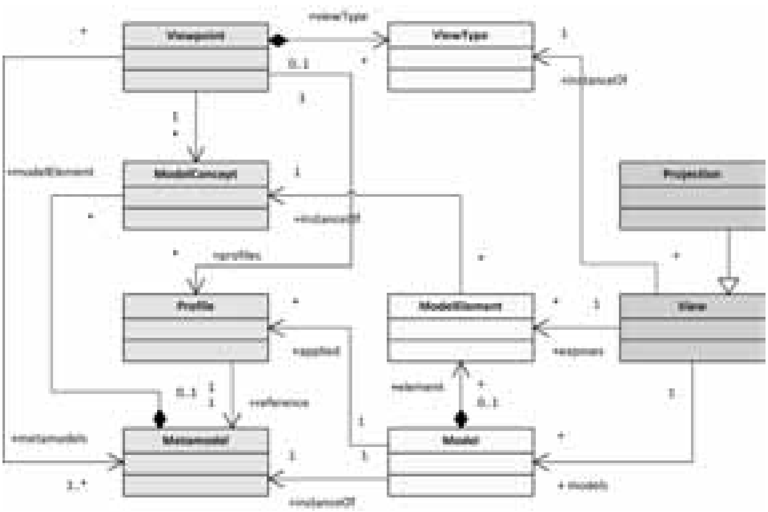


Figure 1: VIBAM's Position on Viewpoint Concept.

We give an additional list of important terms with brief characterizations where we give references to the work that was of major influence on our understanding of the meaning:

Viewpoint (cf. [The11], [Lan09], [CMR03]): The purpose of a viewpoint is to support a stakeholder in contributing to system design from a specific perspective. A viewpoint defines what concepts and relations can be used to define, view, or manipulate model instances within this viewpoint. It is therefore related to a (set of) metamodel(s), a (set of) profile(s) or a part of them. The viewpoint in this sense can restrict the original metamodel(s) but it can also correspond to a metamodel 1:1. The viewpoint is defined by the collection of view types that it offers to the stakeholders and which are instantiated by views.

View Type (cf. [GBB12]): A collection of view types is defined for each viewpoint. A view type serves a basis for view instantiation and offers a specific slice of system perspective to the stakeholders (i.e. human users).

View (cf. [Lan09], [CMR03]): A view defines the presentation of model elements to a stakeholder and the way(s) how they can be modified (this is usually achieved by diagram types). It enables the user to interact with particular aspects of one or more models that adhere to the viewpoint's metamodel.

Projection (cf. [Pra11], [IBM03]): Function that maps a model instance M to another model instance M' where M' is a restriction of M in the sense that it contains only elements that are also contained in M and both M and M' are instances of the same metamodel. A projection is a special case of defining a view.

An intuitive reading of the metamodel depicted in Figure 1 is that viewpoints are either directly defined on metamodels or on top of profiles. Viewpoints offer a set of views which allow the stakeholder for whom the viewpoint was defined to access the model instances. Based on the general terms and the analysis of existing definitions, we derived a harmonized viewpoint definition that serves as a basis for the work in the VIBAM project.

Definition: A viewpoint is defined in relation to one or more metamodels. For each viewpoint a non-empty set of view types is defined. In a viewpoint instance any number of instance views for each of the view types can be dynamically created.

In the following we list features of viewpoint definitions which we consider important and which influenced our understanding on viewpoints during related work analysis. Moreover, Table 1 shows whether these features are considered within existing definitions (cf. [Sof07], [Val01], [Hil11], [The11], [Pra11], [IBM03], [SADL04], [Kru95], [RW05], [Zac09], [Nus94], [ACWG94], [DQPvS04], [FEKN92]).

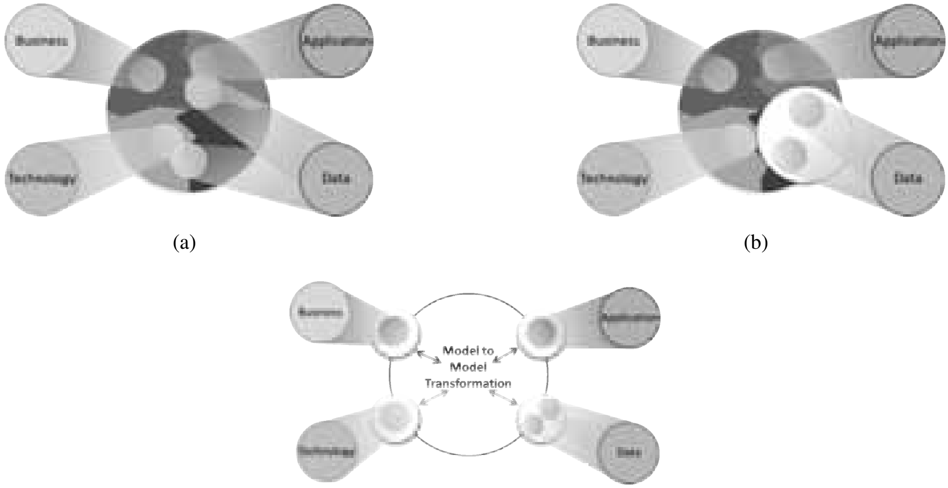
Key: ✓ = Yes ✗ = No / = Implied	[Sof07]	[Val01]	[Hil11]	[The11]	[Pra11]	[IBM03]	[SADL04]	[illegible]	[Zac09]	[Nus94]	[ACWG94]	[DQPvS04]	[FEKN92]
A viewpoint is a partial specification of a system.	✓	✓	✓	✓	/	/	/	✗	/	✓	✓	/	✓
A viewpoint is composed of one or more views.	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	/
A viewpoint is a specification for creating views.	✓	✗	✓	✓	/	/	✓	✓	✗	/	✗	✓	✗
A viewpoint is defined by means of a metamodel.	✓	✓	✓	✓	/	✗	/	/	✓	/	/	/	✓
Metamodels are centralized.	✗	✗	✗	✗	/	✗	✗	✗	✗	✗	✗	✗	✗
Metamodels are decentralized.	/	/	/	/	✗	✗	/	/	/	/	/	/	✓
There is assignment of stakeholders.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓
There is a method which adopts the definition	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗

Table 1: Viewpoint Definition Features.

A viewpoint is a partial specification of a system: A viewpoint contains certain functional description and information types which are implemented and used in the developed system at run-time. As we can see from Table 1, almost all of the research shares this opinion over the viewpoint definition. *A viewpoint is composed of one or more views:* A viewpoint is defined by a language which we can refer to as a metamodel, and it explicitly addresses specific stakeholders. Only a few authors (IEEE [Sof07] and ISO [Hil11] standards, Zachman [Zac09] and Finkelstein et al. [FEKN92]) use this notion of the viewpoint in their definitions, whereas this leads us to a more comprehensive and consistent system description. *A viewpoint is a specification for creating views:* A viewpoint is a pattern or template from which to develop individual views. Thus a view is a concrete instance of a viewpoint. Most of the authors see the viewpoint as this, whereas specification of views as view typing is how the authors of this paper understand this term. *A viewpoint is defined by means of a metamodel:* A viewpoint is a type of metamodel for view creation. As in the previous point, almost everybody agrees on metamodels being a basis for viewpoints defining their target domain usage. *Metamodels are centralized:* This means there is no clear separation between viewpoints and views. Each viewpoint governs which kind of model element can be represented, the consistency rules and completeness rules that needs to be applied, and the different view that can be provided. Only PRAXEME [Pra11] method partly defends this point of view. *Metamodels are decentralized:* This means a viewpoint is a loosely coupled, locally managed, coarse-grained object which encapsulates partial knowledge about the system and domain, specified in a particular, suitable representation scheme. Apart from PRAXEME [Pra11] and RUP [IBM03] other methods are not supporting this point of view. *There is assignment of stakeholders:* Each view targets a specific group of stakeholders, thus separating modeling concerns and assuring consistency. Each stakeholder is then responsible for designing his model part with the aid of constructs provided by the assigned view. This is agreed upon by everybody except for Ainsworth et al. [ACWG94] and Nuseibeh et al. [Nus94]. *There is a method which adopts the definition:* Existence of a method in research or industry, which uses the current viewpoint definition, is under question here. If a viewpoint definition is not used in any methods, this makes the definition a pure academic matter.

3.2 Applying Viewpoints on TOGAF

In order to clarify our theoretical explanations, this section will exemplify our understanding of the viewpoint concept on the Enterprise Architecture domain using the TOGAF framework as a basis. As already mentioned in Section 2, TOGAF differentiates between four different viewpoints. Figure 2a shows the different stakeholders regarding the TOGAF method for system design. When we want to deal with viewpoint concepts in a technical sense, we can start off with looking at an application domain for which we capture the relevant concepts in one or more metamodels. We refer to the set of these basic metamodels with $\mathcal{M}_b$ (displayed in blue in Figure 2a). To define a viewpoint we select the set of concepts from $\mathcal{M}_b$ that we consider relevant for the viewpoint (see Figure 2b). The selection criteria is whether a specific concept is relevant for the stakeholder which the given viewpoint should then support. The selected concepts form the concepts for the viewpoint metamodel M_v. The relations for all concepts in M_v have to reflect the relations between the corresponding concepts in $\mathcal{M}_b$. We do assume that the relations between concepts in $\mathcal{M}_b$ do not contradict each other.



(a) (b) (c)

Figure 2: TOGAF Perspective to System Design and Information Exchange between Viewpoints.

For a viewpoint metamodel M_v we now can define concrete views that will allow the stakeholder to actually access the model instances that adhere to the viewpoint metamodel M_v. Model to model transformations are used to pass information or model fragments between different viewpoints (see Figure 2c). OCL constraints included in the metamodel representations are used to validate model instances. Even within a viewpoint definition given by a viewpoint metamodel M_v projections can be used to distinguish between different stakeholders and with this support them with different views to the model instances. Projections are defined on the model instance level and can be even defined dynamically by attaching annotations to the model instances. There is no real need to make a copy of a model instance to create a new projection. Rather the defined views are adapted or restricted in a manner that serves the respective stakeholder best. If more than one human user is manipulating the same model instance possibly using different perspective (i.e. by using different projections), consistency of the model instance is of major concern.

3.3 Implications for Tooling

The features of viewpoints given in this section were derived with respect to VIBAM's viewpoint definition to allow for a comparison of viewpoint definitions of relevant methods from literature. In the following we present the comparison of the methods which we consider most important: IEEE 1471-2000 [Sof07], Kruchten [Kru95], RM-ODP [Val01], ISO/IEC 42010 [Hil11], SysML [Obj11], Zachman [Zac09], MODAF [Cro09], TOGAF [The11], Boiten [BBD$^+$00], PRAXEME [Pra11], RUP [IBM03]. The selected viewpoint methods have been examined from two angles: general and dedicated features. The difference between them lies in the scope of the analysis of the viewpoint methods under surveillance. The general features refer rather to the external influence on the viewpoints like support for predefined viewpoints and transformation rules between them. Dedicated features refer to the internal features of the viewpoints like viewpoint definition, consistency rules inside and between viewpoints. A closer look at the analysis tables below gives insight into the tooling features we derived and recommend for further implementation.

General Viewpoint Features of the Methods

Key: ✓ = Yes ✗ = No ! = Implied	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Support for predefined viewpoints	!	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
Support for addressing specific stakeholders	!	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
Support for adaptable presentation formalisms	✗	✗	✓	✗	✓	✓	✓	✓	✓	✗	✓
Support for transformation rules between different viewpoints	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
Support for ad hoc viewpoint creation	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
Support for dynamic viewpoint creation	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Support for dedicated exploitations	✗	✗	!	✗	!	✓	✓	✓	!	✗	✓
Support for adaptation to the organization context	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓
Support for relationship between viewpoints and development lifecycle	✗	!	✗	✗	✗	!	!	!	✗	✓	✓

Table 2: Viewpoint General Features.

Support for Predefined Viewpoints: A predefined viewpoint is defined prior to the application of a method. Usually, there is a fixed set of predefined viewpoints unrelated to any domain, i.e. they are for example defined by the method itself (cf. RM-ODP). As a consequence, this feature can imply a limitation on versatility of the method's application. One example in this regard is TOGAF which is a dedicate enterprise architecture. Apart from ISO/IEC and SysML, all of the methods are defining viewpoints in advance. *Support for Addressing Specific Stakeholders:* This feature defines whether the method targets specific stakeholders and hence proposes specific concepts for these stakeholders. This implicates a number of predefined user groups, which in turn means targeting specific domains in advance resulting in limitation of versatility and more specific stakeholder targeting. As a result, all considered methods except for ISO/IEC and SysML are defining stakeholders a priori. *Support for Adaptable Presentation Formalisms:* Adaptable presentation formalisms provide the ability to adjust the presentation of model elements to the needs of certain users or to conform to certain viewpoints. This can be realized by a profile. Apart from IEEE, Kruchten, ISO/IEC and PRAXEME, all methods allow a flexible definition of views. *Support for Transformation Rules between different Viewpoints:* This feature expresses the existence of constructive rules that allow deriving model elements for a particular viewpoint out of model elements from another viewpoint. These rules can be understood as model transformation between different viewpoints. Although almost all of the viewpoint definitions support decentralized viewpoints, not every method provides transformation rules—except for IEEE, ISO/IEC and PRAXEME. *Support for Ad Hoc Viewpoint Creation:* Some methods recommend to define viewpoints during a project's preparation phase to address unforeseen stakeholder groups. This feature is called ”ad hoc” in contrast to ”predefined” viewpoints. However, only SysML and TOGAF see the need for making room for such developments. *Support for Dynamic Viewpoint Creation:* Creating a viewpoint completely dynamically in a sense of creating a viewpoint on the fly after a project has already begun. SysML is the only method that provides means for giving stakeholders this capability. *Support for Dedicated Exploitation:* A certain need for a usage of a viewpoint unintended during the design time may occur at run-time (e.g. a viewpoint with deactivated dependencies to other viewpoints for protecting sensible information). Most of the methods support this at least partially. *Support for Adaptation to the Organization Context:* This feature defines whether the existing viewpoint of a method can be adapted to a specific organizational context in order to suit the intended usage of the system. The three methods—Kruchten, RM-ODP and Boiten—do not see the need for adapting the viewpoints to specific organizations. *Support for Relationship between Viewpoints and Development Lifecycle:* This features outlines whether the method provides certain guidance or recommendations to relate viewpoints with the development lifecycle of systems. This is not supported by all of the methods—only two of them, namely PRAXEME and RUP, are looking into realizing this feature to the full extent.

Dedicated Viewpoint Features of the Methods

Key: ✓ = Yes ✗ = No / = Implied	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Contains an own viewpoint definition	✓	✓	✓	✓	✗	/	✗	✓	✗	/	✓
Contains impact analysis features	✓	✗	/	/	✓	✗	✓	✓	/	✓	✓
Contains projection features	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Contains filtering features	✓	✗	✗	✗	✗	✗	✗	✗	✗	/	✗
Contains consistency rules between different viewpoints	/	/	✓	✓	✓	/	/	/	/	/	/
Contains consistency rules between views	✓	/	✓	✓	✓	/	/	/	/	/	/
Contains consistency rules within a view	✓	/	✓	✓	✓	/	/	/	/	/	/

Table 3: Viewpoint Dedicated Features.

Contains an own viewpoint definition: This feature signifies whether the method provides its own viewpoint definition. SysML, MODAF and Boiten do not provide own definitions. *Contains impact analysis features:* The question here is whether a traceability methodology is available and, if yes, whether it allows impact analysis of the model changes between viewpoints. All the methods except for Kruchten and Zachman do explicitly state this. *Contains projection features:* This feature is derived from the projection definition in Section 3.1. Hence, the idea is to let different viewpoints edit the same model, whereas certain constructs are represented in different ways in each of the viewpoints. None of the methods apart from PRAXEME and RUP adhere to this feature. *Contains filtering features:* In their filtering capacities, viewpoints will filter out those model elements that are not allowed to appear under a certain view, and thus only elements eligible for a viewpoint will be provided for view modeling. The two methods IEEE and PRAXEME are at least partially support this capability. *Contains consistency rules between different viewpoints:* If a system is modeled using different viewpoints, the model elements which are defined in these viewpoints are usually not completely independent form each other. There might be certain rules that need to be obeyed to ensure the overall consistency of the underlying model. All of the methods support the consistency rules feature. *Contains consistency rules between views:* In the same manner as it was for consistency between different viewpoints, certain rules may need to be obeyed in order to ensure the consistency in a certain viewpoint between its views. All of the methods ask for consistency between the views. *Contains consistency rules within a view:* As in the two features before, there might be a threat for inconsistencies inside a certain view due to editing from the different view instances. As a consequence, certain instance level consistency rules may have to be obeyed. All methods require this feature.

4 Conclusion and Future Work

In this paper we reported on results of the VIBAM project in which viewpoint concepts are investigated. We presented an overview of the current state-of-the-art for viewpoint definitions, concepts and methods. Derived from the definitions we found in literature we present definitions for the list of concepts on the basis of which we define the position that we take in the VIBAM project regarding viewpoint concepts. After we discussed VIBAM's position on viewpoints we presented a list of features that we consider important for tool support of the viewpoint definition presented in this paper.

The next step in our work is to integrated the defined concepts in the commercial modeling tools MODELIO and MEDINI ANALYZE. With respect to the basic technologies the implementation of dynamic viewpoint creations is rather difficult to achieve. Even the definition of views for example on the basis of EMF/GMF is cumbersome if one is not satisfied with the default behavior that is offered for this technology stack. We will investigated what changes would be needed to make the use of the basic technologies more flexible.

Acknowledgements: This work has been funded by the German Federal Ministry of Education and Research (FKZ 01QE1106B/C) and by the French OSEO through ICT Project ViBaM (Viewpoint-Based Modeling) which is running in the context of the European Eurostars Program (E!5529). The authors wish to acknowledge EUREKA and the Commission for their support. We also thank the reviewers for their valuable comments.

References

[ACWG94] M. Ainsworth, A.H. Cruickshank, P.J.L. Wallis, and L.J. Groves. Viewpoint specification and Z. In *Information and Software Technology*, 36(1):43–51, 1994.

[AWHVW98] D.E. Avison, A.T. Wood-Harper, R.T. Vidgen, and J.R.G. Wood. A further exploration into information systems development: the evolution of Multiview2. In *Infor-*

mation Technology & People, 11(2):124–139, 1998.

[BBD+00] E. Boiten, H. Bowman, J. Derrick, P. Linington, and M. Steen. Viewpoint consistency in ODP. In *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 34(3):503–537, 2000.

[BKS10] S. Buckl, S. Krell, and C.M. Schweda. A Formal Approach to Architectural Descriptions—Refining the ISO Standard 42010. In A. Albani, J.L.G. Dietz, W. van der Aalst, J. Mylopoulos, M. Rosemann, M.J. Shaw, and C. Szyperski, editors, *Advances in Enterprise Engineering IV*, Vol. 49 of *Lecture Notes in Business Information Processing*, pp. 77–91. Springer, Berlin Heidelberg, Germany, 2010.

[BW06] P. Balabko, and A. Wegmann. Systemic classification of concern-based design methods in the context of enterprise architecture. In *Information Systems Frontiers*, 8(2):115–131, 2006.

[CMR03] J. Champeau, F. Mekerke, and E. Rochefort. Towards a Clear Definition of Patterns, Aspects and Views in MDA. In *Proc. of the First International Workshop on Engineering Methods to Support Information Systems Evolution*, Geneva, Switzerland, 2003.

[Cro09] Crown. A summary of MODAF views by their use and data types, 2009.

[DQPvS04] R.M. Dijkman, D.A.C. Quartel, L.F. Pires, and M.J. van Sinderen. A rigorous approach to relate enterprise and computational viewpoints. In *Proc. of the 8th IEEE Enterprise Distributed Object Computing (EDOC) Conference*, Monterey, California, USA, 2004.

[DQvS08] R.M. Dijkman, D.A.C. Quartel, and M.J. van Sinderen. Consistency in multi-viewpoint design of enterprise information systems. In *Information and Software Technology*, 50(7-8):737–752, 2008.

[EYA+05] S. Easterbrook, E. Yu, J. Ar, Y. Fan, J. Horkoff, M. Leica, and R.A. Qadir. Do viewpoints lead to better conceptual models? An exploratory case study. In *RE*, pp. 199–208, 2005.

[FEKN92] A. Finkelstein, S. Easterbrook, J. Kramer, and B. Nuseibeh. Requirements Engineering Through Viewpoints. In *DRA Colloquium on Analysis of Requirements for Software Intensive Systems*, pp. 18–26, Defence Research Agency, 1993.

[FKN+92] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke. Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. In *International Journal of Software Engineering and Knowledge Engineering*, 2(1):31–57, 1992.

[GBB12] T. Goldschmidt, S. Becker, and E. Burger. Towards a Tool-Oriented Taxonomy of View-Based Modelling. In E.J. Sinz, and A. Schürr, editors, *Modellierung*, Vol. 201 of *GI-LNI*, pp. 59–74, 2012.

[Hil11] R. Hilliard. Welcome to the ISO/IEC 42010 Website. `http://www.iso-architecture.org/ieee-1471/`, Accessed July 2012.

[IBM03] IBM. Rational Unified Process: A Best Practices Approach, 2003.

[Kru95] P. Kruchten. Architectural Blueprints - The 4+1 View Model of Software Architecture. In *IEEE Software*, 12(6):42–50, 1995.

[KS96] G. Kotonya, and I. Sommerville. Requirements Engineering With Viewpoints. In *Software Engineering Journal*, 11(1):5–18, 1996.

[Lan09] M. Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin Heidelberg, Germany, 2009.

[MP00] R. Motschnig-Pitrik. The viewpoint abstraction in object-oriented modeling and the UML. In *Proc. of the 19th international conference on Conceptual modeling*, ER'00, Salt Lake City, Utah, USA, 2000.

[Nus94] B.A. Nuseibeh. *A Multi-Perspective Framework for Method Integration*. Dissertation, Imperial College of Science, Technology and Medicine, University of London, Department of Computing, 1994.

[Obj03] Object Management Group. MDA Guide Version 1.0.1. `http://www.omg.org/cgi-bin/doc?omg/03-06-01`, Accessed July 2012.

[Obj11] Object Management Group. The Official OMG SysML website. `http://www.omgsysml.org`, Accessed July 2012.

[Pra11] Praxeme Institute. PRAXEME - Opus, the Product. `http://www.praxeme.org/index.php?n=Opus.Opus`, Accessed July 2012.

[RW05] N. Rozanski, and E. Woods. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley Longman, Amsterdam, 2005.

[SADL04] M.W.A. Steen, D.H. Akehurst, H.W.L. Doest, and M.M. Lankhorst. Supporting Viewpoint-Oriented Enterprise Architecture. In *Proc. of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC'04)*, Monterey, California, USA, 2004.

[Sof07] Software Engineering Standards Committee of the IEEE Computer Society. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Software Engineering Standards Committee of the IEEE Computer Society, 2007.

[Sta73] H. Stachowiak. *Allgemeine Modelltheorie*. Springer, Berlin Heidelberg, Germany, 1973.

[The11] The Open Group. Welcome to TOGAF Version 9 - an Open Group Standard. `http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html`, Accessed July 2012.

[Val01] A. Vallecillo. RM-ODP: The ISO Reference Model for Open Distributed Processing. *DINTEL Edition on Software Engineering*, 3:66–69, 2001.

[WHAA85] A.T. Wood-Harper, L. Antill, and D.E. Avison. *Information systems definition: the Multiview approach*. Blackwell Scientific Publications, Oxford, UK, 1985.

[Zac00] J.A. Zachman. Concepts of the Framework for Enterprise Architecture. `http://www.ies.aust.com/papers/zachman3.htm`, Accessed July 2012.

[Zac09] J.A. Zachman. The Zachman Framework: The Official Concise Definition. `http://old.zachmaninternational.com/concise%20definition.pdf`, Accessed July 2012.

Key Information Requirements for Process Audits – an Expert Perspective

M. Schultz, N. Mueller-Wickop, M. Nuettgens

School of Business, Economics and Social Sciences
University of Hamburg
Max-Brauer-Allee 60
D-22765 Hamburg
{martin.schultz | niels.mueller-wickop | markus.nuettgens}@wiso.uni-hamburg.de

Abstract: In the audit domain it is agreed that a comprehensive understanding of business processes is crucial for the effectiveness and efficiency of internal and external audits on financial reporting or regulatory compliance. However, a review of current modeling methods revealed that domain specific concepts are not comprehensively supported and only little empirical research has been performed on what modeling concepts are required to support an understanding of business processes from an audit perspective. For this reason, we conducted 17 semi-structured expert interviews to reconstruct key concepts of the audit domain especially focusing on process audits. As a result we present twelve relevant audit concepts and their relations in a concept map. Unlike for concepts, the expert understanding of concept relations was quite diverse. We interpret this result as an indication of complexity for the topic in focus. The presented concept map is a first step towards a domain specific modeling language.

1 Introduction

Nowadays it is widely recognized that auditors play a crucial role in preventing accounting scandals like Enron 2001, MCI WorldCom 2002, Parmalat 2003, Satyam 2009, HRE 2011 or Olympus 2011. Inadequatily conducted audits can result in unprecedented business turbulences with corporate fraud and possible bankruptcy. In order to cope with increasing data volume, auditors focus on the audit of business processes [Be97] [Ru03] [Ru06]. This approach is based on the assumption that well-controlled business processes lead to correct preparation, presentation and disclosure of financial statements. For this reason, the auditing of business processes is also required by international audit standards like ISA 315.81: “the auditor should obtain an understanding of the information system, including the related business processes, relevant to financial reporting (...)” [IFAC10]. Consequentially, increasing attention on this topic in academia can be recognized. Researchers with different backgrounds are working in the broad field of information systems support for the audit domain. Diverse foci are set: from automated tool support to basic business process modeling guidance. A constantly evolving range of IS-based approaches can be observed (see section 2).

Our research project is located within the process mining domain with an explicit focus on audits of business processes which are linked to the financial statements or effect the regulatory compliance of a company. To support business process auditors in the best possible way, inter alia, we evaluated process modeling methods with respect to their suitability for process audits. As a first step in our research project we intended to survey requirements from literature. To our surprise no comprehensive empirical requirements engineering could be found. Most papers dealing with the topic of audit requirements are devoted to requirements derived from abstract audit standards or resulted from a discourse with *single* experts not explicitly describing the applied research method. To our best knowledge there are no papers dealing with the topic of audit concept requirements engineering. Since, "*it is widely acknowledged within the software industry that software engineering projects are critically vulnerable when these activities (editor's note: requirements engineering) are performed poorly*" [ICS05, Chapter 2], we expanded our research agenda to include an empirical requirements engineering.

The research and results presented in this paper form a first step of a broader study on the requirements of auditors for business process modeling and audits. Following the approach of Ahlemann and Gastl [AG07] for the construction of an empirically grounded reference model, we conducted semi-structured expert interviews as a first step to capture expert's domain knowledge and collect initial empirical data. Against this background the contribution of the paper is twofold: on the one hand we present a rigor collection of information requirements for the audit domain in terms of relevant concepts. On the other hand the identified concepts are set into relation forming a conceptual model which characterizes the audit domain while neglecting technical aspects of implementation [Fr07]. Thus, we chose a concept map to depict our conceptual model as summarizing result of this paper. The next phase of our study will be a quantitative survey of audit concepts incorporating the results of this paper combined with an exhaustive literature review. This multi-method approach will set a rigor basis for the development of a domain-specific audit modeling language or the proper adoption of an existing one. The major benefit of such a modeling language was stated in all of our interviews: more effective and efficient execution of process audits. Primary stakeholders of this research are internal and external auditors. Additional stakeholders include - but are not limited to - process owners, risk managers, the board of directors and the audit committee.

The next section describes related research. This is followed by the introduction of the applied research method including qualitative data analysis. In section four our research results are presented. All concepts and their corresponding relations are listed and explained by expert statements. Furthermore, a concept map illustrates our results graphically. The paper closes with a conclusion and implications for future work.

2 Background and Related Research

All big audit firms consider process audits as an integral part of their present year-end audit approaches [St12, p.13] [Be97]. Audit standards also enforce an in-depth analysis of the organization's operations [Ia09]. In business process modeling literature audit-related concepts are mainly discussed from a risk management or compliance perspective. The paper of Rosemann and zur Muehlen [RZ05] is among the first to consider the concept *risk* in the business process modeling context. As existing modeling notations like ER, UML, Petri Nets or IDEF do not explicitly cover risk-related information they provide taxonomies and modeling techniques to embed risks in process models. These taxonomies present a supplement to a business process meta model consisting of *organizational goals*, *organization*, *process structure*, *information systems* (IS), and *data* [Zu04]. A meta model with related concepts is described in the UML class diagram notation by Karagiannis [Ka08] and suggested as an extension to an existing enterprise modeling approach [KMS07]. In addition this meta model comprises the concepts *account*, *control objective*, and *control*. Likewise, Strecker et al. [SHF11] considers an existing enterprise modeling approach and investigate its potential to support audit risk assessment. Domain specific terminology is conceptualized as an enhancement for the modeling approach. With their domain model for *internal controls* Namiri and Stojanovic [NS07b] introduce similar concepts. Moreover, they mention *recovery action* and specific types of controls like *IT general controls* and *application controls*. Jakoubi et al. [JTQ07] present an approach for risk-aware process modeling which introduces a separate modeling layer for *threats* and corresponding *counter measures*. These are linked to activities on the process model layer. However, all conducted research work mentioned before is mainly based on a review of relevant literature, standards and frameworks (e.g. COSO). Domain experts are not comprehensively involved.

With a strong focus on compliance Sadiq et al. [SGN07] present a modeling approach for *control objectives* using a specialized modal logic and a corresponding process model annotation. Their aim is to cope with complex *compliance requirements* of business processes at design time in a formalized manner. Related approaches are provided by Lu et al. [LSG08] and Governatori et al. [GMS06] [Go09] which help to test or measure the compliance of a business process model against a set of directives and rules. Namiri and Stojanovic [NS07a] develop another formalized definition of business process compliance which is based on the concepts *business process*, *risk*, *control* and *balance sheet account*. Earlier approaches use petri nets to formally model and evaluate *controls* within business processes for audit purposes [PP97] [CL03]. Goedertier and Vanthienen [GV07] describe a declarative approach for process modeling to capture the semantics of *internal and external regulations* with the help of business rules.

Based on a review of auditing literature and corresponding standards Carnaghan [Ca06] identified modeling concepts relevant for process level audit risk assessment like *process objectives*, *risks*, *controls* and *financial statement line items* (accounts), to name only a few. A number of commonly used business process modeling notations (e.g. UML, EPC, BPMN) were evaluated regarding their support of these concepts. This review revealed that no notation covers all identified concepts. Especially *controls* and linkages between different concepts were difficult to map to the modeling notations [Ca06]. Although providing a comprehensive list of audit relevant modeling concepts [Ca06] pointed out that only little empirical research has been performed on what modeling concepts are needed to support an understanding of business processes from an audit perspective. Regarding the related work mentioned above this appraisal holds true. Especially more complete semantics of these modeling concepts and reasonable ways for combining the information needed are seen as fruitful areas for further research [Ca06].

3 Research Method

The presented study uses an expert interview approach. The decision for expert interviews was based on the following characteristics:

1. The semi-structured nature of the interviews enable the interviewees to think about core concepts in a new way and link their experiences and perceptions [KB04], as well as to talk about new ideas and perspectives.
2. Expert interviews enable us to learn without preoccupation about the requirements experts have for process audits.
3. The authors are highly familiar with process audits. Therefore drawing up a guideline in advance was possible [Pf09, p.459].
4. According to Trinczek semi-structured interviews are the best choice when interviewing managers. This is because managers are generally in the position to ask questions rather than being asked. A guideline supports the special interview situation with managers in the best possible way [Tr09].
5. The results of expert interviews based on a guideline are already semi-structured and hence easier to analyze [Se97, p.13].

3.1 Expert Interview Conduct

17 process audit experts were interviewed throughout a five-month period (January 2012 to May 2012). Table 1 describes this sample in brief detail. Each interview lasted approximately 30 minutes up to one hour. We identified experts following the purposeful sampling approach by Patton. He lists different strategies. We decided to combine type five "*Typical case sampling: Illustrates or highlights what is typical, normal, and average*" and six "*Stratified purposeful sampling: Illustrates characteristics of particular subgroups of interest; facilitates comparisons*" [Pa90 p.182]. We are aware of the possible shortcomings of the sampling strategy proposed by Patton. Kaya and Himme state that: "*(…) the subjective judgment of the researcher about what is considered "important" or "typical" is to be considered*" [KH07, p.81]. This argument is countered by the perennial working experience of the authors.

The sampling population was defined according to the following two criteria: first, the individual had to be highly familiar with process audits: persons having work experience in the business process audit domain of more than five years met this requirement. This relatively low threshold seemed necessary because of the corporate structure of audit firms and internal audit departments: auditors are working on an operative level for around six years. By reaching the managerial level, the involvement in the operative execution of business process audits decreases rapidly and a high level understanding becomes more important. Both expert groups are essential for our research. For this reason around half of the interviewees are working on an operative level and therefore have a relatively short work experience of at least five years. Whereas experts working on a managerial level had at least seven years of work experience. The second criterion was the experts' employer: either one of the top five auditing firms (Big 4 (Deloitte, Ernst&Young, KPMG and PwC) and BDO) or internal audit departments of international companies. This requirement is based on the assumption that comprehensive process audits are generally more often performed in bigger companies.

Expert ID	Background	Interview Conduct	Duration
Expert 1	Operational	Face-to-Face	36 min
Expert 2	Operational	Face-to-Face	53 min
Expert 3	Management	Face-to-Face	66 min
Expert 4	Management	Telephone	88 min
Expert 5	Operational	Telephone	50 min
Expert 6	Operational	Face-to-Face	30 min
Expert 7	Operational	Face-to-Face	40 min
Expert 8	Management	Face-to-Face	64 min
Expert 9	Management	Face-to-Face	46 min

Expert ID	Background	Interview Conduct	Duration
Expert 10	Operational	Face-to-Face	35 min
Expert 11	Operational	Telephone	48 min
Expert 12	Management	Face-to-Face	50 min
Expert 13	Operational	Face-to-Face	54 min
Expert 14	Management	Telephone	34 min
Expert 15	Management	Telephone	63 min
Expert 16	Management	Face-to-Face	39 min
Expert 17	Management	Face-to-Face	40 min

Table 1: Summary details of the sample interviewed

The country of origin was not defined as a selection criterion; nevertheless all experts in this survey are German. This fact is not likely to have any influence on the conducted research, since international standards and internal international company guidelines force auditors to use the same approach worldwide [IIA12] [IFAC10]. Furthermore, gender specific aspects are knowledgeably excluded. Only a few publications are available covering the difference in interviewing female and male experts [Ab09][Li09]. Other gender specific attributes are fully neglected. To our knowledge none of the mentioned aspects have as yet been fully researched, thus we excluded them.

A target list of process audit experts from internal audit departments and top five audit firms was created and all experts were contacted via email. The guideline was pilot-tested with two persons from two different Big 4 audit firms. A face-to-face or telephone interview was then set up. Telephone interviews were conducted taking into account the suggestions from Christmann [Ch09]. All interviews were held with two researchers, one taking the lead, the other assisting to keep the interview flowing and taking notes. The guideline supported this intention with its open question design (see Exhibit 1).

Q1: Please introduce yourself directing particular attention to your professional career.

Q2: What is your understanding of a process audit and how do you describe the execution?

Q3: Please think of an ideal world: Which information do you need for a process audit?

a. *Which information do you need as "input" information?*
b. *How do you process this information?*
c. *Which information is provided as a result?*

Exhibit 1: Expert interview guideline questions

In total five of the 17 interviews were conducted by telephone. Differences in quality between face-to-face and telephone interviews were not expected [Ro76] and could not be noticed by us. The first guideline question was designed to get the interview started and get information about the experts background. The following two questions were designed to learn about the relevant process audit concepts and their relations.

3.2 Data Analysis

All but one interview were recorded and transcribed using the software F4. Expert 4 refused to be recorded. Therefore we had to take notes, which resulted in a significantly longer duration of the interview. The notes were complemented by a protocol from memory directly written after the interview. Audit concepts and their relations were coded according to the method suggested by Myers [My09, p.167]. We also took Strauss and Corbin [SC90] and further explanations in [Ke05] into account. "*Codes are tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study*" [My09, p.167]. The coding was done in MAXQDA. Following a bottom-up approach all inferences were derived only from the transcribed interviews, not taking into account known audit concepts from literature and work experience. In a first step one author coded a sample of transcripts and built up a code book as suggested by Ryan and Bernard [RB00]. This was validated by a second author. Differences were discussed and eliminated. Afterwards all interviews were independently coded by two authors, both of whom are knowledgeable in terms of data coding. Only marginal differences were noted. In case of a discrepancy between the results a joint coding was done. Following this, a content analysis as a quantitative method for analysing qualitative data was performed. According to [My09, p. 172] this analysis seeks to apply statistical techniques to a coded text. In our case we counted the occurrence of concepts and their relations in the expert interviews. These numbers are used to identify key concepts and relations. Both results are depicted in a comprehensive concept map.

4 Analysis and Research Findings

In this section we present audit concepts and their relations mentioned by our experts. At this stage it is not our intention to provide complete semantics for each concept. This would be beyond the scope of this paper. Our focus is to point out which concepts are considered most relevant for process audits. The absolute majority of the experts highlighted exactly the same audit concepts. In the following all concepts are described in detail especially focusing on their relations. For this purpose we intentionally use citations to minimize the influence of our interpretation on the expert statements.

4.1 Concepts and Relations

Controls: *Controls* were one of the most frequently mentioned concepts in our interviews. Not only because of their frequent occurrence but also due to the number of relations to other concepts, controls seemed to be among the most important concepts. All 17 experts agreed about the purpose of controls: "*controls contrast with risks (...). The auditor needs to assess to which degree the controls mitigate the identified or supposed risks*" (ex. 12). Experts stress that "*(…) only identified key controls get tested. Those are the significant controls*" (ex. 13). They further distinguish between "*(…) manual and automated controls in information systems*" (ex. 2). Automated controls were especially highlighted in the context of "*application controls, access rights, security, etc*" (ex. 4). Furthermore, "*it is important how the control is embedded in the* *organization, which departments are affected, who is control owner, who is accountable, who is responsible and who executes the control*" (ex. 15). Besides their organizational embedding, the integration of controls in business *processes* was frequently mentioned: "*A process audit is mainly a controls audit, the process is just a link between controls*" (ex. 14). This is mainly because *"controls are (just) process activities*" (ex. 3).

Process: As expert 3 stated: "*activities in companies can result in an accounting transaction - buying goods for example: the transaction would either be the order, the goods receipt, the invoice receipt or the payment. The latter three trigger postings on different accounts also called financial statement line items (FSLI). As a whole these in turn compose the financial statements*" (ex. 3). This relation between processes and the financial statements is seen by nearly all experts. The relation was mostly stated when explaining how the scoping of processes is done: "*Basically what I do is going backwards: starting from the financial statements I assess which processes post on my material FSLI in scope*" (ex. 8). Along with this approach the idea of Major Classes of Transactions (MCOTs) was mentioned by four experts. These MCOTs describe *"(…) different variants of processes, e.g. a process has different inputs and therefore different process branches are run through. Let me give you an example: the feed-in remuneration*[1] *heavily depends on whether you run photovoltaics, a thermal power plant or wind power plants. The accounting system will automatically differentiate between these cases*" (ex. 2). Process Key Performance Indicators (KPIs) represent one specialty. They were just highlighted by experts working as internal auditors.

Organization: As described in section "controls", nearly all experts emphazised the importance of *organizational* aspects and *controls*. Due to the association of controls and processes, experts correspondingly set *processes* and the *organizational structure* in relation. However, the number of occurrences of this relation was rather low most probably because of the higher importance of controls. It is commonly accepted that even nowadays, different departments and therefore employees are involved in diverse roles in processes and controls. Hence, organizational aspects are also important from an audit perspective. Analogous to controls, "*persons responsible and/ or accountable for processes, are so-called process-owners*" (ex. 7). The importance is not least based on the fact that "*staff works in processes and executes transactions. These transactions result in financial entries and then end up in the financial statements*"(ex. 3).

Risk: Again all experts named the concept of *risk* as one of the most important. But within this concept some rather different perspectives were described. On the one hand audit risks (detection, control and inherent risk) and on the other hand risks referring to the three COSO objectives categories (Operations, Financial Reporting, Compliance, old COSO Cube of 1992) [CO92] were mentioned. "*The audit risk is defined as the risk that material misstatement is not detected*" (ex. 9). This risk can be broken down into the following: "*the inherent risk describes material misstatement of FSLI, (...) the next stage is the control risk, namely the risk that misstatement will not be prevented or detected by the entity's internal controls. The last risk is the detection risk, namely the risk that we (as auditors) won't detect this misstatement*" (ex. 14). "*Using the COSO model, there*

[1] compensation for electricity fed into the grid

are ultimately three types of risk: first, the risk of misstatement in financial disclosure, second, the compliance risk, i.e. violations of legal requirements, last there are business risks that do not affect your financial statements directly, or violate any laws, but might make you buy over-expensive goods. External auditors generally don't care about it, but it is eventually harmful for your business" (ex. 2). Additionally, we would like to present one-risk subcategory in detail - interfaces. They were repetitively mentioned in the context of risk. "*You could discuss risk in the context of interfaces (...). Especially if you have different IS (...) the audit can get pretty complicated (...), in other words interfaces also pose considerable IS risks*" (ex. 12).

Financial Statements: 16 out of the 17 experts described *financial statements* as one of the key concepts. The only expert not mentioning financial statements in the interviews was from an internal audit department responsible for process audits with a strong focus on performance and compliance. Thereby, the strongest relation was to *risks* and *processes*, as indicated in the following example: "*Account groups included in the financial statements and the FSLI themselves have an inherent risk of misstatement in respect of the company's net assets, financial position and results of operations (…). This inherent risk has to be addressed in our audits. By applying controls in his processes the client already addresses this risk (...). We have a risk assessment upfront each audit, just to evaluate the risks on financial statement basis. Depending on the easiest way of auditing each FSLI in scope, we decide how to proceed. Mostly a controls audit (process audit) does make sense*" (ex. 14). Again, this statement reflects the frequently stated view that processes "fill" the financial statements. A further mentioned link of the concept financial statements is to *IS*. "*There are no middle or big companies preparing their financial statements without the support of IS. (...) For this reason every auditor has to ask himself how the financial statements are produced, viz. data storage, software, information systems, etc. supporting this process have to be taken into account*" (ex. 3). Another expert states that "*financial statements have to be IS based in the future in different countries because of legal requirements, like the German "E-Bilanz"* (ex. 8).

Information Systems (IS): In the description of the procedures of a process audit 16 experts mentioned *IS* as one of the key concepts which need to be assessed at the beginning of each process audit. The following concepts were mentioned in this context: "*Beginning with the scoping, the identification of audit relevant processes, organizational aspects like responsibilities, and supporting IS are most important*" (ex. 2). Thus, "*ideal process audits consider both aspects: IS and human interaction*" (ex. 8). A further facet of IS lies in its relation to *data*: "*IS play an important role when it comes to audit evidence. Nowadays most of the documents are stored digitally*" (ex. 1). Besides the pure data (storage), "*IS are also supporting processes*" (ex. 11). It should also be noted that it is important "*where the data is stored, where this data is first recorded, so to say - which information system is the first one -, how does the data get aggregated, processed and reported*" (ex. 15). "*This data-flow is always driven by IS. At least I haven't seen anything different in ages*" (ex. 3).

Materiality: 14 experts pointed out the importance of *materiality* as means to focus only on areas with significant *risks* and the corresponding business *processes* during financial audits. "*Materiality threshold is applied to the financial statements to identify significant FSLI and business transactions. Business processes affecting these significant FSLI are categorized as significant and should be subject to a process audit*" (ex. 4). "*This is done upfront during the scoping phase*" (ex. 2).

Audit Objective: 13 experts mentioned *audit objective* as a concept which can broadly be seen as an overarching goal of process audits. "*Which objective is striven for is up to the context the process audit is performed in*" (ex. 9). In general, depending on the subject of the audit diverse "*(…) risks are identified which need to be addressed during an audit*" (ex. 14). To cope with these risks the overall audit objective is broken down into audit objectives on a more detailed level referred to by the experts as assertions or control objectives. "*Within a financial audit the objective (…) is to provide assurance that business transactions which ran through the business processes throughout the financial year are correctly recorded in the financial statements*" (ex. 9). "*What is meant by correctly recorded is divided into a number of assertions. Five assertions are mapped to each balance sheet item: completeness, existence [and occurrence], valuation, rights and obligations as well as presentation and disclosure*" (ex. 14). "*To give an example: if we look at accounts receivables - there is a risk of loss. The corresponding assertion would be the valuation of receivables, e.g. there might be a control implemented that every seven days the CFO checks the aging structure of the receivables and decides for which open item to follow up. This control supports the assertion that receivables are correctly valued*" (ex. 14). "*Some audit companies do not distinguish between audit objective and assertions at all*" (ex. 4). "*Audit objective and assertions are used synonymously*" (ex. 14) in the context of financial audits.

Comparable to assertions "*a control objective is basically an intermediate level between control and risk. A control objective is derived from a risk and a related control in fact addresses the control objective*" (ex. 9). "*It is opposed to the risk and tries to mitigate the risk*" (ex. 16). "*Generally, there are several controls addressing the same control objective*" (ex. 14). An example for a control objective can be *"(…) to comply to legal requirements*" (ex. 13). Besides assertions which are closely linked to the financial statements three experts also mentioned a standard set of audit objectives related to data processing in business processes and supporting *IS*. "*In a process completeness, accuracy, validity, and restricted access need to be addressed by controls to ensure a sound processing and transfer of information*" (ex. 2).

Standards and Regulations: As an origin for audit objectives domain-specific frameworks (e.g. COSO), accounting standards, audit standards, generally accepted accounting principles (GAAP), legal requirements (e.g. Sarbanes Oxley Act), and company specific guidelines were mentioned by the experts. Standards in terms of "*policies and procedures guide the execution of processes and controls*" (ex. 10). "*Therefore a comprehensive process documentation ideally contains internal and external policies and procedures which are relevant for the process itself and included controls*" (ex. 10). "*Standards also guide an audit of a process*" (ex. 14) as "*specific audit procedures are obligatory due to audit standards*" (ex. 5). In this way "*audit standards operationalize the area of discretion for the auditor*" (ex. 4). Also domain-specific frameworks have a remarkable impact. "*If we have a look at the goals of COSO*

- efficient and effective operations, accurate financial reporting and compliance with laws and regulations – this is what a system of controls should look like. This of course affects our approach, e.g. which controls are relevant" (ex. 4).

Business Objective: "*In general processes support business goals and auditors also have specific expectations regarding business processes. Upstream to an audit the business environment of an auditee e.g. industry, competitive situation is analyzed to identify specific risks and areas exposed to high risks*" (ex. 3) and "*industry-specific processes*" (ex. 7) as well as "*target figures of a process*" (ex. 13). "*The question is what the company wants to achieve with this process and which minimum requirements need to be met by the process with respect to the overall business objectives*" (ex. 9).

Data: Any kind of *data* which is produced by *processes* and processed manually or stored in *IS*, like documents, records and vouchers are also of relevance for conducting a process audit. "*For an auditor it is interesting how the data flow of a process is defined: where is information generated, where it is used resp. which information is necessary to perform a process activity especially control activities*" (ex. 9). At first "*data and documents are used to get an understanding of a process during a walkthrough of each process step*" (ex. 13). "*For example a purchase transaction - we look at the flow of the data from the first entry of a purchase order to a goods receipt and finally a corresponding invoice*" (ex. 14). "*Each process activity has input and output data*" (ex. 10) "*this can be used as evidence for the operation of a control and a process*" (ex. 9).

Audit result: The *audit result* for a *control* is twofold. "*Controls are assessed regarding their design effectiveness and their operating effectiveness*" (ex. 9). "*Design effectiveness answers the question: is the control properly designed and implemented to support the addressed control objective?*" (ex. 4). In a second step "*it has to be assessed if the control was performed as described regularly throughout the audit period. This is called operating effectiveness*" (ex. 2). The operating effectiveness is tested *"(…) based on a sample approach. System reports, documents and system configuration for past business transactions of this process are examined to determine if the control has been performed as designed*" (ex. 9). "*Design effectiveness not only refers to controls but also to the process level. All controls of a process can be perfectly designed but significant risks - the process is exposed to - are not mitigated. In this case there is an issue with the process*" (ex. 15). Hence, when conducting process audits auditors need a comprehensive view of controls on the model layer as well as on the instance layer. The former is necessary to test the design effectiveness, the latter is needed for the operating effectiveness testing of controls.

4.2 Concept Map

A *concept map* is a graphical representation where nodes represent concepts and links connecting nodes reflect relationships between concepts [ST08]. It is a tool to organize and symbolize knowledge [NC08][Za11]. This mapping technique is generallly used to elicit cognitive structures that individuals or groups used to interpret a specific domain [ST08]. Siau and Tan emphasizes that concept maps help to design complex structures and externalize expert's knowledge [ST08]. They list several examples in the IS development domain using this mapping technique e.g. for conceptual modeling or technical communication. In our context concepts can be considered as key terms of the process audit domain derived from the expert interviews. Against this background the identified concepts and their relations are depicted in a concept map shown in Figure 1.

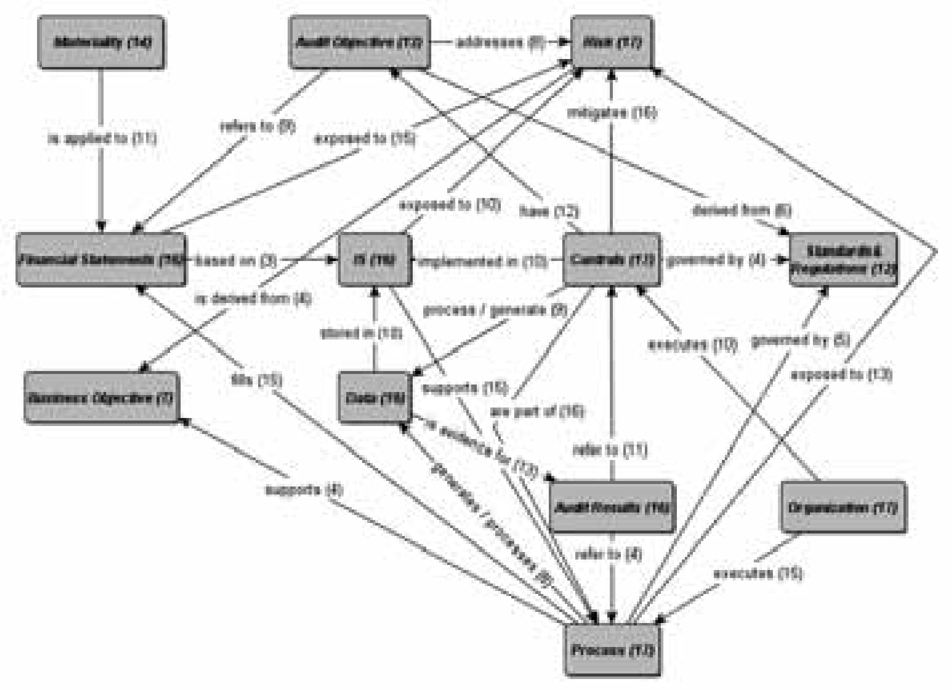


Figure 1 Concept map for audit relevant information

As a final step we provided the aggregated results including the concept map to the experts for review. Only minor remarks were given as feedback. In general the experts agreed with all concepts and their relations presented in this paper.

5 Conclusion and Future Research

We conducted semi-structured expert interviews with 17 domain experts from big audit firms and internal audit departments of international companies to reconstruct key concepts of the audit domain especially focusing on process audits. These interviews were transcribed and independently open coded by two of the authors. Subsequently all concepts were set into relation based on the expert statements. The contribution of the paper is twofold: key information requirements for process audits were conceptualized. These concepts were then set into relation and presented in a concept map. Our work was motivated by the lack of empirically grounded requirements engineering.

As a result of our analysis we identified the key concepts *processes*, *controls,* and *risks*. They were mentioned by all experts and have the most relations to other concepts. But also the remaining concepts: *organization, financial statements, IS, audit result, materiality, data,* and *audit objective* were stated by at least three-quarter of the experts. Only *business objectives* and *standards and regulations* were mentioned less. As said in section "2 Background and Related Research" the concepts process, business objective, organization and, information systems mentioned by the experts are already considered in existing enterprise modeling approaches. As suggested by Strecker et al. [SHF11] and Karagiannis [Ka08] research work done in this area can be also be applied to the audit domain. In addition more domain-specific concepts like a*udit objective, risk* and *control* turned out to be beset with diverse meanings. This needs to be considered when modeling these concepts.

However, we also had some unexpected findings especially regarding the relations. The number of mentioned relations broadly varies from three to 16. We interpret this unequally distribution as a strong indication of complexity for the topic in focus. Particularly the relations between "*standards&regulations and controls*" (four), "*process and business objectives*" (four), „*risk and business objective*" (four), "*process and audit result*" (four), and "*financial statements and IS*" (three) were only rarely referred to. The only differences between internal and external auditors we noticed was the usage of process KPIs and the weight shifted away from financial statements related risks to business risks.

Based on our results different future research opportunities arise. The most obvious to us is a further investigation in the topic of audit concepts. Due to the qualitative nature and according shortcomings of interviews, we' would like to verify our results by expanding our research with a quantitative approach. In this context, a comparison of research already conducted with our results would be meaningful. Another possible research action might be the creation of an ontology. Last but not least the incorporation of our results into a domain specific data model and domain specific modeling language could be one of the next logical steps.

The underlying project for this paper is funded by BMBF (German Federal Ministry of Education and Research) and supervised by PT-DLR (project references: 01IS10041B).

6 References

[Ab09] Abels, G.: Expert interviews in political science. A secondary-analytical reflection on gender theory and policy studies effects: Experteninterviews Theorien, Methoden, Anwendungsfelder. Verlag für Sozialwissenschaften, Wiesbaden, 2009; pp. 159–179.

[AG07] Ahlemann, F.; Gastl, H.: Process model for an empirically grounded reference model construction. In (Fettke, P.; Loos, P. Eds.): Reference Modeling for Business Systems Analysis. IGI Global, Hershey, PA, 2007; pp. 77–97.

[Be97] Bell, T.: Auditing Organizations Through a Strategic-Systems Lens. The KPMG Business Measurement Process. University of Illinois Press, Urbana Ill, 1997.

[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, 2006, 7; pp. 170–204.

[CL03] Chen, K. T.; Lee, R. M.: Knowledge-based evaluation of internal accounting control systems - a pattern recognition approach: Proceedings of the American Accounting Association Conference, Honolulu, Hawaii, 2003.

[CO92] Committee of Sponsoring Organizations of the Treadway Commission, C.: Internal Control - Integrated Framework, New-York, 1992

[Fr07] Frank, U.: Evaluation of Reference Models. In (Fettke, P.; Loos, P. Eds.): Reference Modeling for Business Systems Analysis. IGI Global, Hershey, PA, 2007; pp. 118–140.

[Ch09] Christmann, G.: Telefonische Experteninterviews: ein schwieriges Unterfangen. In (Bogner, A.; Littig, B.; Menz, W. Eds.): Experteninterviews: Theorien, Methoden, Anwendungsfelder. VS Verlag für Sozialwissenschaften, 2009.

[GMS06] Governatori, G.; Milosevic, Z.; Sadiq, S.: Compliance checking between business processes and business contracts: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference, 2006. Hong Kong, 2006; pp. 221–232.

[Go09] Governatori, G. et al.: Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. In (Ardagna, D. et al. Eds.): Business Process Management Workshops. Springer Berlin Heidelberg, 2009; pp. 5–17.

[GV07] Goedertier, S.; Vanthienen, J.: Declarative Process Modeling with Business Vocabulary and Business Rules. In (Meersman, R.; Tari, Z.; Herrero, P. Eds.): On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. Springer Berlin / Heidelberg, 2007; pp. 603–612.

[Ia09] International Auditing and Assurance Standards Board (IAASB). (2009). ISA 315, “Identifying and Assessing the risks of Material Misstatement through Understanding the Entity and its Environment” and related appendices.

[ICS05] IEEE Computer Society.: Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): 2004 Version. IEEE Computer Society Press, 2005.

[IFAC10] International Federation of Accountants, I.: Handbook of International Quality Control, Auditing, Review, Other Assurance, and Related Services Pronouncements, 2010.

[IIA12] The Institute of International Auditors, I. A.: Global Technology Audit Guide 1 – 16.

[JTQ07] Jakoubi, S.; Tjoa, S.; Quirchmayr, G.: Rope: A Methodology for Enabling the Risk-Aware Modelling and Simulation of Business Processes: ECIS 2007 Proceedings, 2007.

[Ka08] Karagiannis, D.: A Business process Based Modelling Extension for Regulatory Compliance: Multikonferenz Wirtschaftsinformatik, 2008; pp. 1159–1173.

[KB04] Kramp, M. K.; Bennett, K.: Exploring life and experience through narrative inquiry. In Foundations for research Methods in education and the social science, 2004; pp. 1–22.

[Ke05] Kelle, U.: "Emergence" vs. "Forcing" of Empirical Data? A Crucial Problem of "Grounded Theory" Reconsidered. In Forum Qualitative Sozialforschung, 2005, 6.

[KH07] Kaya, M.; Himme, A.: Möglichkeiten der Stichprobenbildung. In (Albers, S. et al. Eds.): Methodik der empirischen Forschung, 2007; pp. 79–88.

[KMS07] Karagiannis, D.; Mylopoulos, J.; Schwab, M.: Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. In 15th IEEE International Requirements Engineering Conference, 2007. RE '07, 2007; pp. 315–321.

[Li09] Littig, B.: Interviews mit Experten und Expertinnen: Überlegungen aus geschlechtertheoretischer Sicht. In (Bogner, A.; Littig, B.; Menz, W. Eds.): Experteninterviews: Theorien, Methoden, Anwendungsfelder. VS Verlag für Sozialwissenschaften, 2009.

[LSG08] Lu, R.; Sadiq, S.; Governatori, G.: Measurement of Compliance Distance in Business Processes. In Information Systems Management, 2008, 25; pp. 344–355.

[My09] Myers, M. D.: Qualitative Research in Business&Management. Sage Publications, 2009.

[NC08] Novak, J. D.; Cañas, A. J.: The theory underlying concept maps and how to construct and use them. In Technical Report IHMC CmapTools 2006-01 Rev 2008-01, 2008, Florida Institute for Human and Machine Cognition Pensacola Fl,

[NS07a] Namiri, K.; Stojanovic, N.: A formal approach for internal controls compliance in business processes: 8th Workshop on business process modeling, development, and support, Trondheim, Norway, 2007; pp. 1–9.

[NS07b] Namiri, K.; Stojanovic, N.: Pattern-Based Design and Validation of Business Process Compliance. In (Meersman, R.; Tari, Z. Eds.): On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Springer Berlin / Heidelberg, 2007; pp. 59–76.

[Pa90] Patton, M. Q.: Qualitative evaluation and research methods. Sage Publications, 1990.

[Pf09] Pfadenhauer, M.: Das Experteninterview. In (Buber, R.; Holzmüller, H. H. Eds.): Qualitative Marktforschung. Gabler, 2009; pp. 449–461.

[PP97] Pitthan, D. K. J.; Philipp, D. W. I. M.: Einsatz von Petri-Netzen für die Aufnahme, Dokumentation und Analyse Interner Kontrollsysteme im Rahmen der Jahresabschlussprüfung. In Petri-Netze zur Modellierung verteilter DV-Systeme–Erfahrungen im Rahmen des DFG-Schwerpunktprogramms "Verteilte DV-Systeme in der Betriebswirtschaft, 1997; pp. 87–104.

[RB00] Ryan, G. W.; Bernard, H. R.: Data Management and Analysis Methods. In (Denzin, N.; Lincoln, Y. Eds.): Handbook of Qualitative Research. Thousand Oaks, CA, 2000.

[Ro76] Rogers, T. F.: Interviews by Telephone and in Person Quality of Responses and Field Performance. In Public Opinion Quarterly, 1976, 40; pp. 51–65.

[Ru03] Russell, J.: The process auditing techniques guide. ASQ Quality Press, MIL, 2003.

[Ru06] Ruhnke, K.: Business Risk Audits: State of the Art und Entwicklungsperspektiven. In Journal für Betriebswirtschaft, 2006, 56; pp. 189–218.

[RZ05] Rosemann, M.; Zur Muehlen, M.: Integrating Risks in Business Process Models: Proceedings of the 16th Australasian Conference on Information Systems, 2005; p. 50.

[SC90] Strauss, A. L.; Corbin, J. M.: Basics of qualitative research. Grounded theory procedures and techniques. Sage Publications, Newbury Park, Calif, 1990.

[Se97] Seidman, I.: Interviewing as Qualitative Research. Teachers College Press, NY 1997.

[SGN07] Sadiq, S.; Governatori, G.; Namiri, K.: Modeling Control Objectives for Business Process Compliance. In (Alonso, G.; Dadam, P.; Rosemann, M. Eds.): 5th International Conference on Business Process Management. Springer, Brisbane, 2007; pp. 149–164.

[SHF11] Strecker, S.; Heise, D.; Frank, U.: Prolegomena of a modelling method in support of audit risk assessment - Outline of a domain-specific modelling language for internal controls and internal control systems. In Enterprise Modelling and Information Systems Architectures, 2011, 6; pp. 5–24.

[ST08] Siau, K.; Tan, X.: Use of cognitive mapping techniques in information systems development. In Journal of Computer Information Systems, 2008, 48; pp. 49–57.

[St12] Stuart, I. C.: Auditing and assurance services. An applied approach. McGraw-Hill Irwin, New York, NY, 2012.

[Tr09] Trinczek, R.: Wie befragt man Manager? Methodische und methodologische Aspekte des Experteinterviews als qualitativer Methode empirischer Sozialforschung. In (Bogner, A.; Littig, B.; Menz, W. Eds.): Experteninterviews: Theorien, Methoden, Anwendungsfelder. VS Verlag für Sozialwissenschaften, 2009.

[Za11] Zanibbi, L. R.: A Concept Map for Management Control System Design. In Journal of Management Policy and Practice, 2011, 12; pp. 54–61.

[Zu04] Zur Muehlen, M.: Workflow-based process controlling. Foundation, design, and application of workflow driven process information systems. Logos-Verl, Berlin, 2004.

The Projected TAR and its Application to Conformance Checking

Johannes Prescher[1], Matthias Weidlich[2], Jan Mendling[1]

[1]Institute for Information Business, Wirtschaftsuniversität Wien, Austria
[johannes.prescher|jan.mendling]@wu.ac.at
[2]Technion - Israel Institute of Technology, Haifa, Israel
weidlich@tx.technion.ac.il

Abstract: Relational semantics of business process models have seen an uptake in various fields of application. As a prominent example, the Transition Adjacency Relation (TAR) has been used, for instance, to conduct conformance checking and similarity assessment. TAR is defined over the complete set of transitions of a Petri net and induces order dependencies between pairs of them. In this paper, we consider TAR in a more general setting, in which the order dependencies shall be derived only for a subset of projected transitions. We show how to derive this projected variant of TAR from the reachability graph of a reduced Petri net. We elaborate the projected TAR for conformance checking in a case from industry.

1 Introduction

Business process models capture activities of a business process and the way their execution is coordinated to achieve a certain goal [Wes07]. Process models are an important means for process improvement and process conformance. In fact, the analysis of business processes is often traced back to an analysis of process models. Conclusions drawn for process models then allow for more effective process support.

Recently, relational semantics of process models have seen a particular uptake for answering analysis questions. For instance, behavioural relations have been the basis for checking conformance between a process model and a process log [WPD+11], to assess the similarity of process models [ZWW+10], or to manage process model variants [SWMW12]. Most relational semantics capture order dependencies for pairs of actions, or transitions in Petri net terms. Such order may be grounded on direct successorship of transitions, as proposed by the Transition Adjacency Relation (TAR) [ZWW+10], also referred to as a footprint [vdA11]. TAR captures direct successorship based on all transitions. Once a business process is captured, however, only a subset of the transitions of a model may have actual business semantics, i.e., represent activities of the business process. Other transitions may be considered to be silent steps, needed purely for syntactically reasons. Still, these transitions affect the order dependencies between transitions that have business semantics. As a consequence, any analysis, conformance analysis in the setting of this paper, that is based on the standard notion of TAR is biased by the influence of these silent transitions.

In this paper, we provide a solution to this problem by presenting a projection of TAR. Given a set of transitions of a Petri net system, it lifts the order dependencies of TAR to these transitions, neglecting transitions that are not part of the projection. Our contribution is threefold. First, we define the novel variant of TAR, called *projected TAR (pTAR)*. Second, we show how it is derived by exploiting the state space of a Petri net once existing reduction rules have been applied. As such, we provide the basis for using TAR-based techniques in a broader context. Third, we present an experimental evaluation based on an industry case.

The remainder of the paper is structured as follows. Section 2 illustrates the problem of applying TAR defined over all transitions with an example and Section 3 gives formal preliminaries. In Section 4, we define the projected TAR and elaborate on its derivation. We present experimental results on applying the projected TAR to conformance checking in Section 5. We review related work in Section 6, before Section 7 concludes the paper.

2 Background

Business process models are typically defined using conceptual modelling languages such as BPMN or EPCs. These languages tend to be well accepted by business professionals due to their intuitive representation of process semantics. As a downside though these languages are often not readily equipped with execution semantics. Therefore, formal analysis is typically conducted using the theoretical concepts of Petri nets. One specific instance of such analysis is conformance checking. This type of analysis is concerned with the problem to both a) quantify the degree of execution conformance of an execution log with the behaviour as defined by the process model and to b) point to those parts of the model which are violated by the cases.

On a technical level, the application of Petri net concepts requires the mapping from, for instance, BPMN as a conceptual model to a corresponding Petri net. In general, this mapping can be performed for each BPMN construct in an automated way yielding semantically equivalent Petri net components. In their work [DDO08], Dijkman et al. define such a mapping for a subset of BPMN constructs. Although basic elements such as activities can be easily mapped, more advanced constructs such as error subprocesses can result in Petri net components of considerable complexity. Further, the mapping of many BPMN constructs produces silent transitions which will not be observable in the execution log of the process. As an an example, consider the AND-split of BPMN. It is typically translated using a silent transition that produces tokens of each place representing the start of the branches diverging from the AND-split. Such transitions on the Petri net level do not have business semantics. More complicated components with silent transitions stem from, for instance, exception handling in BPMN.

To illustrate this mapping, consider the BPMN process model shown in Figure 1. It represents the way in which IT Service Management is performed at a center of an IT service provider. The process starts with the creation of an issue. Subsequently, we observe two parallel branches. The first path contains the customer extension which will be executed

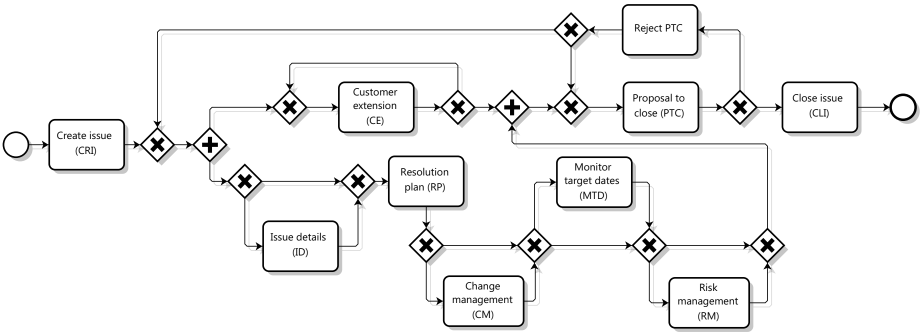


Figure 1: BPMN model of the SIMP process

at least once, optionally multiple times. The second path contains several optional activities: issue details may be updated, a resolution plan may be created, change management activities may be executed, target dates may be monitored and risk management tasks may be documented. After these optional activities, the parallel paths are synchronised and a proposal to close the issue has to be filed. If the issue is resolved, it may be closed. Otherwise, the proposal to close has to be rejected. In case of rejection, either a new proposal to close has to be stated or the process starts again before the two parallel paths are introduced (except the creation of an issue).

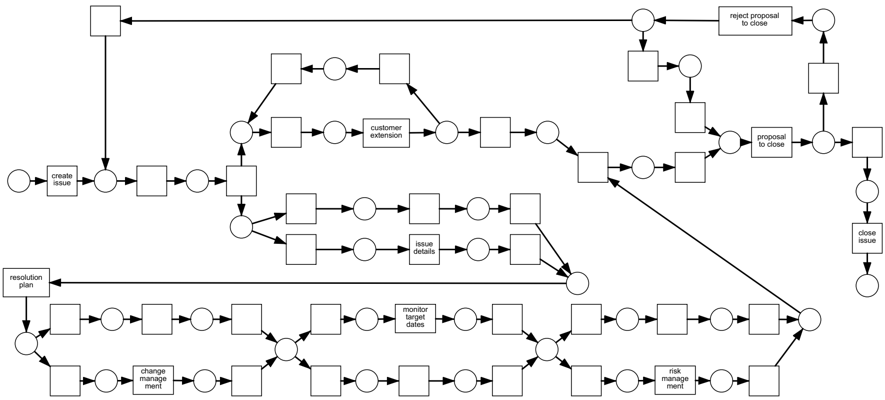


Figure 2: Petri net model of the SIMP process

Using the mapping rules defined in [DDO08], we can construct a Petri net that exactly captures the behaviour represented in the BPMN model. The net is depicted in Figure 2. We observe the following. First, each activity of the BPMN process is represented as a single transition. Second, each gateway of the BPMN model is represented by a component of silent transitions and places which represent its semantics. For instance, the exclusive-OR gateway prior to *issue details* is represented by two silent transitions which either activate the *issue details* path or the silent path. The transitions are silent, as they represent an implicit decision which is, in case of BPMN, not modelled explicitly.

There is a problem if we aim at using the constructed Petri net for analysis that builds on the Transition Adjacency Relation (TAR). TAR defines the set of ordered pairs of transitions (t_1, t_2) that can be executed one after another. Two transitions are part of this relation if there is an execution sequence of the process model, in which t_1 is directly followed by t_2. If we would simply calculate the TAR relation for the Petri net depicted in Figure 2, we would have to define unique identifiers for each of the silent transitions. The TAR relation would then represent, among others, that *resolution plan* can be followed by a silent transition, and that this same silent transition can lead to *change management*. This is problematic in case we want to analyse execution logs, because the option to execute *resolution plan* first and then directly next *change management* is not visible in the TAR relation. Hence, it is not simply possible to neglect TAR relations that refer to silent transitions. On the other hand, it is also not possible to derive new TAR relations in a transitive way. The idea here would be to conclude on (t_1, t_3) in TAR if we observe (t_1, t_2) and (t_2, t_3) in TAR. Such a derivation rule, however, requires assumptions on the absence of behavioural anomalies and does not work for short loops and non-free-choice constructs.

Accordingly, we need a technique to derive the notion of a projected TAR (pTAR). The calculation of the pTAR has to take into account that we are interested only in a particular subset of transitions (projection set). We want to characterise the traces of the Petri net as if occurrences of transitions that are not in the projection set would have been deleted.

3 Preliminaries

We first clarify notions and notations for Petri net systems. Second, we present the existing definition of the Transition Adjacency Relation (TAR).

3.1 Petri Net Systems

Petri nets, in particular classes such as workflow nets [vdA98], are often used to capture process models. We mentioned earlier that many process description languages, such as BPMN and EPCs, may be at least partly be transformed to Petri net systems, cf. [LVD09].

Definition 1 (Net)**.** A *net* is a tuple $N = (P, T, F)$ with P and T as finite disjoint sets of places and transitions, and a flow relation $F \subseteq (P \times T) \cup (T \times P)$.

We write $X = P \cup T$ for all nodes. For $x \in X$, $\bullet x := \{y \in X \mid (y, x) \in F\}$ is the pre-set, $x\bullet := \{y \in X \mid (x, y) \in F\}$ is the post-set, and $\bullet(x\bullet) := \{z \in X \mid y \in X \wedge (x, y) \in F \wedge (z, y) \in F\}$.

To define semantics, we need notations for sequences. A *sequence* over a set S of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \rightarrow S$. If $\sigma(i) = s_i$ for $i \in \{1, \ldots, n\}$, we write $\sigma = \langle s_1, \ldots, s_n \rangle$. The set of all finite sequences over S is denoted by S^*.

Let $N = (P, T, F)$ be a net. $M : P \mapsto \mathbb{N}$ is a *marking* of N, $\mathbb{M}$ denotes all markings of N.

$M(p)$ returns the number of *tokens* in place p. For any transition $t \in T$ and any marking $M \in \mathbb{M}$, t is *enabled* in M, denoted by $(N, M)[t\rangle$, iff $\forall\, p \in \bullet t\ [\ M(p) \geq 1\ ]$. Further, we identify the flow relation F with its characteristic function on the set $(P \times T) \cup (T \times P)$. Then, marking M' is reachable from M by *firing* of t, denoted by $(N, M)[t\rangle(N, M')$, such that $M'(p) = M(p) - F(p, t) + F(t, p), p \in P$, i.e., one token is taken from each input place of t and one token is added to each output place of t.

A sequence of transitions $\sigma = \langle t_1, \ldots, t_n\rangle, n \in \mathbb{N}$, is a *firing sequence*, iff there exist markings $M_0, \ldots, M_n \in \mathbb{M}$, such that for all $i \in \mathbb{N}$, $1 \leq i \leq n$ it holds $(N, M_{i-1})[t_i\rangle(N, M_i)$. We say that σ is enabled in M_0, denoted by $(N, M_0)[\sigma\rangle$. For any two markings $M, M' \in \mathbb{M}$, M' is reachable from M in N, denoted by $M' \in [N, M\rangle$, if there exists a firing sequence σ leading from M to M'. Firing of σ in M is denoted by $(N, M)[\sigma\rangle(N, M')$. A *net system*, or *system*, is a pair $S = (N, M_i)$, where N is a net and M_i is the *initial marking* of N.

3.2 Transition Adjacency Relation

The Transition Adjacency Relation captures behavioural characteristics of a net system by means of an ordering relation defined over the Cartesian product of transitions [ZWW+10]. It captures direct succession of two transitions in some firing sequence of a Petri net system.

Definition 2 (TAR)**.** Let $S = (N, M_i)$ be a system with $N = (P, T, F)$. The *TAR* $> \subseteq T \times T$ contains a pair (x, y), iff there exists a firing sequence σ with $(N, M_i)[\sigma\rangle$ such that $\sigma(i) = x$ and $\sigma(i + 1) = y$ for some $1 \leq i$.

Note that, by definition, TAR and the inverse relation $>^{-1} = \{(x, y) \mid (y, x) \in >\}$ partition the Cartesian product of transitions.

4 Conformance Checking with the Projected TAR

In this section, we present the notion of the *projected TAR (pTAR)*. It captures behavioural characteristics while projecting transitions that are given as a projection set. We first define the pTAR. Then, we present a derivation algorithm which uses reduction rules and state space search. Finally, we apply pTAR for conformance checking.

4.1 The Projected TAR

The projected TAR defines the set of transition pairs that follow each other directly in some projected firing sequence of the net system.

Definition 3 (pTAR)**.** Let $S = (N, M_i)$ be a system with $N = (P, T, F)$. Let $T' \subseteq T$ be a set of transitions called projection set. The *projected TAR* induced by T', $>_{T'} \subseteq T' \times T'$ contains a pair (x, y), iff there exists a firing sequence σ with $(N, M_i)[\sigma\rangle$, such that

$\sigma(i) = x$, $\sigma(j) = y$ for some $1 \leq i < j$ and $\sigma(k) \notin T'$ for all $i < k < j$.

Considering only the Cartesian product of transitions in the projection set, we see that pTAR actually extends TAR. That is, for transitions in the projection set additional successorships may be identified.

Property 1. For a system $S = (N, M_i)$ with $N = (P, T, F)$ and $T' \subseteq T$ holds $(> \cap (T' \times T')) \subseteq >_{T'}$.

The property follows directly from the definition of the relations. Every pair of transitions in the projection set that is part of TAR is, by definition, also part of pTAR.

4.2 Derivation

We approach the derivation of pTAR in two stages. First, we reduce the original Petri net, then we identify the pTAR using the state space techniques. Figure 3 illustrates the set of reduction rules we consider. They were adapted in [vdADO+08] from the liveness and boundedness preserving reduction rules by Murata [Mur89]. The rules eliminate transitions that are not included in the projection set as follows.

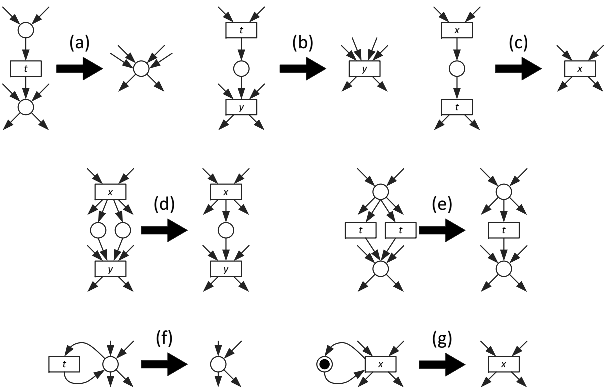


Figure 3: Illustration of reduction rules, t is not part of the projection set, as adapted in [vdADO+08].

Definition 4 (Reduction Rules). Let $S = (N, M_i)$ be a system with $N = (P, T, F)$. Let $T' \subseteq T$ be a projection set. For system S, a reduced system is derived by one of the following rules:

(a) If there exists $p \in P$ and $t \notin T'$ such that $p\bullet = \{t\}$ and $\bullet t = \{p\}$, then $S_a = (N_a, M_i)$ with $N_a = (P_a, T_a, F_a)$ such that
  - $P_a = P \setminus \{p\}$,
  - $T_a = T \setminus \{t\}$,
  - $F_a = F \cup \{(n_1, n_2) | n_1 \in \bullet p \wedge n_2 \in t\bullet\} \setminus \{(n_1, n_2) | n_1 \in \{p, t\} \vee n_2 \in \{p, t\}\}$.

(b) If there exists $p \in P$, $t \notin T'$ and $y \in T$ such that $\bullet p = \{t\}$ and $p\bullet = \{y\}$, then $S_b = (N_b, M_i)$ with $N_b = (P_b, T_b, F_b)$ such that

- ∘ $P_b = P \setminus \{p\}$,
- ∘ $T_b = T \setminus \{t\}$,
- ∘ $F_b = F \cup \{(n_1, y) | n_1 \in \bullet t\} \setminus \{(n_1, n_2) | n_1 \in \{p, t\} \vee n_2 \in \{p, t\}\}$.

(c) If there exists $p \in P$, $t \notin T'$ and $x \in T$ such that $\bullet p = \{x\}$,$\bullet t = \{p\}$ and $p\bullet = \{t\}$, then $S_c = (N_c, M_i)$ with $N_c = (P_c, T_c, F_c)$ such that

- ∘ $P_c = P \setminus \{p\}$,
- ∘ $T_c = T \setminus \{t\}$,
- ∘ $F_c = F \cup \{(x, n_2) | n_2 \in t\bullet\} \setminus \{(n_1, n_2) | n_1 \in \{p, t\} \vee n_2 \in \{p, t\}\}$.

(d) If there exists $p_1, p_2 \in P$ such that $\bullet p_1 = \bullet p_2$ and $p_1\bullet = p_2\bullet$, then $S_d = (N_d, M_i)$ with $N_d = (P_d, T_d, F_d)$ such that

- ∘ $P_d = P \setminus \{p_2\}$,
- ∘ $T_d = T$,
- ∘ $F_d = F \setminus \{(n_1, n_2) | n_1 = p_2 \vee n_2 = p_2\}$.

(e) If there exists $t_1, t_2 \notin T'$ such that $\bullet t_1 = \bullet t_2$ and $t_1\bullet = t_2\bullet$, then $S_e = (N_e, M_i)$ with $N_e = (P_e, T_e, F_e)$ such that

- ∘ $P_e = P$,
- ∘ $T_e = T \setminus \{t_2\}$,
- ∘ $F_e = F \setminus \{(n_1, n_2) | n_1 = t_2 \vee n_2 = t_2\}$.

(f) If there exists $t \notin T'$ such that $\bullet t = t\bullet$, then $S_e = (N_f, M_i)$ with $N_f = (P_f, T_f, F_f)$ such that

- ∘ $P_f = P$,
- ∘ $T_f = T \setminus \{t\}$,
- ∘ $F_f = F \setminus \{(n_1, n_2) | n_1 = t \vee n_2 = t\}$.

(g) If there exists $p \in P$ such that $M_i > [i]$ and $\bullet p = p\bullet$, then $S_g = (N_g, M_i)$ with $N_f = (P_g, T_g, F_g)$ such that

- ∘ $P_g = P \setminus \{p\}$,
- ∘ $T_g = T$,
- ∘ $F_g = F \setminus \{(n_1, n_2) | n_1 = p \vee n_2 = p\}$.

Once the system is reduced according to these rules, we can calculate the projected TAR based on the reachability graph as outlined in Algorithm 1.

The algorithm uses the auxiliary data structure $TxM \subseteq T \times \mathbb{M}$ to keep track of which transition $t \in T$ led to a certain marking $M \in \mathbb{M}$. Once such pair has been investigated it is added to the structure $vTxM \subseteq T \times \mathbb{M}$ representing visited pairs. We initialise the algorithm with all markings reachable from the initial marking by firing a single transition (line 2). Then, all pairs of transitions and markings are evaluated (lines 3 to 19). A pair is selected and the structures TxM and $vTxM$ are updated (lines 4 to 6). Then, for each transition that is enabled in the respective marking, we check whether it is part of

Algorithm 1: Derivation of projected TAR based on Reachability Graph

Input: $S = (N, M_i)$, a net system with $N = (P, T, F)$, $T' \subseteq T$, a projection set.
Output: $>_{T'}$, the projected TAR of S induced by T'.

```
1  >_{T'}, TxM, vTxM ⟵ ∅;
2  foreach t ∈ T with (N, M_i)[t⟩ do TxM ⟵ TxM ∪ {(t, M)} with (N, M_i)[t⟩(N, M);
3  while TxM ≠ ∅ do
4      select (t, M) ∈ TxM;
5      TxM ⟵ TxM \ {(t, M)};
6      vTxM ⟵ vTxM ∪ {(t, M)};
7      T_e ⟵ {t ∈ T | (N, M)[t⟩};
8      foreach t_e ∈ T_e do
9          if t_e ∈ T' then
10             if t ∈ T' then >_{T'} ⟵ >_{T'} ∪ {(t, t_e)};
11             if (t_e, M') ∉ vTxM with (N, M)[t_e⟩(N, M') then
12                 TxM ⟵ TxM ∪ {(t_e, M')} with (N, M)[t_e⟩(N, M');
13             end
14         end
15         else if (t, M') ∉ vTxM with (N, M)[t_e⟩(N, M') then
16             TxM ⟵ TxM ∪ {(t, M')} with (N, M)[t_e⟩(N, M');
17         end
18     end
19 end
```

the projection set (line 9). If so, the projected TAR relation may be updated (line 10) and we proceed by adding new pairs of transitions and reachable markings to TxM for investigation (line 12). If not, then we add the pair comprising the original and the marking reached by firing the transition that is not part of the projection set to structure TxM (line 16). Intuitively, this captures the fact that the marking may be reached by firing the original transition and a sequence of silent transitions, i.e., transitions that are not in the projection set. Further, whenever TxM is updated, we need to check whether a pair has been processed already to ensure termination of the algorithm.

4.3 Checking pTAR Conformance

Having introduced the pTAR, we turn to its application for conformance checking. For a transition t of a given net system, we define its conformance set and its violation set. The projected TAR defines the set of permissible successors of a transition, providing the basis for defining both these sets. Intuitively, the conformance set of t comprises all transitions that are allowed to occur in some observable execution sequence. In contract, the violation set contains the transitions that are not allowed to directly succeed t. Formally, we define:

Definition 5 (Conformance Set and Violation Set)**.** Let $S = (N, M_i)$ be a system with $N = (P, T, F)$. Let $T' \subseteq T$ be a projection set and $pTAR$ its projected TAR.

- ◦ The *conformance set* for $t \in T$ is defined as $conf(t, pTAR) = \{x | (t, x) \in pTAR\}$.
- ◦ The *violation set* for $t \in T$ is defined as $viol(t, pTAR) = T' \setminus conf(t, pTAR)$.

Assume that we are given a net system $S = (N, M_i)$, $N = (P, T, F)$, such that the projection set $T' \subseteq T$ comprises all transitions that have a business meaning. Hence, those transitions are expected to occur in the execution log that captures the observed behaviour of the business process. Formally, such an observed execution sequence is a finite sequence $\sigma \in T'^*$ over the transitions in the projection set.

To detect deviations of σ from the behaviour as defined in S, we proceed as follows. For the observed execution sequence $\sigma = \langle t_1, \ldots, t_n \rangle$, let t_i be a transition with $1 \leq i < n$. Then, transition t_{i+1} succeeding t_i in σ is either in the conformance set $conf(t_i, pTAR)$ or in the violation set $viol(t_i, pTAR)$ of t_i. The conformance according to pTAR is achieved once all considered succeeding transitions are in the respective conformance sets.

Definition 6 (pTAR Conformance)**.** Let $S = (N, M_i)$ be a system with $N = (P, T, F)$. Let $T' \subseteq T$ be a projection set and $pTAR$ its projected TAR. An observed execution sequence $\sigma = \langle t_1, \ldots, t_n \rangle \in T'^*$ is valid according to *pTAR conformance* iff for all $1 \leq i < n$ it holds that $t_{i+1} \in conf(t_i, pTAR)$.

Once an observed execution sequence is not valid according to pTAR conformance, all non-empty violation sets of transitions of this sequence hint at the behavioural deviations.

5 Evaluation

In this section, we apply our conformance checking technique in an industry case. The data of this case relates to a Security Incident Management Process (SIMP). The process model and the corresponding execution sequences are taken from [WPDM10]. The process and its execution log were captured on-site at an IT service providers centre during five years. Overall, the study contains 852 cases of process execution for the SIMP process. First, we will investigate the effect on the state space of applying reduction rules. Then, we present the results of conformance analysis.

5.1 The Effect of Petri Net Reduction

The original Petri net shown in Figure 2 in Section 2 includes several silent transitions that do not carry any business semantics, i.e., they do not represent business activities. We reduce the net by considering all transitions that represent business activities as part of the projection set and apply the presented reduction rules. While the original net contains 33 silent transitions, the resulting net is reduced and shows solely nine silent transitions. Table 1 lists measures which allow for the comparison of both Petri nets. Altogether, the size of the Petri net shrinks from 43 to 19 transitions, 38 to 14 places and 88 to 40 arcs. Relatively speaking, the amount of the nets components decreases by more than 50%.

(a) (b)

Figure 4: Reachability graph (a) of original Petri net and (b) of reduced Petri net

The significance of this reduction relates to the state explosion problem of Petri nets with concurrency [Val96]. As our projected TAR calculation relies on the reachability graph, it is important to note that the number of transitions and states decreases by more than 85% (see Figure 5.1). This decrease leads to a smaller state space and a more efficient calculation of conformance sets.

Table 1: Reduction of Petri nets complexity

Category	Original Net	Reduced Net
Petri net		
— transitions	43	19
— places	38	14
— arcs	88	40
Transition System		
— transitions	248	38
— states	121	18

5.2 Conformance Results

Turning to the conformance checking, we derive 68 distinct transition pairs representing undesirable sequences according to the violation sets of the respective transitions. The sequencing of these pairs was violated 1453 times in the overall amount of 852 different cases of the SIMP process. Successorship according to TAR and pTAR does not only relate to distinct transitions, but may relate a transition to itself. Nine out of the 68 pairs relate to such self-relations, such that these transitions must not be a direct successor of themselves. Within our study these nine pairs (13.2%) relate to 1205 violations (82.9%), whereas the remaining 59 pairs (86.8%) relate to only 248 violations (17.1%).

Figure 5 illustrates the results. For each transition, we state the absolute amount of violations (V) related to this transition, and its relative share with respect to all detected violations. In general, a high amount of violations for a specific transition indicates that the execution of this activity and its context is worth to be investigated in detail. Note that the point of violation does not allow for any implications regarding the current state of the process. A violation might stem from progressing with non permissible other transitions or from forbidden multiple executions. For the later case, it is striking that the transition *proposal to close* relates to 466 violations out of which 403 were forbidden multiple executions.

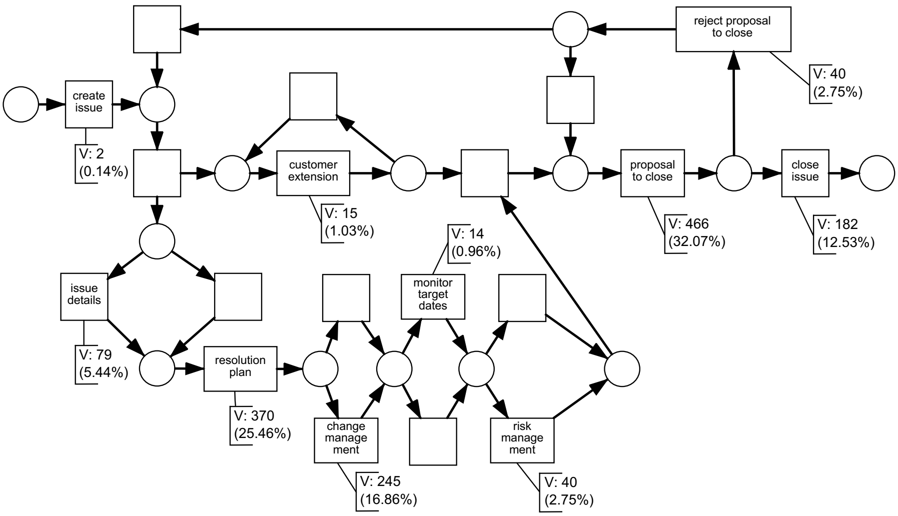


Figure 5: Reduced Petri net system of the SIMP

Altogether, we conclude that the projected TAR helps to discover activities that frequently relate to violations.

5.3 Results in Comparison to Existing Techniques

Albeit reduced, the net system for our case as visualised in Figure 5 still contains a several silent transitions. As such, application of the TAR to detect deviations between model and observed execution sequences will be biased as discussed in Section 2.

However, a similar yet different approach to conformance checking based on behavioural relations has been proposed in [WPD$^+$11]. It relies on the notion of behavioural profiles that do not capture direct successorship of transitions, but an indirect successorship. As such, they define a base relation that captures whether a transitions is *eventually* followed by another transition in some firing sequence of the net system, which avoids the problems induced by silent transitions. On the downside, this comes at the cost of lost precision. Relying on indirect successorship, e.g., means that transitions in cycles appear to be unordered since they may follow each other eventually in either order.

Table 2 provides further insights in this aspect by comparing the violations detected using the relations of the behavioural profile with the violations detected using pTAR. Here, the relative values are based on the joint result of both approaches. The provided ratio is supposed to visualise the effectiveness of the approaches.

Table 2: Detected violations for the case using different behavioural relations

Approach	Pairs with Violations	Violations	Violations/Pairs
BP	19 (21.84%)	201 (12.15%)	10.58
projected TAR	68 (78.16%)	1453 (87.85%)	21.37
$\sum$(BP + pTAR)	87	1654	19.01

According to Table 2 the technique presented in this paper is able to detect a significantly higher amount of process violations. A detailed look at the data of the case reveals the major reason for the observed deviations: 1205 out of 1453 violations are detected because our technique checks forbidden repetition of single transitions. Behavioural profiles, in turn, allow only for assessing whether a transition may occur only once or may be repeated. However, they lack the ability to check whether it single transitions may be repeated directly, without the occurrence of any other transition in between.

6 Related Work

The research presented in this paper is relates to the derivation of behavioural relations from process models and to conformance checking. Several sets of relations have been defined for capturing the behaviour of process models. The causality, conflict and concurrency relation have been proposed for Petri net systems based on unfoldings [McM95, ERV02]. The α-relations originally defined for mining processes [vdAWM04] have been adapted for process models, yielding the TAR [ZWW$^+$10] that is the starting point for our work. The behavioural profile and an efficient calculation for sound free-choice workflow nets is presented in [WMW11], and extended with a causality relation in [WPMW11]. All these relations can be calculated at varying degrees of complexity. Behavioural profiles can be determined in cubic time for certain net classes. Here, we use a state space technique to determine the projected TAR. Unfolding techniques might be applicable to improve performance.

Several approaches have been defined for conformance checking. Rozinat and van der Aalst introduce a fitness measure which builds on a state-based replay of execution sequences from a log [RvdA08]. The concept of a violation set shares some characteristics of negative events as discussed in [GDWM+11]. Earlier we mentioned that the relations of the behavioural profile may also be used for conformance checking [WPD+11]. This approach has been extended towards monitoring in [WZM+11]. We discussed that a downside of behavioural profiles is that they represent a behavioural abstraction, which has major implications for cyclic structures in particular. In contrast, our approach enables the monitoring of behaviour while relying on behavioural relations to precisely capture any behavioural deviation.

7 Conclusion

In this paper, we have presented an approach for conformance checking based on the projected TAR relation. Our contribution is the definition of the projected TAR and its calculation based on efficient reduction rules and the state space. We applied the technique for a service management process demonstrating its applicability. The advantage of our novel technique is a combination of an efficient representation of behaviour in terms of the projected TAR and higher precision in comparison to existing approaches.

In future research, we aim to improve the theoretical complexity of the calculation of the projected TAR. Several concepts including the process structure tree [VVK09] and the efficient calculation of the concurrency relation [Esp04] will be helpful to this end. We also aim to conduct further industry evaluations in the service management domain. The characteristics of this domain (process models available but not enforced, cases documented in ticketing systems) are perfect to challenge conformance checking techniques.

References

[DDO08] Remco M. Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and analysis of business process models in BPMN. *Inf. & Software Techn.*, 50(12):1281–1294, 2008.

[ERV02] Javier Esparza, Stefan Römer, and Walter Vogler. An Improvement of McMillan's Unfolding Algorithm. *Formal Methods in System Design*, 20(3):285–310, 2002.

[Esp04] Javier Esparza. A Polynomial-Time Algorithm for Checking Consistency of Free-Choice Signal Transition Graphs. *Fundam. Inform.*, 62(2):197–220, 2004.

[GDWM+11] S. Goedertier, J. De Weerdt, D. Martens, J. Vanthienen, and B. Baesens. Process discovery in event logs: An application in the telecom industry. *Applied Soft Computing*, 11(2):1697–1710, 2011.

[LVD09] Niels Lohmann, Eric Verbeek, and Remco M. Dijkman. Petri Net Transformations for Business Processes - A Survey. *T. Petri Nets and Other Models of Concurrency*, 2:46–63, 2009.

[McM95] Kenneth L. McMillan. A Technique of State Space Search Based on Unfolding. *Formal Methods in System Design*, 6(1):45–65, 1995.

[Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[RvdA08] Anne Rozinat and Wil M. P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.*, 33(1):64–95, 2008.

[SWMW12] Sergey Smirnov, Matthias Weidlich, Jan Mendling, and Mathias Weske. Action patterns in business process model repositories. *Computers in Industry*, 63(2):98–111, 2012.

[Val96] Antti Valmari. The State Explosion Problem. In *Petri Nets*, LNCS 1491, pages 429–528. 1996.

[vdA98] Wil M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[vdA11] Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[vdADO+08] Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, and Eric Verbeek. Conformance checking of service behavior. *ACM Trans. Internet Techn.*, 8(3), 2008.

[vdAWM04] Wil M. P. van der Aalst, A.J.M.M. Weijters, and Laura Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142, 2004.

[VVK09] Jussi Vanhatalo, Hagen Völzer, and Jana Koehler. The refined process structure tree. *Data Knowl. Eng.*, 68(9):793–818, 2009.

[Wes07] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.

[WMW11] Matthias Weidlich, Jan Mendling, and Mathias Weske. Efficient Consistency Measurement Based on Behavioral Profiles of Process Models. *IEEE Trans. Software Eng.*, 37(3):410–429, 2011.

[WPD+11] Matthias Weidlich, Artem Polyvyanyy, Nirmit Desai, Jan Mendling, and Mathias Weske. Process compliance analysis based on behavioural profiles. *Inf. Syst.*, 36(7):1009–1025, 2011.

[WPDM10] Matthias Weidlich, Artem Polyvyanyy, Nirmit Desai, and Jan Mendling. Process Compliance Measurement Based on Behavioural Profiles. In *CAiSE*, LNCS 6051, pages 499–514. 2010.

[WPMW11] Matthias Weidlich, Artem Polyvyanyy, Jan Mendling, and Mathias Weske. Causal Behavioural Profiles - Efficient Computation, Applications, and Evaluation. *Fundam. Inform.*, 113(3-4):399–435, 2011.

[WZM+11] Matthias Weidlich, Holger Ziekow, Jan Mendling, Oliver Günther, Mathias Weske, and Nirmit Desai. Event-Based Monitoring of Process Execution Violations. In *BPM 2011*, LNCS 6896, pages 182–198. 2011.

[ZWW+10] Haiping Zha, Jianmin Wang, Lijie Wen, Chaokun Wang, and Jiaguang Sun. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry*, 61(5):463–471, 2010.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005" – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik – Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics
GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der 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 Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^{2}CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf
INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

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