Christian Stary (Ed.)

LNBIP 104

# S-BPM ONE – Scientific Research

4th International Conference, S-BPM ONE 2012
Vienna, Austria, April 2012
Proceedings

Springer

Lecture Notes
in Business Information Processing 104

Christian Stary (Ed.)

# S-BPM ONE – Scientific Research

4th International Conference, S-BPM ONE 2012
Vienna, Austria, April 4-5, 2012
Proceedings

Springer

Volume Editor

Christian Stary
Johannes Kepler University Linz
Department of Business Information Systems
Communications Engineering
Linz, Austria
E-mail: christian.stary@jku.at

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-29132-6 e-ISBN 978-3-642-29133-3
DOI 10.1007/978-3-642-29133-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012933873

ACM Computing Classification (1998): H.3.5, H.4, D.2, H.5



*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

“Enabling Transition” was the theme of the 2012 S-BPM ONE conference. In the tradition of previous S-BPM ONE events, starting in 2009, it provided a lively interaction platform for (S-)BPM researchers, developers, educators, and practitioners. S-BPM as a discipline is characterized by a seamless approach toward the analysis, modeling, implementation, execution, and management of business processes with an explicit stakeholder focus (see also the S-BPM primer we have provided at the end of these proceedings). This year’s event not only intensified the discourse of S-BPM protagonists on other BPM paradigms, it also marked the starting point toward scientifically grounded BPM transformations. According to the results of the rigorous peer-review process, 12 contributions were selected out of 36 submissions, and included in these proceedings.

Egon Börger’s keynote demonstrates the benefits of a theoretically grounded approach to BPM, in particular when linking the concept of S-BPM to the abstract state machines (ASM) method. The contribution reveals the influence of high-level interpreting S-BPM-defined business processes on representation and model execution. Besides such grounding, researchers and developers target a variety of topics:

- Stakeholder-oriented business process management
- Enterprise modeling and cross-organizational engineering
- Role and communication management
- Information structure architecting
- Activity and agency
- Active knowledge modeling
- Formal BP semantics for modeling and processing
- Work flow design and management
- Control-driven BPM suite development and tool applications

The contributions address most of the life-cycle activities, in particular analyzing business objectives, subject behavior design and integration, and automating complex work procedures. Some tendencies enabling transitions seem to be significant for further research and development:

1. S-BPM triggers contextual design of processes and corresponding semantic processing. As such, the more general concept of System Thinking, as originally proposed by Peter Senge, influences all BPM life-cycle activities. It also allows BPM paradigms to interface alternative approaches beyond BPEL — a worthwhile transition in planning, designing, and operating business processes.
2. The development of the S-BPM modeling language reflects the trend toward contextual specification and semantic processing, as the approaches capturing BPM objectives and access issues demonstrate. Hence, related areas,

such as change management, can be addressed via S-BPM in a seamless but still structured way.

3. S-BPM can be understood as scalable and domain-independent approach when applied in practice. The scope of S-BPM applications and application domains can be widened in a seamless way, as demonstrated for complex service industries, such as hospital management (while capturing inherent peculiarities) and for industrial process control systems (when coupled with high-level business processes).
4. Economically plausible, but sophisticated architectures and infrastructures becoming common use, such as Web services and cloud computing, increasingly follow a choreographic approach. By utilizing S-BPM, not only can heterogeneous BPM approaches become part of integrated business settings, but also adapting business processes to novel settings and access facilities on-the-fly can be resolved effectively.

Experiencing the active engagement of BPM activists facilitates hosting this conference. However, the success of such an event relies on creative and constructive hands, most notably:

- The authors of the various contributions sharing their expertise
- The members of the international Program Committee reviewing each of the contributions thoroughly
- The Chair of the sessions handling the highly interactive presentation formats

Moreover, we need to thank the many persons running the conference facilities, and guiding us through the social program of S-BPM ONE 2012. Their efforts allowed us to elaborate ideas and network in rewarding settings. Special thanks go to the Institute of Innovative Process Management (I2PM, www.i2pm.net), serving as umbrella to a variety of S-BPM activities. It ensures continuity as well as adjustments of research and development.

Finally, we cordially thank Ralf Gerstner and Viktoria Meyer from Springer for their assistance and support in publishing these proceedings in the LNBIP series.

April 2012 Christian Stary

# Organization

## Executive Committee

### Conference Chair

| | |
|---|---|
| Christian Stary | Johannes Kepler University Linz, Austria |

### Organizing Chair

| | |
|---|---|
| Stefan Oppl | Johannes Kepler University Linz, Austria |

## Organizing Committee

| | |
|---|---|
| Stefan Oppl (Chair) | Johannes Kepler University Linz, Austria |
| Dominik Wachholder | Institute of Innovative Process Management, Germany |
| Werner Schmidt | Ingolstadt University of Applied Sciences, Germany |

## Program Committee

| | |
|---|---|
| Christian Stary (Chair) | Johannes Kepler University Linz, Austria |
| Franz Barachini | BIC-Austria, Austria |
| Thomas Bahlinger | FH Nürnberg, Germany |
| Reza Barkhi | Virginia Tech, USA |
| Noureddine Belkhatir | University of Grenoble-LIG France, France |
| Freimut Bodendorf | University of Erlangen-Nürnberg, Germany |
| Yeong-Long Chen | National Chiao Tung University, Taiwan |
| Anke Dittmar | University of Rostock, Germany |
| Peter Forbrig | University of Rostock, Germany |
| Alexander Gromoff | Moscow National Research University, Higher School of Economics (HSE), Russia |
| Lutz Heuser | AGT Group (R&D) GmbH, Germany |
| Ebba Thora Hvannberg | University of Iceland, Iceland |
| John Krogstie | Norwegian University of Science and Technology, Norway |
| Florian Lautenbacher | SysTec-CAx GmbH, Germany |
| Juhnyoung Lee | IBM T.J. Watson Research Center, USA |
| Christopher Lueg | University of Tasmania, Australia |
| Tansel Özyer | Ekonomi ve Teknoloji Üniversitesi, Turkey |
| Carlos Pedrinaci | The Open University, UK |
| Stefan Reinheimer | BIK GmbH, Germany |
| Gustavo Rossi | LIFIA. F. Informatica, UNLP, Argentina |

| | |
|---|---|
| Gabriele Saueressig | University of Applied Science Würzburg-Schweinfurt, Germany |
| Werner Schmidt | Ingolstadt University of Applied Sciences, Germany |
| Detlef Seese | Karlsruhe Institute of Technology, Germany |
| Robert Singer | FH Joanneum Graz, Austria |
| Renate Strazdina | Ernst&Young Baltic SIA, Latvia |
| Alexandra Totter | ByElement GmbH, Switzerland |
| Eric Tsui | The Hong Kong Polytechnic University, China |
| Gerrit van der Veer | Open University Netherlands, The Netherlands |
| Nikolas Vidakis | Technological Education Institution of Crete, Greece |
| James Weber | St. Cloud State University, USA |
| Cornelia Zehbold | Ingolstadt University of Applied Sciences, Germany |
| Erwin Zinser | FH Joanneum Graz, Austria |

## Sponsoring Institutions

Metasonic AG, Pfaffenhofen-Hettenshausen, Germany
VALIAL Solution GmbH, Ilmmünster, Germany
Infomedia Services G.m.b.H., Vienna, Austria
Competence Center on Knowledge Management, JKU, Linz, Austria
IANES (Interactive Acquisition, Negotiation and Enactment of Subject-Oriented Business Process Knowledge) – EU FP 7 Marie Curie IAPP
Reiner ConsSys, Cham, Switzerland
Format Werk GmbH, Gunskirchen, Austria

# Table of Contents

## Keynote



## Contributions

## Tutorial

# The Subject-Oriented Approach to Software Design and the Abstract State Machines Method*

Egon Börger

Università di Pisa, Dipartimento di Informatica,
Largo Bruno Pontecorvo 3, 56127 Pisa, Italy
boerger@di.unipi.it

**Abstract.** In [33, Appendix] we have shown that the system which implements the Subject-oriented approach to Business Process Modeling (S-BPM) has a precise semantical foundation in terms of Abstract State Machines (ASMs). The construction of an ASM model for the basic S-BPM concepts revealed a strong relation between S-BPM and the ASM method for software design and analysis. In this paper we investigate this relation more closely. We use the analysis to evaluate S-BPM as an approach to business process modeling and to suggest some challenging practical extension of the S-BPM system.

## 1 Introduction

The recent book [33] on the Subject-oriented approach to Business Process Modeling (S-BPM) contains a precise high-level definition, namely in terms of Abstract State Machines (ASMs), of the semantics of business process models developed using the S-BPM tool environment.[1] The construction of an ASM which rigorously describes the basic S-BPM concepts revealed an intimate relation between on the one side S-BPM, whose conceptual origins go back to Fleischmann's software engineering book [31, Part II], and on the other side the ASM method [27], a systems engineering method which too has been developed in the 90'ies of the last century by a community effort building upon Gurevich's discovery of the notion of ASM [41] (at the time called by various names, in 1994 'evolving algebras', for the historical details see [12] or [27, Ch.9]).

* The paper has been originally published in Düsterhöft, A., Klettke, M., Schewe, K.-D. (eds.) Conceptual Modelling and Its Theoretical Foundations. LNCS, vol. 7260, pp. 52-72. Springer, Heidelberg (2012).

[1] In the appendix, which is written in English, an ASM interpreter is defined for the behavior of such business process models. The software used to transform the pdf-file generated from latex sources into a Word document and printer-control-compatible format produced a certain number of partly annoying, partly misleading mistakes in the printed text. The interested reader can download the pdf-file for the correct text from [63].

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 1–21, 2012.

In this paper we investigate the striking methodological and conceptual similarities (Sect. 2) and some differences (Sect. 3) of these two independent developments. We propose to enhance the current S-BPM system by offering the modeler tool support for the use of the full ASM-refinement method which generalizes the refinement scheme S-BPM provides the software engineer with.

We use this analysis to evaluate S-BPM in terms of six well-known principles for reliable software development (Sect. 4), an evaluation which shows that S-BPM provides practitioners with suitable means to precisely and faithfully *capture* business scenarios and *analyze*, *communicate* and *manage* the resulting models.[2]

What nowadays is called S-BPM is really a version tailored for the development of business processes (BPs) of a more general subject-oriented software engineering method and environment for the development of concurrent systems proposed in [31, Part II] and called there SAPP/PASS: 'Structured Analysis of Parallel Programs' with a subject-oriented modelling language named 'Parallel Activities Specification Scheme'. We use invariably the today apparently prevailing term S-BPM to refer to Fleischmann's approach.

We assume the reader to have some knowledge of the basic concepts of at least one of the S-BPM [33] or the ASM methods [27].

## 2 Common Features of S-BPM and the ASM Method

The S-BPM and ASM methods share their main goal, namely to reliably link the human understanding of real-life processes to their execution by machines via some implementing software. In fact the ASM method is introduced in [27, p.1] by stating that

> 'The method bridges the gap between the human understanding and formulation of real-world problems and the deployment of their algorithmic solutions by code-executing machines on changing platforms.'

Similarly, a recent presentation of the S-BPM approach states for the 'transformation process of model descriptions to executable ones' [34, Sect.2, p.3-4] that:

> 'end-to-end control is what business stakeholders need to build process-managed enterprise' and that
> 'Any mapping scheme should allow propagating the information from a value chain perspective to a software-development perspective in a coherent and consistent way'.

We explain in this section that as a consequence both methods share three major methodological concerns for descriptions of (concurrent) processes:

[2] In [16] we showed that the OMG standard BPMN [50], the workflow patterns of the Workflow Pattern Initiative [1] and their (academic) reference implementation YAWL [45] fail to achieve this.

– the ground model concern (Sect. 2.1),
– the refinement concern (Sect. 2.2),
– the subject-orientation concern to make the executing agents and their distinct internal and external (communication) actions explicit (Sect. 2.3).

Also both come with 'a *simple scientific foundation*, which adds precision to the method's practicality [27, p.1]'.

Although the two methods realize these three concerns differently, due to the more focussed BPM target of the (current incarnation of the) S-BPM method and the different definitions in the two methods of what constitutes agent behavior (described by Subject Behavior Diagrams (SBDs) resp. ASMs, see Sect. 2.3), and although their scientific foundation comes from different sources, the similarities of the two approaches to software engineering are remarkable because 'the *ground model method* for requirements capture, and the *refinement method* for turning ground models by incremental steps into executable code' form together with the concept of ASMs 'the three constituents of the ASM method for system design and analysis' [27, p.13] through which the method

> '*improves current industrial practice* in two directions:
> – On the one hand by accurate high-level modeling at the level of abstraction determined by the application domain ...
> – On the other hand by linking the descriptions at the successive stages of the system development cycle in an organic and effectively maintainable chain of rigorous and coherent system models at stepwise refined abstraction levels.' [27, p.1]

### 2.1 Ground Model Concern

In the S-BPM literature there is no mention of the name 'ground model' (or 'golden model' as they are called in the semiconductor industry [57]) but the ground model concern is present. The ASM ground model method [8,9,11,13,15] is about constructing prior to code development, as specification for the code, models which are

> 'blueprints that describe the required application-content of programs ... in an abstract and precise form' and are 'formulated in terms of the application domain and at a level of detailing that is determined by the application domain' [15, Sect.1].

Thus ground models satisfy needs of different stakeholders, in particular the domain experts and the software designers. First of all the domain experts (e.g. analysts or users of BPs) need ground models for a '*correct development and understanding* by humans of models and their relation to the application view of the to-be-modeled BP' [16, Sect.5].[3] *Correctness* as used here (together with its companion concept *completeness*) is intrinsically not a mathematical notion, but an epistemological relation between a model and the piece of reality the model is intended to capture, a relation the application experts have to understand and only they (not the software technologists) can judge.

But then also the software designers need ground models, namely as a *complete* specification, where the completeness—every behaviorally relevant feature is stated—makes a correct implementation of the specification reliable. The reliability property links these two roles of ground models. It 'means that the appropriateness of the models can be checked by the application domain experts, the persons who are responsible for the requirements, and can be used by the system developers for a stepwise detailing (by provably controllable ASM refinement steps) to executable code.' [23, p.1923]

Therefore an approach for building satisfactory (i.e. correct, complete and consistent) ground models requires to have solved before 'a *language and communication problem* between the software designers and the domain experts or customers ... the parties who prior to coding have to come to a common understanding of "what to build"' [15, Sect.2.1.1]:

> 'The language in which the ground model is formulated must be appropriate to naturally yet accurately express the relevant features of the given application domain and to be easily understandable by the two parties involved.[4] This includes the capability to calibrate the degree of precision of the language to the given problem, so as to support the concentration on domain issues instead of issues of notation.'(ibid.)(See also the 'language conditions for defining ground models' formulated ibid., Sect.2.3.)

To solve this problem S-BPM starts from two observations of language theory [34, Sect.3, p.5]:

– 'When structuring reality, humans use subjects, predicates and objects.'
– 'humans use natural language structures as primary means to ensure mutual understanding'.

Consequently S-BPM aligns BP descriptions to those three constituents of elementary sentences in natural languages and to the coordination role of

[3] The request in [34, Sect.1,p.1] of a minimal 'semantic distance to human understanding' for S-BPM corresponds to the request for satisfactory ground model ASMs of a 'direct', coding-free relation between the basic domain elements (agents, objects, functions, properties, operations) and the corresponding ASM ground model items [9, Sect.6.2]. The ASM ground model method satisfies this request by offering 'The freedom to choose how to represent the basic objects and operations of the sytem under consideration' and by its attention to 'distinguish between concepts (*mathematical modelling*) and notation (*formalization*)' [9, Sect.5].

[4] The S-BPM literature speaks about 'duality of expressiveness' which is needed for the description language [34, Sect.2, p.4].

communication between subjects.[5] To stay close to natural language, where domain experts formulate process requirements, BP descriptions in S-BPM express the behavior of each subject involved in the BP (read: the agents which perform the described behaviors) as a sequence of possibly guarded basic ('internal') computation or ('external') communication actions of the following form (their content is discussed in Sect. 2.3):

$$\textsc{SbpmAction}(\mathit{Condition}, \mathit{subject}, \mathit{action}, \mathit{object}) = \\ \quad \textbf{if } \mathit{Condition}(\mathit{subject}) \textbf{ then } \mathit{subject}\ \textsc{Performs}(\mathit{action}) \textbf{ on } \mathit{object}$$

These basic S-BPM actions *mutatis mutandis* correspond to basic ASM transitions, even if the two methods have a different view on what is allowed, in general, to constitute an action and on their parallel resp. sequential execution (see Sect. 2.3 and 3.1). In fact in the S-BPM interpreter the ASM rule BEHAVIOR(*subj*, *state*)—which formalizes the execution by the *subj*ect of the action (called *service*(*state*)) associated with its SID-*state*—has exactly the above form, as the reader can check in [33, p.351].

In this way in S-BPM BPs are modeled using a precise language which is understandable by both parties, domain experts (analysts/managers/users) and software developers: it is constituted by elementary sentences which can be understood as (not formalized) natural language sentences, but nevertheless have a precise *operational* meaning (*modulo* a precise meaning of the constituent parts). The resulting BP ground models are as close to the intended real-world processes (read: their intuitive application-domain-views) as are the subjects, their actions and the objects which are chosen by the analyst (as BP model designers are called) to appear in the ground models. Thus the S-BPM approach offers for BPs an interesting solution to a challenge listed in [23, p.1924], namely 'supporting the extraction of ground model elements from natural language descriptions of requirements'.

The 'abstract operational' character of ASM ground models, which makes them directly executable, mentally by definition as well as mechanically by appropriate execution engines, has been recognized in [9, Sect.7] as crucial for the needed '*experimental validation* of the application-domain-based semantical correctness for *ground models*' [15, p.226]. It is a key criterion also for S-BPM, expressed as follows in [34, Sect.1, p.2]:

> 'The novelty of the approach can be summarized by two key benefits, resulting for stakeholders and organization developers:
> 1. Stakeholders need only to be familiar with natural language ... to express their work behavior ...[6]
> 2. Stakeholder specifications can be processed directly *without further transformations*, and thus, experienced as described'.

---

[5] Notably *communication* and *coordination* appear as two of the seven categories of the *Great Principles of Computing* [29].

[6] Obviously such a natural language expression of the work behavior has to be sufficiently precise, in particular to avoid misunderstandings that may arise from cultural differences among the stakeholders.

The *ASM ground model method* realizes the ground model concern in a similar way, but tailored for a more general system engineering setting, using the more comprehensive notion of ASM compared to S-BPM's SBDs as they are used to describe the behavior of BP subjects, see below. Not to repeat for an explanation of this difference what has been described in various articles on the theme [8,9,11,13,15] we invite the reader to read the systematic epistemological discussion of the method in [15]. We limit ourselves here to point to a typical ASM ground model 'at work' S-BPM experts may be interested in, namely the interpreter model for SBDs in [33, Ch.12 and Appendix] (see also [63]). It illustrates the characteristic properties of ASM ground models by exhibiting the direct, strikingly simple and easy to grasp correspondence between the S-BPM concepts and their mathematical, operational formalization by ASMs.

**Scientific Foundation.** The just mentioned ASM ground model for an SBD-interpreter constitutes the mathematical part of the scientific foundation of S-BPM. The epistemological part of its foundation is rooted in language theory. The ASM method has its simple scientific foundation directly in mathematical logic and its epistemological roots in a generalized Church-Turing thesis (see Sect. 2.3).

### 2.2 Refinement Concern

In S-BPM the specification of the processes which constitute a BP model is done in two steps. For each process its SBD (also called PASS graph) describes only the sequence in which the executing subject performs its basic actions. The detailed content of these actions is specified by refinements which describe 'the local variables of a process and the operations and functions defined on the local variables' [31, p.206].

Four types of operations and functions are considered, reflecting the classification of actions described in more detail in Sect. 2.3. Two types of *communication* are specified by describing a) the parameters of the communicated messages and b):

– for to-be-received messages the state change they yield, i.e. their 'effect ... on the values of the local variables, depending on the values of the message parameters and the current values of the loca variables' (ibid.)[7]
– for to-be-sent messages the definition of their content depending on the current state, i.e. 'how the values of the message parameters are obtained from the values of the local variables' (ibid.)[8]

So-called *internal operations* are specified by describing their update effect on the current state (here the values of the local variables), where one is allowed

[7] This is described in the S-BPM interpreter model by the RECORDLOCALLY submachine of ASYNC(*Receive*) and SYNC(*Receive*) [33, p.367-368].
[8] This is described in the S-BPM interpreter model by the functions *composeMsg* and *msgData* of the PREPAREMSG$_{Send}$ submachine [33, p.361].

to use so-called *internal functions* (whose applications in the current version of S-BPM are not distinguished any more as separate kind of operations), that is mathematical (side-effect-free), in ASM terminology dynamic functions (i.e. functions whose result for given arguments depends on the current state).

To define these specifications and their implementation in S-BPM the approach 'is open for the integration of existing and proved development methods' [31, p.199] and in particular 'all the object oriented concepts can be applied' (ibid., p.206). These two programming-practice inspired refinement types in S-BPM (Pass graph refinement and its implementation) are instances of the concept of ASM refinement.

The ASM refinement method was conceived in the context of modelling the semantics of ISO Prolog by ASMs [5,6,7,18] (surveyed in [8]), when I was challenged by Michael Hanus to also develop an ASM for the Warren Abstract Machine (WAM)—an early virtual machine whose optimization techniques changed the performance of Prolog to a degree that made practical applications feasible—and to prove the compilation of ISO Prolog to WAM code to be correct. The challenge was solved by refining the Prolog interpreter model in 12 proven to be correct refinement steps to a WAM interpreter model [24,25,26]. The adopted refinement concept (which has been implemented in KIV for a machine verification of the WAM correctness proof [55,56,52,53,54]) is described in detail in [14]. It

- supports sequences of refinement steps whose length depends on the complexity of the to be described system, and
- links the refinement steps in a documented and precise way so that their correctness can be objectively verified.[9]

Since the ASM refinement notion is in essence more general than the programming-focussed one used in S-BPM, we discuss the details in Sect. 3.2.

### 2.3 Subject-Orientation Concern

In this section we elucidate for the S-BPM and ASM methods the feature which gave the name to S-BPM and is emphasized in the comparative analysis in [31, Ch.5], [33, Ch.14],[34, Sect.4] as distinctive with respect to traditional system description methods, namely the primary role of agents (called subjects) which execute step by step two distinct kinds of actions following the 'program' (behavioral description) each agent is associated with: communications ('external' actions) and 'internal' actions on corresponding objects.

**Agents.** *Subjects* are placed into the center of S-BPM process descriptions as the 'active elements' of a process which 'execute functions offered by the passive elements' (i.e. objects of abstract data types) [31, p.199] and have to be identified

[9] It is an important aspect for certifiability that these verifications are documented to become repeatable by mathematical 'experiment' (read: proof checking). See Sect. 3.3.

as first elements of any process description: ‘start with identifying the involved subjects and after that define the behaviour specifications of acting parties’ [34, Sect.3, p.8]. The ASM method shares this view: in the list of the six ‘Fundamental Questions to be Asked’ when during requirements capture one starts to construct an ASM ground model the first question is:

> Who are the system *agents* and what are their relations? [27, p.88]

This corresponds to the fact that by its very definition an ASM is a family of pairs $(a, Pgm(a))$ of different agents, belonging to a set (that may change at runtime), and the (possibly dynamically associated) programs $Pgm(a)$ each agent executes [27, Def.6.1.1].[10] S-BPM has the same definition: ‘An S-BPM *process* ... is defined by a set of subjects each equipped with a diagram, called the *subject behavior diagram* (SBD) and describing the behavior of its subject in the process.’ [33, p.348] In both definitions we see multiple agents whose behavior is to execute the (sequential) program currently associated with them. Since this happens in a concurrent context, S-BPM and the ASM method both classify the basic ‘actions’ an agent can perform in a program step by their role for information exchange among the agents, as we are going to explain now.

**Classification of Agent Actions.** In S-BPM the ‘actions’ agents perform when executing their program are of two kinds, to ‘exchange information and invoke operations’ [31, p.372]. Information exchange is understood as sending or receiving messages. The information exchange actions are named ‘external’ because they involve besides the executing subject also other, ‘external’ subjects. The invoked other operations are understood as agent-‘internal’ (read: communication-free) computations on given objects [31, p.205].

Similarly the ASM method explicitly separates agent-internal operations from external data exchange operations (communication) with other cooperating agents, namely through the so-called classification of locations (i.e. containers of abstract data). Agent-internal operations come in the form of read/writes of so-called *controlled* locations which are performed under the complete and exclusive control of the executing agent. Data exchange (communication with cooperating agents) comes in the two forms of a) reading so-called *monitored* locations that are written by the cooperating agents (an abstract form of receiving messages sent by other agents) and b) writing so-called *output* locations to be read by the cooperating agents (an abstract form of sending messages to other agents).

In the interaction view of an S-BPM subject behavior diagram each internal or communication action counts as one step of the corresponding *subject*, namely to perform what is called the *service* associated with the subject in the given state. In the detailed (refined) interpreter view of the *subject* as defined in [33, Appendix, Sect.3] this ‘abstract’ interaction-view-step usually is rather complex

[10] To name the agent can be omitted (only) in the special case where a single ASM is contemplated (which may interact with an environment that is considered as run by one other agent).

since it is constituted by the sequence of 'detailed-view-steps' performed by the *subject* to execute the underlying internal or communication action— more precisely in the S-BPM interpreter it is the sequence of the START and all PERFORM steps made by the *subject* to execute its BEHAVIOR(*subject*, *SID_state*), otherwise stated the sequence of detailed steps *subject* performs from the moment when it enters the *SID_state* corresponding to the action (read: the associated *service*) until the moment when it exits that state to enter the *SID_state*′ corresponding to the next action, see [33, p.351].

The ASM method started out to provide in full generality the means to abstract into one single-agent step an entire internal computation which may be needed to perform an action in a given state. Therefore one has to separately describe the interaction the considered agent may have with the cooperating agents in its environment to perform the action, namely receiving data from cooperating agents before it starts the abstract step and sending data to cooperating agents after (probably as a result of) the abstract step. The agent's sending interactions are collectively incorporated into its one abstract step, namely as updates of all corresponding output locations; this is without loss of generality given the parallel nature of a single ASM step which performs simultaneously an entire set of location updates. Analogously the agent's receiving interactions directly preceding (and probably influencing) its abstract step are collectively described by a separate so-called 'environment' step which precedes the agent's abstract step and is assumed to be executed by another agent representing the environment of the considered agent; this environment step performs simultaneously all the relevant updates of the corresponding monitored locations, thus completing the definition of the state in which the considered agent performs (the internal part of) its abstract step (see the formal definition in [27, Def.2.4.22, p.75]).

The difference in the technical S-BPM/ASM realization of the identical concept of distinguishing internal and external 'actions' is a result of the different origins of the two methods. The motivating target of S-BPM was to incorporate in an explicit and practically feasible way into the software engineering techniques of the time the missing high-level concept of communication between process agents, in particular for developing BPs where communication is fundamental to control the actions of the cooperating agents. Therefore it was natural to develop an orthogonal communication concept (inspired by CCS [49] and CSP [44]) which is compatible with the principal (at the time prevailingly object-oriented) programming concepts and their implementation so that it can be integrated in a modular way into any practical software engineering method. This led to the interesting input-pool-based S-BPM notion of a synchronous or asynchronous communication (send or receive) 'step' as *pendant* to and *à la pari* with any internal computation 'step'. The notion of an ASM the development of the ASM method started from grew out of an epistemological concern, namely to sharpen the Church-Turing thesis for 'an alternative computation model which explicitly recognizes finiteness of computers' [39,40] (see [12],[27, Ch.9] for the historical details). Therefore it was natural to abstract for the definition of what constitutes an ASM step from any particular form of communication mechanism and to represent a communication (receive or send) action abstractly the same way as any other basic computational action, namely as reading the value of an abstract 'memory location' resp. as updating (writing) it—clearly at the price of having to define an appropriate practical communication model where needed, a task Fleischmann accomplished for S-BPM with his input-pool concept. This concept provides an interesting contribution to the challenge listed in [23, p.1923] to develop 'practically useful patterns for communication and synchronization of multi-agent ASMs, in particular supporting omnipresent calling structures (like RPC, RMI and related middleware constructs) and web service interaction patterns.'[11]

**Behavior of Agents.** In S-BPM the behavior of a single agent is represented by a graph of the Finite State Machine (FSM) flowchart type (called SBD or PASS graph) which 'describes the sequences in which a process sends messages, receives messages and executes functions and operations' [31, p.207]. This corresponds exactly to the so-called control-state ASMs [27, Sect.2.2.6] and their FSM-flowchart like graphical display[12] so that not surprisingly the high-level S-BPM interpreter in [33, Appendix, Sect.7] for the execution of SBDs is defined as a control-state ASM.

## 3 Differences between S-BPM and the ASM Method

In this section we discuss three major differences between the S-BPM and the ASM method. They concern the notion of *state* and state *change* (update) by actions of agents (Sect. 3.1), the notion of *refinement* of models (Sect. 3.2) and the *verification* concern which helps in the ASM method to increase the system reliability and to reduce the amount of experimental system validations (Sect. 4). Through these features the ASM method offers the practitioner additional possibilities for certifiably correct design of software-intensive systems, although we see no reason why they could not be included into S-BPM, as we are going to suggest, to increase the degree of reliability of S-BPM-designed BPs by certifiable correctness.

### 3.1 Notion of State and State Change

**State.** As we have seen in Sect. 2.2, S-BPM shares the traditional programming view of states: 'the values of all local variables define ... the local state of a

[11] The various theoretical communication concepts surveyed in [42] appear to have been defined to suit parallel and so-called interactive forms of the ASM thesis and seem to have had no practical impact.

[12] Control-state ASMs have been introduced in [11] as 'a particularly frequent class of ASMs which represent a normal form for UML activity diagrams and allow the designer to define machines which below the main control structure of finite state machines provide synchronous parallelism and the possibility of manipulating data structures.' [27, p.44]

process' [31, p.206]. In contrast, 'the notion of ASM *states* is the classical notion of mathematical *structures* where data come as abstract objects, i.e. as elements of sets (also called *domains* or *universes*, one for each category of data) which are equipped with basic operations (partial *functions* in the mathematical sense) and predicates (attributes or relations).'[27, p.29] In logic these structures, which have been formulated as a concept by Tarski [60] to define the semantics of first order logic formulae, are also called Tarski structures.[13] The relevant fact for the modelling activity is that the sets and functions which form the state of an ASM can be chosen in direct correspondence with the to-be-modelled items of the application domain, tailored with 'the greatest possible *freedom of language*' [9, Sect. 5] to the intended level of abstraction of the model and 'avoiding the formal system straitjacket' (ibid.). Thus ASM states realize an advice from a great authority: 'Data in the first instance represent abstractions of real phenomena and are preferably formulated as abstract structures not necessarily realized in common programming languages.' [62, p.10]

To provide a characteristic example we can refer to the abstract elements and functions which appear in the ASM model for S-BPM [33, Appendix] as part of the interpreter state, like all the SBD-graph structure related items, the *service*s associated with SID-states and their completion predicate *Completed*, *inputPool* with its related functions, the different sets providing *Alternative*s together with their *select*ion functions, message related functions to *composeMsg* from *msgData*, etc.

Also the object oriented slightly more complex version of the programming view of states as defined above, which comes with the suggestion to use object oriented techniques for the specification of PASS graph refinements [31, p.210], is an instance of the ASM notion of state since 'the instantiation of a relation or function to an object $o$ can be described by the process of parameterization of, say, $f$ to the function $o.f$, which to each $x$ assigns the value $f(o,x)$.'[27, p.29][14]

**State Change.** The most general kind of a basic action to change a structure or algebra (i.e. a set of functions) appears to be that of a *function update*, i.e. change the value of a function at given arguments, which has the following form:

$$f(t_1,\ldots,t_n) := t$$

Such updates, executed by an agent (denoted by **self**) under appropriate conditions which guard the application of ASM rules:

$$\textsc{AsmRule}_{\mathbf{self}}(\mathit{Condition}, \mathit{Updates}) = \mathbf{if}\ \mathit{Condition}\ \mathbf{then}\ \mathit{Updates}$$

are exactly what constitutes the basic action of an ASM agent in a *state*, where $f$ is an arbitrary $n$-ary function symbol[15] and $t_1,\ldots,t_n$ are arbitrary terms

[13] If predicates are considered to be canonically represented by their characteristic functions, a Tarski structure becomes what is called an algebra. Viewed this way an ASM state is a set of functions or Parnas tables [51,10].

[14] Recently this parameterization facility for ASM states has been exploited to define a general ambient concept in terms of ASMs [17].

[15] 0-ary functions $f$, i.e. where $n = 0$, are the variables of programming.

(expressions) at whose values in the current *state* the new value of the function (which will be the value of the successor state of the current *state*) is set to the value of $t$ in the current *state* (if the indicated condition under which this action is requested to be performed is true in the current state). Given the abstract nature of the functions and objects (elements of the universe) which constitute an ASM state one can express updates at any level of abstraction, using corresponding functions $f$ and expressions $t_i, t$ of given complexity or level of abstraction.

This lifts variable assignment to *destructive assignment at any level of abstraction* and thus supports abstract operational modelling (providing what is nowadays often called execution semantics of a system). A typical use is illustrated by the abstract yet precise definition of the two communication actions $ComAct \in \{Send, Receive\}$ of S-BPM agents by the interpreter submachines ASYNC(*ComAct*) and SYNC(*ComAct*) in [33, Appendix,3.3.,3.4].

**Expressivity Question.** Due to its original epistemological goal the definition of ASMs had to solve an expressivity issue for the proposed simple algorithmic language, namely to guarantee that this language provides whatever may be needed to 'directly' (coding-free and thus without extraneous overhead) model any computational system. This is what the ASM thesis [39,40] was about and explains why a) the states of ASMs have to be Tarski structures and why b) differently from their static nature in mathematics and logic here these structures must be treated as updatable by basic actions of ASM agents, namely by (a set of simultaneous)[16] updates.

By its focus on modelling BPs by sets of SBDs each of which is described by constructs that are close to sentences of natural language, S-BPM derives the guarantee to be expressive enough for modelling any desired BP from the expressivity of natural language. The price paid is the focus of ground models on the level of abstraction of (sets of) SBDs which are reached by system decomposition (using data flow diagram techniques) until every communicating subject has become explicit,[17] as will become clearer in the next section where we compare the programming-oriented S-BPM refinement concept explained in Sect. 2.2 with the more general ASM refinement notion.

A positive return is the ease with which an S-BPM model can be transformed into a precise (though verbose) natural language text, essentially by

[16] The synchronous parallelism of single-agent actions in the ASM-computation model, which differs from the sequential-program view of actions of S-BPM agents, provides 'a rather useful instrument for high-level design to *locally describe a global state change*, namely as obtained in one step through executing a set of updates' and 'a convenient way to *abstract from sequentiality* where it is irrelevant for an intended design' [9, p.30].

[17] This interesting termination criterion for the 'decomposition of a system into processes'—the first of the two major system development steps in the S-BPM method—is a consequence of the communication focus (read: subject orientation): 'Finally all processes and shared objects, the messages exchanged between processes and the shared objects they use, are identified.' [31, p.204 and Ch.10]

paraphrasing each SbpmAction in every SBD of the model by the obvious corresponding natural language sentence. Given the similarity between ASM rules and SbpmActions, in a similar way such a transformation can also be defined for ASM models, as has been illustrated in [21]. There the contributing authors of the book [35] had been asked to formulate in natural language a precise and complete set of requirements for a small case study by first defining a formal specification which captures the given informal requirements and then retranslating this specification into natural language. For S-BPM a converter has been written which transforms S-BPM models into natural language texts [32] (see also [58]). Although we believe that the methodological better way to explain and document ASMs (and also S-SBM models) is to use a *literate modeling* style in the spirit of Knuth's literate programming [47], it could nevertheless be useful to write a similar *Asm2NatLang* converter to facilitate the integration of ASMs into natural language S-BPM documents for users who are not familiar with symbolic mathematical notations.

### 3.2 Refinement Concept

The conceptual distance between an SBD (PASS graph) to its refinement, which represents an operational specification of the communication and internal actions the subject performs in the SBD, is not very large. The next step (which we consider as another refinement step) consists in the coding of this specification where the S-BPM method adopts 'methods which are common in standard sequential programming' [31, p.296]. Therefore alltogether the 'semantic gap' between a user model (ground model PASS graph) for a BP and its code is judged not to be very large. In fact it is claimed that 'Once the interaction patterns among actors (subjects) have been refined in terms of exchange of messages, suitable program code can be generated automatically' [34, Sect.1, p.2]; this has to be understood *cum grano salis*, probably meant to hold for 'the standard part of the code' [31, p.295] resp. for code meaning method headers.

This does not solve the problem in case the distance between a ground model and the code is too large to be bridged in one or two steps in such a way that a human can understand the refinement and verify its correctness. Such a situation was at the origin of the ASM refinement method [14] and is typical for its successful applications. Mentioning a few examples should suffice here to illustrate the practical relevance of the ASM refinement notion.

The historically first example is the Prolog-to-WAM compiler verification mentioned in Sect. 2.2 where we needed 12 refinement steps to explain Warren's ideas and to prove the main theorem. The refinement correctness proofs have later been machine verified using the KIV system [55,56]. Interestingly enough to enable the KIV machine to finish its proof, for one of the optimizations in the WAM an additional refinement step had to be introduced into the handwritten proof developed to convince ourselves and our peers. The elaboration of the method for the Occam/Transputer parallel computation model (with nondeterminism) yielded 17 natural refinement steps [19] to explain the rationale and prove the correctness of the standard (INMOS) compilation scheme.

Another real-life example to be mentioned (among many others concerning architectures, control software, protocols, algorithms, etc. and surveyed in [27, Ch.9]) is the stepwise refinement of ASM interpreters for Java and the JVM, using both horizontal and vertical refinement steps. These models have been used to verify various properties of interest for the language and its virtual machine, like type safety, compiler correctness, soundness and correctness of the bytecode verifier, soundness of thread synchronization, etc. The reader can find the details in the JBook [59]. That the method could be applied also to C# [20] and .NET CLR [36,38,37] should not come as a surprise.

A natural place to integrate into S-BPM the ASM refinement method is where one has to code complex internal actions of a subject. It is still a challenge to provide tool support for the ASM refinement method, in particular in combination with verifications of refinement correctness, e.g. building upon the implementation of the ASM refinement concept in [52] which has later been extended and been used for numerous other verification projects, see www.informatik.uni-augsburg.de/swt/projects/. Some first steps in this direction seem to appear in the area of software product lines where feature-based modeling is linked to the stepwise validation and verification of properties [61,46,4,28].

### 3.3 Verification Concern

The presentation of the ASM method quoted at the beginning of Sect. 2 continues as follows:

> 'It covers within a single conceptual framework both *design and analysis*, for procedural single-agent and for asynchronous multiple-agent distributed systems. The means of analysis comprise as methods to support and justify the reliability of software both *verification*, by reasoning techniques, and experimental *validation*, through simulation and testing.' [27, page 1]

This shows how much the ASM method cares about both, verification by proving model properties and validation by simulation and testing of models. However it turned out to be an advantage for their use in systems engineering to pragmatically separate these two activites from the modeling (design) activity [9, Sect.4,5], differently from what do other methods (notably the conceptually very close B-method [2,3]) which link design and verification (definition and proof) to always go together.

The ASM method allows one to validate and/or verify properties of models at any level of abstraction since by their definition

- ASMs are mathematical objects so that they satisfy the rigour needed to enter a mathematical or machine supported proof,
- ASMs are conceptually executable, due to their operational character, and have been made mechanically executable by various tools.[18]

[18] See [27, Ch.8] for a survey of various ASM verification and validation tools and [30] for the more recent CoreASM execution engine.

Verification cannot replace validation, but as early design-error detection technique it can considerably reduce the amount of testing and error correction after the system is built.

The SAPP/PASS approach shares the validation and verification concern. For 'checking whether a process is correct' two aspects are distinguished [31, p.312, Sect.16.3]:

- A system must have certain properties, e.g. livelock free, deadlock free which are independent of the application. This is implicit correctness.[19]
- A specified system must do what a designer has intended. This is explicit correctness.

Both aspects are reported to have been supported by prototypical Prolog-based validation tools providing for each system modeled in PASS a sort of expert system which 'allow(s) the behavior of a process system to be analysed and can determine whether a system does what it was intended to do' (ibid., p.321).

However this verification concern seems not to be supported by the present S-BPM tool set, although the validation concern is, namely by a testing mechanism that allows one to feed concrete values for messages and function arguments and values into the system to run BP scenarios prior to coding method bodies[20].

We suggest to integrate into the current S-BPM system the possibility to

- formulate application-specific BP properties of interest to the user or manager, presumably ground model properties which go beyond the usual graph-theoretic properties like liveness, fairness, deadlock fredom, etc.,
- prove such properties for the ground model as well as their preservation through ASM refinement steps of internal actions,
- document the properties and their verifications so that they can be checked (also by third parties like certification bodies) and used to certify the correctness of the BP implementation.

This could be realized for any of the reasoning techniques the ASM method allows one to apply for the mathematical verification of system properties, at different levels of precision and under various assumptions, e.g. [15, Sect.1]

- outline of a proof idea or *proof sketch* whereby the designers communicate and document their design idea,
- *mathematical proof* in the traditional meaning of the term whereby a design idea can be justified as correct and its rationale be explained in detail,
- *formalized proof* within a particular logic calculus,
- *computer-checked* (automated or interactive) *proof*.

[19] We have pointed out in [16, 4.2] that for BPs 'implicit correctness' properties are less interesting than the ones for 'explicit correctness' which typically are ground model properties to be preserved through refinement steps.

[20] This is exactly the method used in the Falko project at Siemens to validate the ASM ground model for the given scenarios, see [22].

Each technique comes with a different amount of tool support[21] and of effort and cost to be paid for the verification and provides a different level of objective, content-based 'certification' of the professional quality of the analysed system.

# 4 Evaluation of S-BPM

In this section we evaluate S-BPM as an approach to BPM (Sect. 4.2) using six classical evaluation criteria for practical software engineering methods (Sect. 4.1).

## 4.1 The Evaluation Criteria

The three major purposes of business process (BP) descriptions are the *design and analysis*, the *implementation* and the *use* of models of BPs. For each purpose pursued by the various BP stakeholders the models play a specific role, namely to serve a) as conceptual models (in particular for high-level development-for-change and management support), b) as specification of software requirements that are implemented by executable models and c) as user model for process execution, monitoring and management. This is reflected in the following six criteria (paraphrased from [16, Sect.5]) a satisfactory BPM system must satisfy:

**Ground Model Support.** Provide *support for a correct development and understanding* by humans of models and their relation to the application view of the to-be-modeled BP, which is informally described by the process requirements. This human-centered property is often neglected although it is the most critical one for software development systems in general[22] and in particular for BPM systems. It is crucial to support such an understanding *for both model design and use* because these models serve for the communication between
  - the BP expert, who has to explain the real-world BP that is to be implemented,
  - the IT expert who needs a precise specification of the coding goal,
  - the BP user who applies or manages the implemented process and needs to understand for his interaction with the system that his process view corresponds to what the code does.

**Refinement Support.** Provide *support for faithful implementations* of models via systematic, controlled (experimentally validatable and/or mathematically verifiable) refinements. This model-centered property is methodologically speaking the simpler one to achieve because an enormous wealth of established refinement, transformation and compilation methods can be used for this—if the construction of satisfactory (precise, correct, complete and minimal) ground models is supported the implementation can start from.

[21] [27, 9.4.3] surveys some tool supported ASM verifications.

[22] See the discussion in [15] for the verified software challenge [43] originally proposed by Hoare.

**Change Management.** Provide *support for effective change management* of models. This involves the interaction between machines and humans who have to understand and evaluate machine executions for BP (ground or refined) models, bringing in again conceptual (ground model and refinement) concerns when it comes to adapt the system to evolutionary changes.

**Abstraction.** Provide *support for abstraction* to help the practitioner in two respects:

- in the daily challenge to develop abstract models (ground models and their stepwise refinements) out of concrete, real-life problem situations. This implies, in particular, the availability in the modeling language of a rich enough set of abstract data types (sets of objects with operations defined on them) to use so that one can
  - express the application-domain phenomena to be modeled (objects and actions) at the conceptual level without the detour of language-dependent encodings;
  - refine the abstractions in a controlled manner by more detailed operations on more specific data structures.
- to develop coherent definitions of different system views (control-flow view, data flow view, communication view, view of the actors, etc.).

**Modularization.** Provide *support for modularization* through rigorous abstract behavioral interfaces to support structured system compositions into easy-to-change components.[23] For BPM it is particularly important that *modeling-for-change* is supported at all three major stakeholders levels: at the *Ground Model* and *Change Management* support levels because it is the BP users and managers who drive the evolutionary adaptation of BP models, at the *Refinement* support level because the high-level model changes have to be propagated (read: compiled) faithfully to the implementing code.

**Practical Foundation.** Come with a *precise foundation a practitioner can work with*, i.e. understand and rely upon when questions come up about the behavioral meaning of constructs used by the tool.

### 4.2 Applying the Criteria to S-BPM

In this section we recapitulate what has been said showing that S-BPM [33] and its tool [48] support correct development and understanding, faithful implementation and effective management of BP models via practical abstraction and modularization mechanisms which are defined on the basis of a fundamental epistemological and mathematically stable foundation.

S-BPM satisfies the *Ground Model* criterion, as shown in Sect. 2.1.

In Sect. 2.2 we have explained to which extent S-BPM satisfies the *Refinement* criterion and in Sect. 3.2 how it can be enhanced to satisfy the full *Refinement* criterion. Modulo the same remark S-BPM satisfies the *Abstraction* criterion.

[23] These two features, abstraction and modularization, also appear in the *Design* section of *Great Principles Category Narrative* in [29] listed under *simplicity* as one of the five 'driving concerns' of software design and used to 'overcome the apparent complexity of applications'.

The *Change Management* criterion is satisfied by S-BPM via its technique to decompose BPs into sets of SBDs, for which in turn modeling for change is supported by two model extension schemes which allow the modeler to smoothly integrate into a given SBD some new (whether normal or interrupt) behavior [33, Appendix, Sect.6].

To satisfy the *Modularization* criterion S-BPM contributes in various ways. Besides the just mentioned constructs for extending normal or interrupt behavior actions can be atomic or composed. In particular structured alternative actions are available. To accurately model alternative (whether asynchronous or synchronous) communication actions it is sufficient to use an appropriate selection function and the traditional iteration construct to loop through the offered alternatives [33, Appendix, Sect.3.1]. For alternative internal actions a structured split-join mechanism is used which allows the modeler to have the selection simply as non-deterministic choice or to condition the choice by static or dynamic possibly data-related criteria (ibid., Sect.4). Further modular composition constructs include the rigorously defined use of macros, of a normalization to interaction views of SBDs and support for process hierarchies (networks) (ibid., Sect.5).

Notably the model itself which defines the semantics of these features is formulated in a modular way using stepwise ASM refinement (ibid.).

Last but not least S-BPM has a *Practical Foundation* via the accurate definition of its semantics using the language of ASMs—a mathematically precise, wide-spectrum action description language which uses rules of the the same form as guarded basic SBD actions (see Sect. 2.1) and thus is familiar to all BP stakeholders.

## References

1. van der Aalst, W., ter Hofstede, A.: Workflow patterns home page (created and maintained since 1999), `http://www.workflowpatterns.com/`
2. Abrial, J.R.: The B-Book. Cambridge University Press, Cambridge (1996)
3. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press, Cambridge (2010)
4. Batory, D., Börger, E.: Modularizing theorems for software product lines: The Jbook case study. J. Universal Computer Science 14(12), 2059–2082 (2008)
5. Börger, E.: A Logical Operational Semantics for Full Prolog. Part I: Selection Core and Control. In: Börger, E., Kleine Büning, H., Richter, M.M. (eds.) CSL 1989. LNCS, vol. 440, pp. 36–64. Springer, Heidelberg (1990)
6. Börger, E.: A Logical Operational Semantics of Full Prolog. Part II: Built-in Predicates for Database Manipulation. In: Rovan, B. (ed.) MFCS 1990. LNCS, vol. 452, pp. 1–14. Springer, Heidelberg (1990)
7. Börger, E.: A logical operational semantics for full Prolog. Part III: Built-in predicates for files, terms, arithmetic and input-output. In: Moschovakis, Y.N. (ed.) Logic From Computer Science, Berkeley Mathematical Sciences Research Institute Publications, pp. 17–50. Springer, Heidelberg (1992)
8. Börger, E.: Logic programming: The Evolving Algebra approach. In: Pehrson, B., Simon, I. (eds.) IFIP 13th World Computer Congress. Technology/Foundations, vol. I, pp. 391–395. Elsevier, Amsterdam (1994)

9. Börger, E.: Why Use Evolving Algebras for Hardware and Software Engineering? In: Bartosek, M., Staudek, J., Wiedermann, J. (eds.) SOFSEM 1995. LNCS, vol. 1012, pp. 236–271. Springer, Heidelberg (1995)
10. Börger, E.: Evolving Algebras and Parnas tables. In: Ehrig, H., von Henke, F., Meseguer, J., Wirsing, M. (eds.) Specification and Semantics. Dagstuhl Seminar No. 9626, Schloss Dagstuhl. Int. Conf. and Research Center for Computer Science (July 1996)
11. Börger, E.: High-Level System Design and Analysis Using Abstract State Machines. In: Hutter, D., Stephan, W., Traverso, P., Ullmann, M. (eds.) FM-Trends 1998. LNCS, vol. 1641, pp. 1–43. Springer, Heidelberg (1999)
12. Börger, E.: The origins and the development of the ASM method for high-level system design and analysis. J. Universal Computer Science 8(1), 2–74 (2002)
13. Börger, E.: The ASM Ground Model Method as a Foundation of Requirements Engineering. In: Dershowitz, N. (ed.) Verification: Theory and Practice. LNCS, vol. 2772, pp. 145–160. Springer, Heidelberg (2004)
14. Börger, E.: The ASM refinement method. Formal Aspects of Computing 15, 237–257 (2003)
15. Börger, E.: Construction and analysis of ground models and their refinements as a foundation for validating computer based systems. Formal Aspects of Computing 19, 225–241 (2007)
16. Börger, E.: Approaches to modeling business processes. A critical analysis of BPMN, workflow patterns and YAWL. J. Software and Systems Modeling (2011), doi:10.1007/s10270-011-0214-z
17. Börger, E., Cisternino, A., Gervasi, V.: Ambient Abstract State Machines with applications. J. Computer and System Sciences (2011); special Issue in honor of Amir Pnueli, http://dx.doi.org/10.1016/j.jcss.2011.08.004
18. Börger, E., Dässler, K.: Prolog: DIN papers for discussion. ISO/IEC JTCI SC22 WG17 Prolog Standardization Document 58. National Physical Laboratory, Middlesex, England (1990)
19. Börger, E., Durdanović, I.: Correctness of compiling Occam to Transputer code. Computer Journal 39(1), 52–92 (1996)
20. Börger, E., Fruja, G., Gervasi, V., Stärk, R.: A high-level modular definition of the semantics of C#. Theoretical Computer Science 336(2-3), 235–284 (2005)
21. Börger, E., Gargantini, A., Riccobene, E.: Abstract State Machines. A method for system specification and analysis. In: Frappier, M., Habrias, H. (eds.) Software Specification Methods: An Overview Using a Case Study, pp. 103–119. HERMES Sc. Publ. (2006)
22. Börger, E., Päppinghaus, P., Schmid, J.: Report on a Practical Application of ASMs in Software Design. In: Gurevich, Y., Kutter, P.W., Odersky, M., Thiele, L. (eds.) ASM 2000. LNCS, vol. 1912, pp. 361–366. Springer, Heidelberg (2000)
23. Börger, E., Prinz, A.: Quo Vadis Abstract State Machines? J. Universal Computer Science 14(12), 1921–1928 (2008)
24. Börger, E., Rosenzweig, D.: From Prolog Algebras Towards WAM – a Mathematical Study of Implementation. In: Schönfeld, W., Börger, E., Kleine Büning, H., Richter, M.M. (eds.) CSL 1990. LNCS, vol. 533, pp. 31–66. Springer, Heidelberg (1991)
25. Börger, E., Rosenzweig, D.: WAM Algebras – a Mathematical Study of Implementation, Part 2. In: Voronkov, A. (ed.) RCLP 1990 and RCLP 1991. LNCS, vol. 592, pp. 35–54. Springer, Heidelberg (1992)
26. Börger, E., Rosenzweig, D.: The WAM – definition and compiler correctness. In: Beierle, C., Plümer, L. (eds.) Logic Programming: Formal Methods and Practical

Applications, Studies in Computer Science and Artificial Intelligence, ch. 2, vol. 11, pp. 20–90. North-Holland (1995)

27. Börger, E., Stärk, R.F.: Abstract State Machines. A Method for High-Level System Design and Analysis. Springer, Heidelberg (2003)
28. Delaware, B., Cook, W., Batory, D.: Product lines of theorems. In: Proc. OOPSLA 2011, Portland (October 2011)
29. Denning, P.J., Martell, C.: Great principles of computing (2007), http://cs.gmu.edu/cne/pjd/GP/GP-site/welcome.html (consulted July 26, 2011)
30. Farahbod, R., et al.: The CoreASM Project, http://www.coreasm.org
31. Fleischmann, A.: Distributed Systems: Software Design and Implementation. Springer, Heidelberg (1994)
32. Fleischmann, A.: Sbpm2NatLang converter. e-mail of September 8 to Egon Börger (2011)
33. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement. Hanser-Verlag, München (2011); see [63] for a correct version of the appendix
34. Fleischmann, A., Stary, C.: Whom to talk to? A stakeholder perspective on business process development. Universal Access in the Information Society, 1-26 (June 2011), doi: 10.1007/s10209-011-0236-x
35. Frappier, M., Habrias, H. (eds.): Software Specification Methods: An Overview Using a Case Study. HERMES Sc. Publ., Paris (2006)
36. Fruja, N.G.: Type Safety of C# and .NET CLR. Ph.D. thesis, ETH Zürich (2006)
37. Fruja, N.G.: Towards proving type safety of .net cil. Science of Computer Programming 72(3), 176–219 (2008)
38. Fruja, N.G., Börger, E.: Modeling the .NET CLR Exception Handling Mechanism for a Mathematical Analysis. Journal of Object Technology 5(3), 5–34 (2006), http://www.jot.fm/issues/issue_2006_04/article1
39. Gurevich, Y.: Reconsidering Turing's Thesis: Toward more realistic semantics of programs. Technical Report CRL-TR-36-84, EECS Department, University of Michigan (September 1984)
40. Gurevich, Y.: A new thesis. Abstracts, American Mathematical Society 6(4), 317 (1985)
41. Gurevich, Y.: Evolving algebras 1993: Lipari Guide. In: Börger, E. (ed.) Specification and Validation Methods, pp. 9–36. Oxford University Press (1995)
42. Gurevich, Y.: Interactive algorithms 2005 with added appendix. In: Goldin, D., Smolka, S.A., Wegner, P. (eds.) Interactive Computation: The New Paradigm, pp. 165–182. Springer, Heidelberg (2006)
43. Hoare, C.A.R., Misra, J., Leavens, G.T., Shankar, N.: The verified software initiative: a manifesto. ACM Computing Surveys (2009)
44. Hoare, C.: Communicating Sequential Processes. Prentice Hall (1985)
45. ter Hofstede, A., van der Aalst, W., Adams, M., Russell, N. (eds.): Modern Business Process Automation. Springer, Heidelberg (2010)
46. Kim, C.H.P., Batory, D., Khurshid, S.: Reducing combinatorics in testing product lines. In: Proc. Aspect Oriented Software Development Conference. ACM (2011)
47. Knuth, D.: Literate Programming, CSLI Lecture Notes, vol. 27. Center for the Study of Language and Information, Stanford (1992)
48. Metasonic: Metasonic-suite, http://www.metasonic.de/metasonic-suite
49. Milner, R.: A Calculus of Communication Systems. LNCS, vol. 92. Springer, Heidelberg (1980)

50. OMG: Business Process Model and Notation (BPMN) (2011), http://www.omg.org/spec/BPMN/2.0 (formal, January 03, 2011)
51. Parnas, D.L., Madey, J.: Functional documents for computer systems. Sci. of Comput. Program. 25, 41–62 (1995)
52. Schellhorn, G.: Verifikation abstrakter Zustandsmaschinen. Ph.D. thesis, Universität Ulm, Germany (1999)
53. Schellhorn, G.: Verification of ASM refinements using generalized forward simulation. J. Universal Computer Science 7(11), 952–979 (2001)
54. Schellhorn, G.: ASM refinement and generalizations of forward simulation in data refinement: A comparison. Theoretical Computer Science 336(2-3), 403–436 (2005)
55. Schellhorn, G., Ahrendt, W.: Reasoning about Abstract State Machines: The WAM case study. J. Universal Computer Science 3(4), 377–413 (1997)
56. Schellhorn, G., Ahrendt, W.: The WAM case study: Verifying compiler correctness for Prolog with KIV. In: Bibel, W., Schmitt, P. (eds.) Automated Deduction – A Basis for Applications. Applications, vol. III, pp. 165–194. Kluwer Academic Publishers (1998)
57. Semiconductor Industry Assoc.: International technologoy roadmap for semiconductors. Design (2005), http://www.itrs.net/Links/2005ITRS/Design2005.pdf
58. Sneed, S.H.: Exporting Natural Language: Generating NL Sentences Out of S-BPM Process Models. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 163–179. Springer, Heidelberg (2011)
59. Stärk, R.F., Schmid, J., Börger, E.: Java and the Java Virtual Machine: Definition, Verification, Validation. Springer, Heidelberg (2001)
60. Tarski, A.: Der Wahrheitsbegriff in den formalisierten Sprachen. Studia Philosophica 1, 261–405 (1936)
61. Uzuncaova, E., Khurshid, S., Batory, D.: Incremental test generation for software product lines. IEEE Transactions on Software Engineering 36(3), 309–322 (2011)
62. Wirth, N.: Algorithms & Data Structures. Prentice-Hall (1975)
63. Here the file for the correct text of the appendix of [63] can be, downloaded, http://www.hanser.de/buch.asp?isbn=978-3-446-42707-5&area=Wirtschaft, http://www.di.unipi.it/~boerger/Papers/Bpmn/SbpmBookAppendix.pdf

# Ad-Hoc Adaption of Subject-Oriented Business Processes at Runtime to Support Organizational Learning

Thomas Rothschädl*

Metasonic AG Münchner Straße 29, Hettenshausen, 85276 Pfaffenhofen, Germany
thomas.rothschaedl@metasonic.de

**Abstract.** Current Business Process Management Solutions focus on the definition of an exact description of a business process to be executed as a workflow. But business reality does not fit into deterministic designed models and sometimes it requires unpredicted tasks, differing from the predefined process model, to achieve an optimal result. Subject-oriented business process management (S-BPM) enables people, who are directly involved in a workflow, to represent their processes from an individual view and define the interaction with others through message exchange. Such a subject oriented model can be interpreted as workflow and executed on a workflow engine. This paper introduces an approach how S-BPM based workflow instances can be modified during runtime to give individual workers the possibility to deal with unpredicted events and shows how such a solution supports organizational learning.

**Keywords:** workflow flexibility, ad-hoc workflow adaption, knowledge life cycle, subject-oriented business process management (S-BPM).

## 1 Introduction

Business Process Management (BPM) promises to be the Swiss army knife for dealing with information systems, organizational learning and controlling of a company's business processes. BPM should help to keep the company flexible and agile, it should allow controlling the processes of a company, improve them or completely redesign them in an easy way and helps to become an evolving, learning company. With these promises, and sometimes also a small impulse by the legislative, companies introduce BPM suites and become often disillusioned because it is still a hard way from a promising idea to a running business processes.

The classical Business Process Management (BPM) live-cycle from analyses over modeling to execution and thereupon monitoring and optimisation fits perfectly to organizational learning theory but in business reality this framework emerges rather ponderous. Business processes are collected and designed by BPM specialists. After the business process is modeled it has to, or at least it should,

* Thanks to Nils Meyer and Markus Radmayr for their great input.

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 22–32, 2012.

be discussed with the individuals who are involved and/or are responsible for the execution of the process [7,8,10].

Usually the process model has to be translated to a workflow model to be executable on a workflow engine in the next step. The workflow engine itself has to be integrated within the companies' information systems by technicians. Workflows are monitored by business analysts who initiate a redesign of the business process, if necessary [8].

Subject-oriented BPM (S-BPM) is an approach which focuses on the acting elements in a business process where the subjects are abstract resources which execute defined actions on objects. Subjects synchronize their activities by the exchange of messages [6, p. 90]. In S-BPM the transformation of a business process to a workflow is no longer necessary, a Subject-oriented process model is interpretable as a workflow [7]. Due to this fact there is no need to make a difference between process model and workflow model, hence in S-BPM only processes exist. Although this modeling approach enables users to describe, build and adapt process models from their individual perspective, current S-BPM solutions lack, like other common BPM tools, in supporting case related adaption of predefined process models. Current solutions do not offer people responsible for execution, possibilities to deal with unpredicted situations or to try alternative solutions within the workflow engine in an ad-hoc manner. Hence they are not suitable for knowledge intensive work [18] or to support affected employees to find and try alternative and perhaps better solutions for existing problems within the workflow execution environment.

This paper describes an S-BPM based process execution environment which also enables process users to find their own solutions for unpredicted problems.

The next chapter draws the process from basic problem solving over organizational learning with the knowledge live cycle to BPM and describes how current BPM suites already support organizational learning and why the enabling of ad-hoc processes in process execution environments is important for organizational learning. Chapter 3 illustrates that ad-hoc processes also need constraints and describes how these constraints could be realised in S-BPM. The final chapter shows which further work is necessary.

## 2 The Knowledge Live Cycle – From Problem Solving to BPM

Business Process Management Systems (BPMS) use predefined process models to guide users through more or less complex business processes and provide them with the necessary data. In an optimal situation these models are created in cooperation with all process stakeholders. When created, a model is declared as complete and can be published in a BPMS. After the model is published users can start a new instance of the process and start with the execution of the defined work. But there is never work that is 100 per cent predictable, nor is there work that is 100 per cent unpredictable and in most cases users of a BPMS have no tools that allow modification of the process instance at runtime

[18]. If a unpredicted problem within a process instance arises, the user involved has to find a solution by himself. In the best way, the involved user initiates the adaptation of the process model to enable further instances of the process to deal with the problem which has emerged in in a adequate way, but in BPM there is no concept of adapting process instances to a particular problem at the moment.

The approach shown in this paper presents an improvement of S-BPM based BPMS which is based on the theory of Firestone and McElroy presented in [4]. Remarkable and novel of Firestone and McElroys approach is the combination of Argyris/Schön organizational learning theory [1] and Karl Poppers problem solving framework [15] which offers the possibility to view organizational learning as a holistic approach from simple problem solving on the level of an individual up to redefining the governing values of an organization. This chapter shows that the Subject-oriented business process modeling approach fits rather well to this framework.

### 2.1 Decision Execution Cycle

When Firestone and McElroy look at knowledge management in [5], they start at an individual level, on the level of the subjects (they call them agents) and their decisions. According to [5] decisions are produced by planning and are embodied in acting. Decisions lead to actions and actions are the basis of organizational systems. Figure 1 illustrates the phases of this sequence which they call the Decision Execution Cycle (DEC). This cycle uses previously existing knowledge to decide and act. Within this model there is also new knowledge, by using knowledge which has already been generated. But this happens only in a routine way. Until now we are in the Single Loop Learning Process according to [1] which is illustrated in Figure 3. If there was a learning process, it only changes the actions of a subject due to a mismatch between what the acting person wanted and the result of the action. Or, to use the terms of S-BPM, a subject who runs an instance of a process model is moving within the predefined process borders.

But what happens if a situation occurs that is not mapped in the process model? Or to stay at Firestone [5, p. 193], what if the mismatch is great enough from the viewpoint of the individual, and when the individual decides that previous knowledge will not work to reduce this mismatch? Generally there are two possibilities, throw away the tools and run or try hard to solve the problem although there is a gap between what he or she needs to know in order to pursue the goals. This situation is illustrated in Figure 2 where the gap between the desired result and the achieved is too big to be closed with the existing knowledge and new problem solving strategies are needed which can produce new knowledge if they are successful. At this point the subject enters the double loop learning process where new solutions should be found to generate new knowledge. Following Firestone [4] and Popper [15], double loop learning is an emergent, non-deterministic process which starts with the formulation of the problem to the development of different solutions and eliminating the false ones.

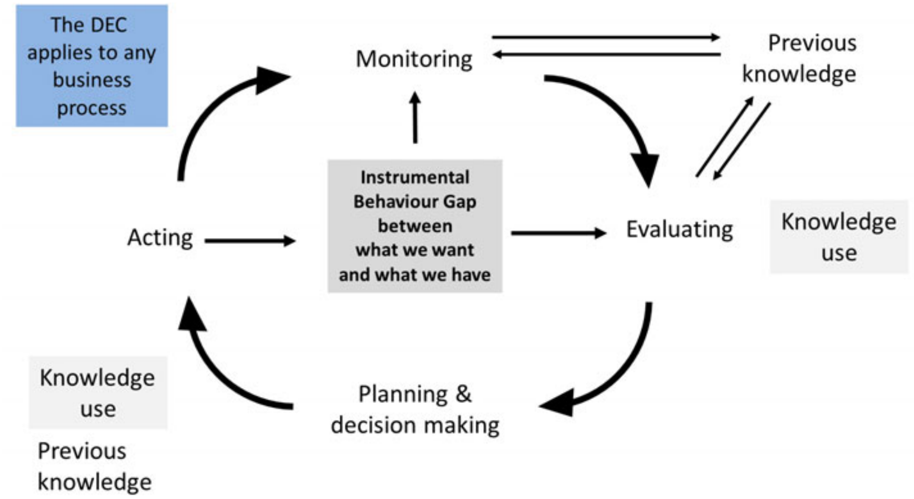



**Fig. 1.** Decision-Execution Cycle according to [3]

The result of the error eliminations can be new knowledge which should be integrated into the memory of an organization so that it will be available for re-use later. According to Popper new knowledge also suggests new problems ($P_2$ in Figure 4) and therefore life can be seen as a continuous problem solving cycle.

### 2.2 How Business Process Management Already Supports the Knowledge Life Cycle

BPMS can support organizational learning in a basic way in both Single Loop Learning (SLL) and Double Loop Learning (DLL). Process instances could be seen as a framework for Single Loop Learning wherein subjects are acting. From the users' perspective a subject is guided through a business process by the BPMS. Within a process instance decisions can be made according to the model. If the process is modeled in a sufficient manner, a subject should be able to plan and act in a satisfactory way within the single loop, based on the governing values from the process model, in combination with the internal governing values of the subject. But the process model usually gives the frame for the performable decisions and actions.

A BPMS support users to run a process instance within the predefined process borders. Users execute "Actions" followed by "Events and Conditions". On a basis of the outcome of this events users can plan to perform actions again. In S-BPM, actions would be expressed through behaviour states of subjects, thus "Send Message", "Receive Message" and "Action/Activities" [6]. Knowledge within a BPMS usually means knowledge about the flow of processes. The flow of repeatable business processes with already existing knowledge can be

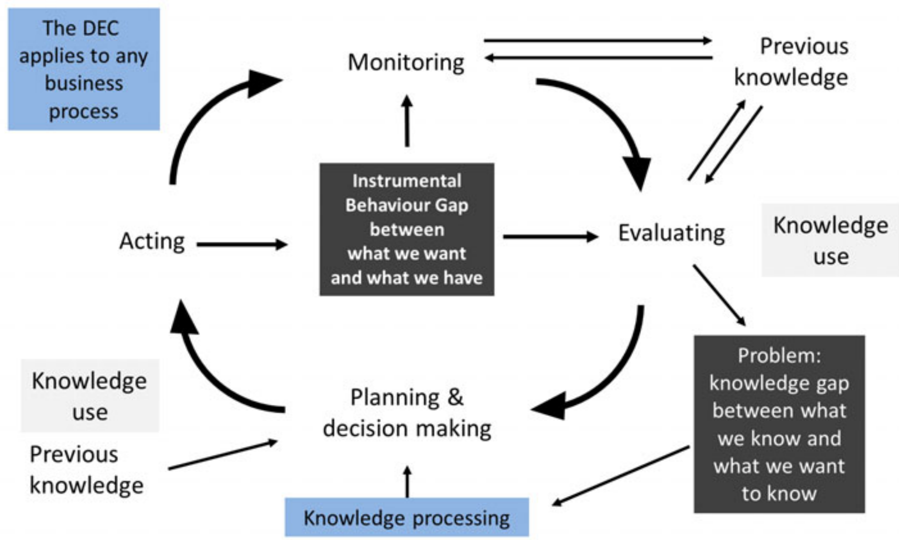



**Fig. 2.** Adding problems to the Decision-Execution Cycle according to [3]

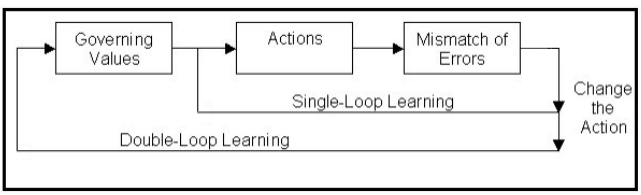



**Fig. 3.** Single Loop Learning and Double Loop Learning according to [1]

supported well by a BPMS. A BPMS can assist users going through business processes and supports them with knowledge to make the right decisions at different process steps.

Business Process Re-engineering can be seen as DLL. Process models could be seen as a tight frame which support actors running their personal Decision Execution Cycle. But as we are not acting in a deterministic world, it could and will happen that there are situations which are not covered by the process model. Hopefully the participants involved will find a solution for the problem by themselves. But they have to leave the BPMS and find a fitting solution by themselves, possibly with support from other information systems. If the problem is successfully solved, the solution can be integrated into the BPMS through Business Process Reengineering. Because there is only one model in S-BPM, this can be done in a fast way, but currently a process has to be modeled with external tools before it could be published on a BPMS.

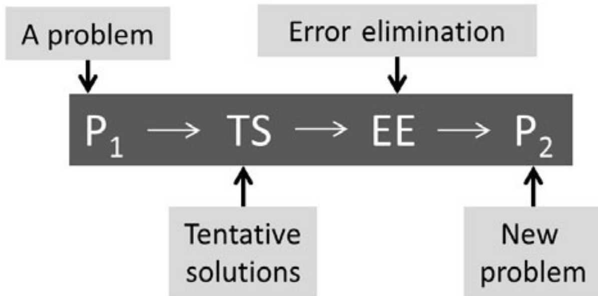



**Fig. 4.** Popper's tetradic schema according to [4]

### 2.3 Lacks of Current BPMS in Dealing with the Knowledge Life Cycle

The last section describes how BPMS already support the Knowledge Life Cycle and also mentioned shortly some lacks of dealing with unpredicted situation. Actors within a running process instance have to recognize that they will not be able to achieve their goal (hence the process goal) with the given tools and have to find a solution for it outside the BPMS. An idea which has been published several times since the late 90s [19,2,16,20,17], would be to give actors the possibility of initializing or adapting process instances in an ad-hoc manner. Although there have been several approaches to finding a compromise between static business processes on the one hand and Computer Supported Cooperative Work (CSCW) there is no solution how ad-hoc processes could be realized. On the one hand, if the system is too restrictive, there is no space to find own problem solving strategies, on the other hand there also exists critical processes which have to be done exactly in a certain way, for instance due to controlling or compliance reasons. The next chapter explains how ad-hoc deviation from the process model, or parts of it, could be granted during design time in an S-BPM environment.

## 3 Different Stages of Ad-Hoc Processes

In S-BPM the communication structure between subjects and the behaviour of subjects can be seen as two different levels where ad-hoc process adaption can occur. The communication structure defines which subjects are involved within a process and which messages they exchange whereas the behaviour structure, the sequence of send or receive messages and actions are executed through the subject [6]. This chapter describes different views to ad-hoc process adaption in an S-BPM environment.

### 3.1 Ad-Hoc Processes through Construction or Restriction

Generally, it can be distinguished between everything is allowed, like in a classical CSCW and a restricted system, which only allows solutions that have been considered during design time. The everything-is-allowed approach lacks in the support of executing standard operation procedures in an efficient way, hence some rules and regulations in which way processes should be executed, are useful and necessary. The preferred system would offer the "ability of the process to execute on the basis of a loosely or partially specified model, where the full specification of the model is made at runtime, and may be unique to each instance" [16, p.516]. As S-BPM offers the possibility to design processes not only through construction, but also through restriction [7, p. 160] it enables a complete new way to design less structured processes in a similar, but different way as [16] describes.

Modeling through construction means the classical way, where a process model is generated from scratch as known from Business Process Modeling Notation (BPMN) or from Event-driven process chain (EPC). Modeling through restriction starts from a universal process model, where each subject can send and receive messages with a generic content to and from any subject at any time. This generic model could be used as the starting point of a process definition which has to become more restricted [7, p. 160].

For an illustration take an application for leave process which has to be accepted by the line manager of a employee. You can start to define the process by modeling through restriction with the two subjects "Employee" and "Line Manager" where the employee sends an application for leave to the line manager and the line manager sends an acceptance or a rejection back to the employee. This is the static part of the process because it is the standard operation procedure for an application for leave within the company. So within the model all message exchangeing between these two subjects is exactly predefined. A more generic model would also keep the possibility to include co-workers, potentially existing project managers or, in an optimal way, at design time completely unknown subjects into the application for leave process by the applicant to facilitate an adequate decision for the line manager or to give the applicant the possibility to rethink and adapt his application. The core process, to grant or turn down the application for leave from an employee by the line manager is exactly predefined, but the exact path to create this application is left to the involved employee.

The execution of such a weakly structured business processes could be supported through the ModelAsYouGo approach [9].

### 3.2 Expressing Expected Behaviour to other Subjects

The possibility to model external subjects within a S-BPM based process model, enables also the explicit inclusion or exclusion of communication with subjects outside the BPMS within a universal process model. In case of containing an external subject, a universal process model would also allow ad-hoc communication with external subjects. In this case, but also if a special reaction of an internal subject is expected, the approach of a Behavioural Interface presented by Meyer et al in [12,13], can be used to express which behaviour from the receiver subject is expected. In context with the previous example this could mean, that an employee sends his application for leave to a project leader together with a

Behavioural Interface, where he describes the expected behaviour of the project leader, namely adding comments to the document whether he agrees or disagrees with the application for leave. If the applicant receives a disagree, the applicant can decide for himself to try to insist on his application and forward it to the line manager, adapt or even drop it and do not bother the line manager with it.

### 3.3 Ad-Hoc Processes within or between Subjects

As previously mentioned, there are two layers for modeling business processes in S-BPM. The universal process model covers the ability to allow an adaptation on communication level where at least two or more subjects are included. However, it is also possible that the communication between subjects is restricted, due to company compliance for instance, but a modeler would like to give a single subject the possibility to adapt its behaviour beyond the modeled behaviour. For this case S-BPM offers the possibility to model behaviour extensions [7, p. 151] or exceptions [7, p. 147] within the internal behaviour of subjects. This can be used to grant a process executing agent, within predefined process steps, the possibility to leave the modeled path and to find ad-hoc different solutions for occurred problems.

## 4 Realizing Subject Oriented Ad-Hoc Processes and Organizational Learning

As mentioned before, S-BPM already offers the possibility to enable ad-hoc process adaptation in design time, but to benefit from this, it also has to be possible to adapt running instances of such an open process model. However, for supporting individual problem solving there need to be options for the end user, who is confronted with the unmatched problem, to deal with them in an adequate way. The adopted business processes should be recorded and stored as tentative solutions. These tentative solutions have to be analysed and the results should be used to improve existing process models or as basis of new ones. This chapter introduces some approach how this could be realized.

Previously, the need for enabling ad-hoc process execution from a knowledge generation perspective was introduced, though the hard try will not only be to give actors a possibility to find their own solutions, but also to ensure that actors use these possibilities to generate new knowledge. The new knowledge can, after an evaluation process, be reintegrated within an enterprises process landscape. All the proposed solutions have to overcome the classical disjunction between process model and process instance and offer possibilities to change a process during runtime which is also called "just-in-time-modeling" [11]. Additionally, everybody who is involved and authorized to an instance, should be able to adapt it to fit to his or her individual needs. Below there are three different approaches for ad-hoc user integration are presented. Each of them can be used to solve problem situations with ad-hoc processes in a different setting.

### 4.1 Graphical Modeling during Process Execution

One approach for designing ad-hoc processes step-by-step in an collective way was presented by Huth et al. in [11]. They proposed allowing the change of a process instance in a collective way and on the fly with the support of a graphical modeling tool which show the current status and can change the running process instance during runtime. The advantage of this approach is, that it enables the participants to stay focused to the goal of a business process. Furthermore it enables an optimisation of the business process on design time through different participants involved. This fits rather in cases where the involved actors know how to express their needs in the S-BPM modeling language. It also allows several different actors to agree on actions before they have to be executed and it would be easier to keep the goal of the business process in sight. But in reality it is questionable if everybody who is involved to a business process will have, and also really needs, the modeling knowledge, which would be required by this solution.

### 4.2 Distributed Modeling while Process Execution with a Table Top Interface

For supporting a spatial distributed modeling of interactions and internal behaviour the use of a table top device would be possible. The feasibility of this synchronous distributed modeling approach was shown by Oppl in [14]. "People that are involved in a work process on an operational level often are hardly able to abstract form and reflect upon their daily routines. Elicitation of knowledge then requires methods and tools to support these people in externalizing their view of their work processes." [14, p. 17] The solution presented enables different located actors to express and articulate their knowledge. These abilities would be useful to present and discuss tentative solutions which should be achieved by process modification between different actors.

### 4.3 Modeling while doing Process Execution

It would be also possible to utilize the ModelAsYouGo approach presented by Gottanka and Meyer in [9]. With ModelAsYouGo the actors can create a process model by acting within the process execution environment. If an actor recognizes that a process will not lead to the desired result within the given process model, and he or she is enabled to change the running process instance, he or she can use ModelAsYouGo for leaving the predefined process path without need for a previous modeling of his process modification. The actors can create the model while acting and would not realize that they are modeling a business process. No further modeling knowledge is necessary. A disadvantage of this approach is, that the single acting person could lose the overview of the process, hence the process goal or that the process goal is only reached in an inefficient way.

## 5 Conclusion and Future Work

This paper shows that the adaptation of process instances in an ad-hoc manner would be eligible from an organizational learning perspective. This could enable subjects to find new ways for solving unpredicted problems within workflows and also the creation of new knowledge. Furthermore it is shown that the S-BPM paradigm already offers semantic elements which allow designing more flexible models. In particular the possibility to create business process models not only through construction but also through restriction. The last chapter shows that the enabling of ad-hoc processes has also to include executing actors and their different needs. Due to the occurred problem and due to the process modeling knowledge of actors involved, different types of user interaction are necessary. Further work will focus on the creation and evaluation of prototypes for modeling processes which enable ad-hoc modification and also for ad-hoc adaptation of running instances.

## References


1. Argyris, C.: Theory in practice: increasing professional effectiveness, 1st ed., 6. print. edn. Jossey-Bass Publ., San Francisco (1992)
2. Bernstein, A.: How can cooperative work tools support dynamic group process? bridging the specificity frontier. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW 2000, pp. 279–288. ACM, New York (2000), http://doi.acm.org/10.1145/358916.358999
3. Firestone, J.: Mind the (knowledge) gap - inside knowledge, http://www.ikmagazine.com/xq/asp/txtSearch.CoP/exactphrase.1/sid.0/articleid.7A83CD76-CFBF-4A86-A040-034EF4ED1C12/qx/display.htm
4. Firestone, J.M., McElroy, M.W.: Key issues in the new knowledge management. Butterworth-Heinemann (2003)
5. Firestone, J.M., McElroy, M.W.: Doing knowledge management. The Learning Organization 12(2), 189–212 (2005)
6. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010), http://www.springerlink.com/content/m4105554174112q7/
7. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement: Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern. Hanser, München (2011)
8. Gadatsch, A.: Management von Geschäftsprozessen: Methoden und Werkzeuge für die IT-Praxis eine Einführung für Studenten und Praktiker. Vieweg, Braunschweig (2001)
9. Gottanka, R., Meyer, N.: ModelAsYouGo: (Re-)Design of S-BPM Process Models During Execution Time. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 91–105. Springer, Heidelberg (2012)
10. Hammer, M.: What is business process management? In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, vol. 1, pp. 3–16. Springer, Heidelberg (2010), http://han.ubl.jku.at/han/SPR-EBOOK-SCS/www.springerlink.com/content/t0801783q8qml054/

11. Huth, C., Nastansky, L.: GroupProcess: partizipatives, verteiltes design und simultane ausführung von ad hoc geschäftsprozessen. In: GeNeMe, pp. 319–334 (2000)
12. Meyer, N., Feiner, T., Radmayr, M., Blei, D., Fleischmann, A.: Dynamic Catenation and Execution of Cross Organisational Business Processes - The jCPEX! Approach. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 84–105. Springer, Heidelberg (2011), http://www.springerlink.com/content/x2157q0x31602701/
13. Meyer, N., Radmayr, M., Heininger, R., Rothschädl, T., Fleischmann, A.: Platform for Managing and Routing Cross-Organizational Business Processes on a Network Router. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 175–189. Springer, Heidelberg (2011), http://www.springerlink.com/content/m16m83u875664204/
14. Oppl, S.: Subject-Oriented Elicitation of Distributed Business Process Knowledge. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 16–33. Springer, Heidelberg (2011), http://www.springerlink.com/content/t88jp875r76k0456/
15. Popper, K.: Alles Leben ist Problemlösen: über Erkenntnis, Geschichte und Politik, 4. aufl. edn. Piper, München (1995)
16. Sadiq, S., Sadiq, W., Orlowska, M.: Pockets of Flexibility in Workflow Specification. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 513–526. Springer, Heidelberg (2001), http://www.springerlink.com/content/c8gpu4ft32lehfg3/
17. Stoitsev, T., Scheidl, S., Spahn, M.: A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes. In: Winckler, M., Johnson, H., Palanque, P. (eds.) TAMODIA 2007. LNCS, vol. 4849, pp. 213–226. Springer, Heidelberg (2007)
18. Swenson, K.: Mastering the unpredictable: how adaptive case management will revolutionize the way that knowledge workers get things done. Meghan-Kiffer Press, Tampa Fla (2010)
19. Voorhoeve, M., Van der Aalst, W.: Ad-hoc workflow: problems and solutions, pp. 36–40. IEEE Comput. Soc. (1997)
20. Wassink, I., Van Der Vet, P., van Dijk, E., van der Veer, G., Roos, M.: New interactions with workflow systems. In: European Conference on Cognitive Ergonomics: Designing beyond the Product Understanding Activity and User Experience in Ubiquitous Environments, p. 29 (2009)

# An Approach towards Subject-Oriented Access Control

Alexander Lawall, Thomas Schaller, and Dominik Reichelt

Hof University Institute of Information Systems,
Alfons-Goppel-Platz 1, 95028 Hof
{alexander.lawall,thomas.schaller,dominik.reichelt}@iisys.de

**Abstract.** Looking at the various applications and systems needed for running a business one thing becomes obvious: in almost every application there is the need to maintain a model of the organization structure, of the roles and the actors in order to define access rights or assign tasks to employees (in case of a workflow management system). It is out of the question that these redundancies lead to a great maintenance overhead that - even for small business - can grow to an immense burden. This problem can be avoided by deploying one logical organization server offering this service to the other applications. In this paper, we will present a model of an organization server that, in contrast to conventional, centralized approaches, gives more power to the subjects running the processes of a company.

**Keywords:** workflow management, BPM 2.0, access control, empowerment, decentralization.

## 1 Motivation

Subject-oriented business process management [7] has gathered quite a community around its original idea that business processes are defined by how individual actors – the subjects – interact in order to achieve a specific goal. The S-BPM paradigm can be regarded as a bottom-up way of modelling business processes, as these subjects are a rather direct abstraction of humans or systems partaking in a business process. Being the real core of a business process, the subjects themselves play a key role throughout the whole enterprise model that is built iteratively on process analyses. So let us have a look how the subjects including their privileges and duties are represented in today's IT systems.

Actually, a typical IT landscape in an organization consists of a multitude of different technical systems, such as operating systems, databases and applications. The access to such backend systems is granted through their individual security components or alternatively using independent security products. A user who needs to have access to the different systems must therefore be created separately and maintained in all the individual systems, in the most basic way as a concrete entry in the end system's security component. Due to the multitude of these end systems and the frequent changes in requirements raised by the evolving business environment, an integrated and comprehensive solution for enterprise-wide access control is a necessity for companies.

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 33–42, 2012.

## 2 Role-Based Access Control (RBAC)

In role-based access control (RBAC), permissions are not directly associated to users but are instead accumulated in roles [3]. Users are then assigned to these roles, thereby acquiring the role's permissions. A role typically contains all clearances needed in an organizational unit or for a specific job function. As the number of roles is usually assumed to be considerably lower than the number of users and permissions, the number of administrative tasks required for maintaining the permissions can be reduced. The mapping between many users and few roles is a prime use case for RBAC. It allows the administrators to have a better overview of the permissions granted to a user. On the one hand, auditors can check the access rights of individual users more easily, and on the other hand, administrators can authorize users in a more controlled way.

In companies working with individual permission assignments, as opposed to employing RBAC, users often accumulate access rights when taking on different positions within the company. Nobody can really determine which permissions belong to deprecated job functions and which of them are really needed by the employee. Thus, the usage of roles associated with such functions increases security.

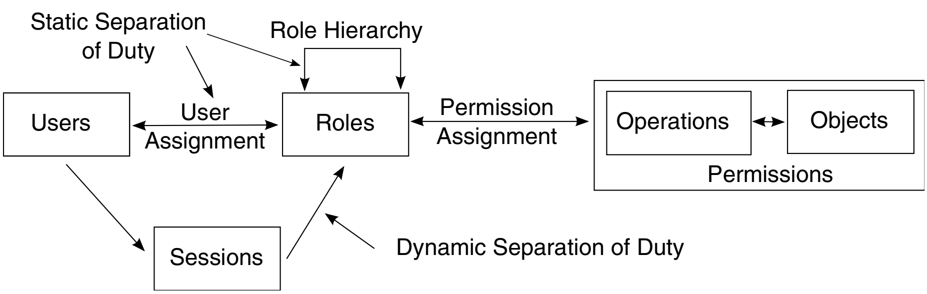



**Fig. 1.** Standard RBAC Model adapted from [3]

Core RBAC defines the basic functionality of roles. It includes five basic sets of data elements, which are users, roles, sessions, objects and operations (cf. Fig. 1). Roles contain permissions for objects (permission assignment), whereas users can be assigned to roles (user assignment). Both types of assignments are many-to-many relations. During a session, a user can activate one or more of their assigned roles. Each session is associated with one user, whereas an individual user can have several different sessions at the same time.

Hierarchical RBAC extends core RBAC with role hierarchies that allow structuring the roles to correspond to functional or organizational hierarchies. In this approach, child roles inherit all permissions assigned to their parent roles. Additionally, the standard differentiates between general and limited role hierarchies. General role hierarchies allow roles to be connected in an arbitrary partial order, whereas limited role hierarchies are restricted to tree structures.

Constrained RBAC adds Separation of Duty (SoD) relations to core RBAC. More precisely, it allows for both static and dynamic SoD. Static Separation of Duty enforces constraints on the assignments of users to roles. For example, two roles can be defined as mutually exclusive, preventing them from being assigned to the same user at the same time. In contrast, Dynamic Separation of Duty constrains the activation of roles for the session that is currently run by the user.

In addition, all different RBAC alternatives contain requirement specifications for administrative functions and review functions. Administrative functions enable administrators to create and delete the RBAC structure and their relations. Review functions provide reporting features such as "All permissions of a user".

# 3 Subject-Oriented Access Control

In this chapter we will give an overview of our approach[1] to access control that differs from traditional concepts presented in the preceding section. One difference is – similar to the S-BPM paradigm – the focus on the subjects of an organization.

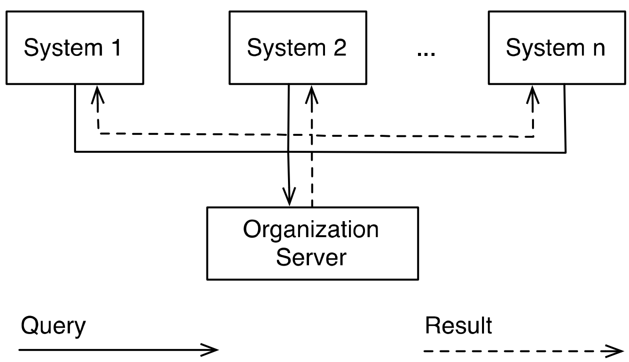



**Fig. 2.** Out-side view of an Organization Server

## 3.1 Overview

The central component in our approach is the organization server (Fig. 2). From an out-side view the server fulfils two tasks [9]. First, it maintains the active components within a company (users, applications, systems). Second it provides a formal organization language (FOL). As a simplified example, an expression in a FOL could look like "clerk(claims department).(Now() - clerk.HiringYear)>10"[2]. On the basis of such an expression, clients are now able to specify access rights or task assignments according to the real world needs. Let us examine a simple policy definition scenario

[1] A formal specification can be found in [6].

[2] We are looking for all clerks working in the claims department that have been working for the company for more than ten years.

first. In Fig. 3, FOL-Expressions are used for defining access permissions. Let us now have a look at the read policy. The general rule is that all managers working for the company longer than half a year can read the daily financial report. The "OR"-term of the expression defines an additional policy exception rule by referring to a specific "ReadFinancialReport"-Flag. At the time a user would like to have access to the secured data object "daily financial report" the client application passes the FOL expression to the organization server. The server resolves the expression to a subset of matching employees that is passed back to the calling application (client). The client will grant access if the user is element of the returned subset.

| Data Object | Read | Write |
|---|---|---|
| Daily Financial Report | Manager(*).(Now() - Manager.HiringYear) > 0,5<br>OR<br>Manager(*).ReadFinancialReport==TRUE | Manager(Controlling)<br>OR<br>Clerk(Controlling).WriteFinancialReport==TRUE |

**Fig. 3.** Access Matrix based on an FOL Expression

The case of task assignment is very similar (Fig. 4). Each activity in a business process is assigned to a set of persons responsible for it. These so-called actors can easily be specified using a FOL expression. At the moment a task has to be executed, its FOL expression is passed to the organization server by the workflow engine. The organization server returns a set of employees satisfying the specification. Based on the context, e.g. the employees' current workload, the workflow management systems decide to whom in the returned subset the task will be assigned.

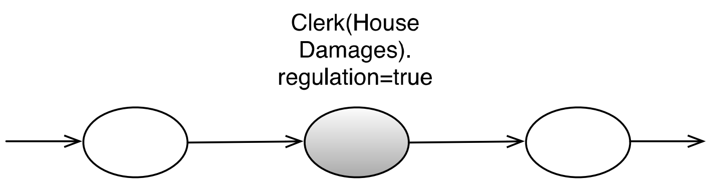



**Fig. 4.** Task assignment using an FOL expression

### 3.2 Subject-Orientation through Levels of Abstraction

Let us now consider a real world scenario. A claims department within an insurance company usually has a manager, a number of clerks and a lawyer. Generally, the lawyer is the deputy of the department head. This abstract specification is depicted in Fig.

In a case study, we examined two concrete departments: one responsible for "Car Claims" the other responsible for "House Damages". Compared to the general structure and policies we observed some differences (Fig. 5).

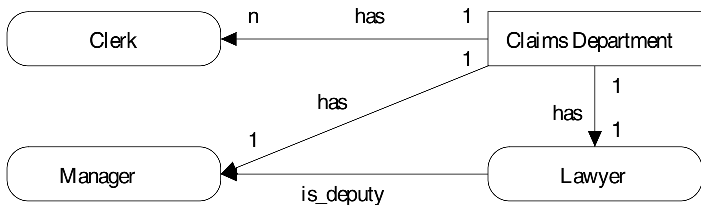



**Fig. 5.** Claims department in general

At "Car Claims" there was an additional secretary position. In absence of the manager, organizational tasks were assigned to the secretary deciding what the next step would be. There also was a change in the deputyship between the department head and the lawyer as well. Byron, the lawyer, had been working in the department for only three weeks and therefore was not very experienced. The clerk Winter has been working in the department for over ten years. Based on that constellation, the department head Smith decided that Winter should be his deputy for a fixed time period. But it was not a general deputyship. The deputyship was depending on some context information like the cash value of a claim for instance (conditions c1 and c2 in Fig. 5).

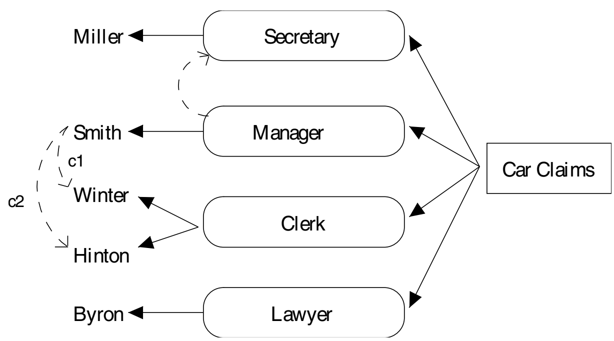



**Fig. 6.** Concrete claims department

When considering the second department "House Damages", we found an interesting mutual deputyship between the lawyers of the two departments (Fig. 6). This observation gets important when thinking about dividing the organization system into types or classes on the one hand and instances on the other. One has to realize that the relationships defined until now are defined on different levels of abstraction [5].

On the top-level, general structural assertions like "a department consists of one to three clerks" are dominant. We call this tier the type level. Knowledge on this tier is based on experience and is changed seldom as time goes by. Looking at real world departments on the second tier things become more concrete and specialized. We are talking of concrete positions and the relationships between them. Please note that the structure according to tier one can be extended or replaced. A detailed discussion of this mechanism can be found in [9]. On the third tier actors that can be intertwined by additional relationships are assigned to the concrete positions. According to the demands of the daily business, the organizational structures on this level are changing more frequently. In the following we will give a brief overview of our algorithm for policy resolution.

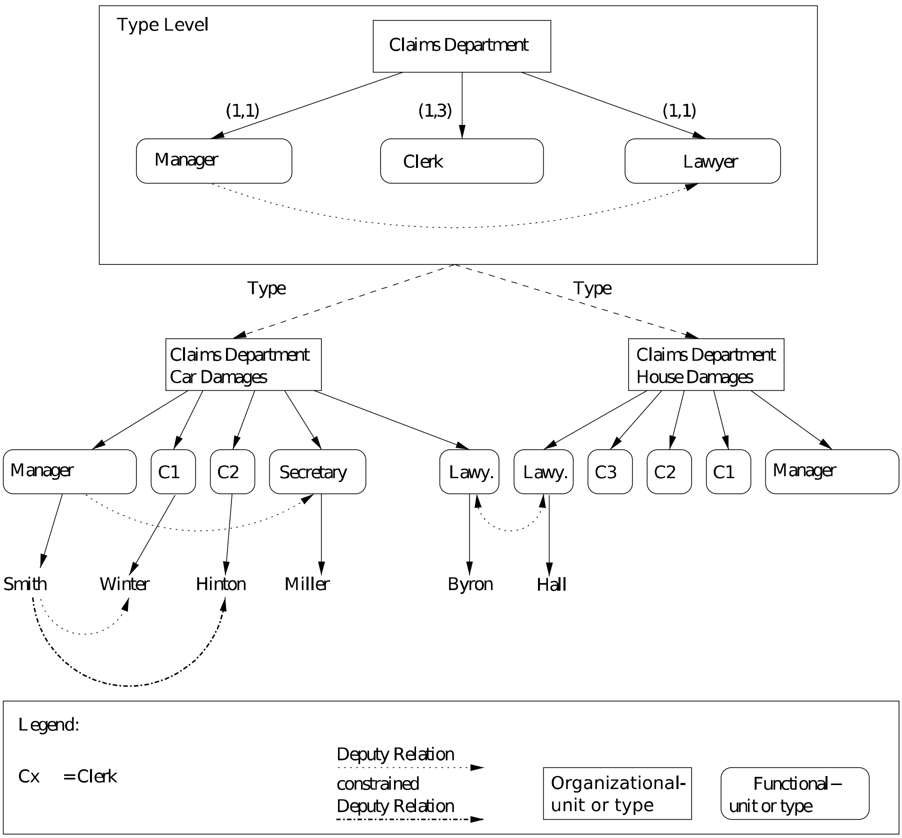



**Fig. 7.** Complete Example

## 3.3 Policy Resolution

Let us assume a workflow management system passes the expression "Manager(Claims Department Car Damages)" to the organization server via the respecting query translator. By traversing the graph in Fig. 6, the algorithm moves to the department "Claims Department Car Damages" looking for a position "Manager". After that the engine determines all the employees assigned to that position now finding manager Smith. If Smith is on job, his identification is handed back to the workflow management system and the search is ended. In case that Smith is not available (e.g. due to vacation or sickness), the algorithm checks for a deputy-relationship between Smith and other employees. Obviously, there are two constrained relationships. If Winter or Hinton appear in the search results depends on if the constraints hold and they are on job. In case of an empty set the algorithm moves to the functional unit manager, looking for a deputy relationship and finding the functional unit secretary assigned to Miller. If Miller is on job her identification will be returned to the workflow management system. If she is not available, the algorithm has the alternative of determining a valid deputy on the type level (tier 1). Let us assume the department is linked to the department type. Within this type the algorithm finds the lawyer as a deputy. It will move back to the instance "Claims Department Car Damages" and checks if there is a functional unit with this name and an actor assigned to that functional unit available. If Byron is on job his identification will be handed back. Otherwise the lawyer of the "Claims Department Car Damages" has a two-way deputy relation with the lawyer of the "Claims Department House Damages". If this functional unit has an actor associated with this unit and the actor is available on job, the algorithm will hand back his identification – here Hall, the lawyer of the "Claims Department House Damages". Otherwise the returned set is empty. In this case the workflow management system has to postpone the execution of the task.

As described in the example, the algorithm starts on the lowest level of abstraction (tier 3). The policies found on this tier are very close to the demands of the daily business and can be defined by the business users in the departments themselves. If policy resolution is not possible, the algorithm moves on to the second tier, where more general rules are defined. Again the department itself can maintain the assignments and rules on this level. Tier 1 represents something like a "last resort" or an exception handling procedure, describing a common denominator. If policy resolution on tier 2 is not possible, the algorithm can be forced to draw the specifications of tier 1 into account.

### 3.4 Implementation

A prototype for verification and validation purposes has been developed (Fig. 7). It became obvious that the resolution of complex role expressions (especially when using lots of user-defined relationships within the resulting organization graph) can lead to significant performance issues. On the one hand, this problem can be solved by the usage of a structural reduced subset of the primarily FOL. On the other hand, the real world policy rules are not so complex as modelled in our test scenarios. When going to praxis, one big issue was the problem that companies already use a directory service like Microsoft's Active Directory and that the various applications come along with their own policy management sub-module that, in general, is a closed shop. For this reason, we are actually working on a method for replicating the policies specified in our organization server to the connected client applications. Specifically, we are dealing with the question of how Microsoft's products can be connected to our organization server and how Microsoft's Active Directory can be extended to represent our notion of an extended directory service.

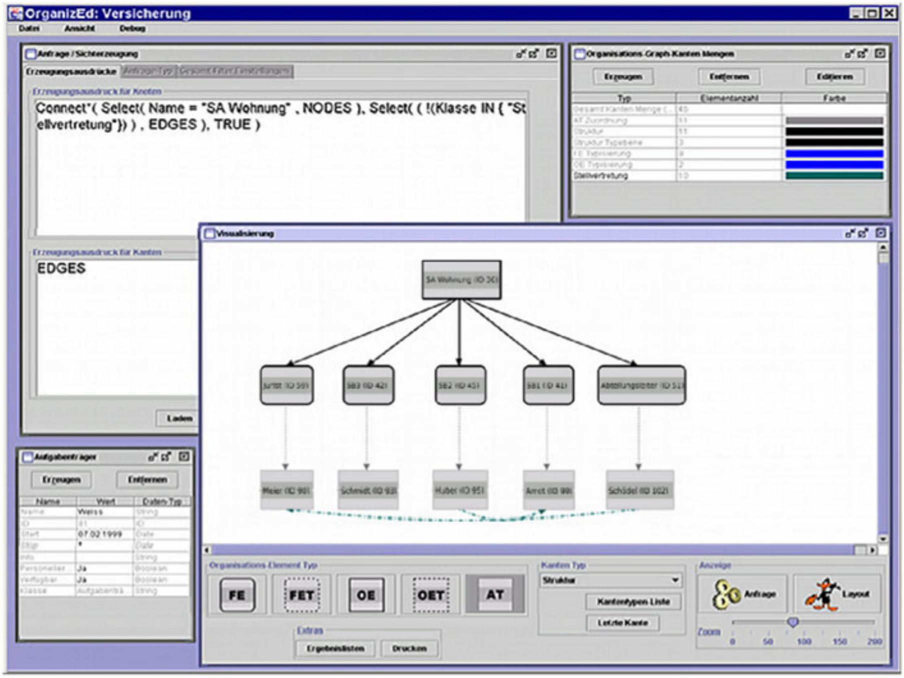



**Fig. 8.** Prototype of the Organization Server

## 4 Conclusions and Outlook

In the paper we presented an approach to a flexible access control that is more subject oriented and therefore more decentralized than traditional concepts. The next logical question is how our approach can be integrated with the S-BPM method. One of the main aspects of S-BPM is the definition of the subjects as process-specific roles, requiring yet another mapping between the organizational roles already defined in the IT landscape and these kinds of processes. In their case study [7] Schmidt et al. describe how subjects are assigned to groups defined in the enterprise-wide directory based on the Lightweight Directory Access Protocol (LDAP). This could be a good starting point for coupling both worlds using our replication approach presented in section 3.4.

On a more detailed level, Sellner and Zinser describe in [8] an approach of how to align the subject-oriented business process management perspective with the general paradigm of business rules. By identifying and utilizing a common business vocabulary based on XML Schema Definitions (XSD), it is shown that business rules can be enacted throughout business processes modelled within the S-BPM environment. This demonstrates the versatility and extensibility of the whole subject-oriented business process management approach. Business rules individually affect multiple processes. They can, however, be enforced in the individual processes due to the extensibility of the S-BPM model. In a similar fashion, access control specifications as modelled within the organization server might be enforced throughout the process landscape.

An additional example of how our approach can be integrated into the S-BPM perspective is depicted in figure 9 showing a simplified model of a business trip according to [10]. The process explicitly includes the involved roles and corresponding business object (request for business trip). At the point the employee sends the request to the supervisor, our proposed policy resolution algorithm evaluates the role "supervisor" and returns the person – respectively the set of persons – corresponding to the supervisor role. Through the usage of a deputy-relationship the blocking of the process can be avoided, if none of the supervisors is on job.

Looking at the security of the business objects the S-BPM approach could be extended using the proposed FOL expressions in combination with the access matrix (see Fig. 3). This way, access to the business object can be defined and the permissions on a more detailed level – i.e. to the data structure fields encapsulated within the object – can be managed as discussed in [10].

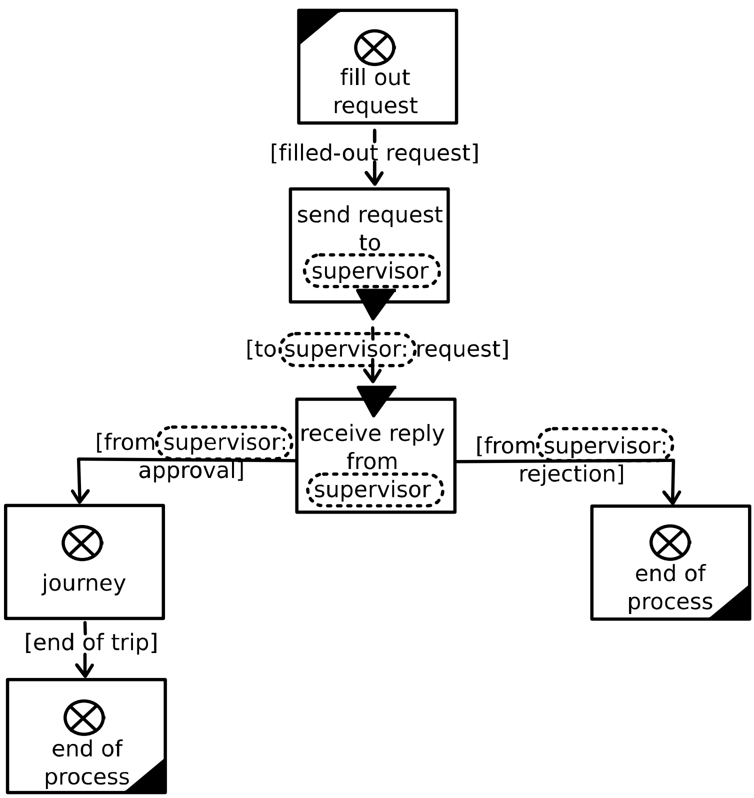



**Fig. 9.** "Business Trip" process model adapted from [10]

## References

1. Kurz, M., Fleischmann, A.: BPM 2.0: Business Process Management Meets Empowerment. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 54–83. Springer, Heidelberg (2011)
2. Kern, A.: Advanced Features for Enterprise-Wide Role-Based Access Control. In: Annual Computer Security Applications Conference, p. 333 (2002)
3. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST Standard for Role-Based Access Control. ACM Transactions on Information and System Security (TISSEC), 224–274 (2001)
4. Wolfgang Martin Team S.A.R.L. Martin: Active Compliance Management with Subject-oriented BPM (S-BPM) (2010) (unpublished manuscript)
5. Lawall, A., Reichelt, D., Schaller, T.: Intelligente Verzeichnisdienste. In: Herausforderungen an die Wirtschaftsinformatik: Betriebliche Anwendungssysteme (AKWI), pp. 87–100 (2011)
6. Lawall, A., Reichelt, D., Schaller, T.: Flexible Role-based Access Control. White Paper. Institute for Information Systems, Hof (2010)
7. Schmidt, W., Fleischmann, A., Gilbert, O.: Subjektorientiertes Geschäftsprozessmanagement. In: Reinheimer, S. (ed.) HMD – Praxis der Wirtschaftsinformatik, vol. 266, pp. 52–62. dpunkt, Heidelberg (2009)
8. Sellner, A., Zinser, E.: Establishing Conceptual and Functional Links between S-BPM and Business Rules. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 121–133. Springer, Heidelberg (2011)
9. Schaller, T.: Organisationsverwaltung in CSCW-Systemen. Dissertation Universität Bamberg, Bamberg (1998)
10. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement, Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern (2011)

# Building a Conceptual Roadmap for Systemic Change – A Novel Approach to Change Management in Expert Organizations in Health Care

Martina Augl

TILAK - Tiroler Landeskrankenanstalten GmbH
Organisationsentwicklung, Abteilung Medizin, Anichstraße 35, A-6020 Innsbruck
martina.augl@tilak.at

**Abstract.** Complex systems, such as the Austrian health care system, mainly are dynamic systems: They do not only face changes with respect to customer orientation and budgeting, but also lack organizational goals and transparency of business processes. Systems thinking allows constructing a conceptual roadmap for change management, facilitating an adaption to these changes in a socially acceptable way, even under tight economic constraints. It considers the system at hand as a social systems framework in which organizations are viewed as communication systems. Sustainable change therefore requires transforming existing patterns of communication to contextual collaboration. The different professionals within (health) expert organizations need to negotiate and agree on interactions empowering the organization for high-quality patient care. ”Boundary objects” serve as mediums for coordination, translation and creation of shared meaning, while putting communication and interacting subjects to the centre of process and change management. In this way project designs can be restructured for systemic change.

**Keywords:** Austrian health care system, boundary objects, change management, communication systems, complex systems, empowerment, framework, health expert organization, learning organization, living organization, organisational development, participation, performance, process modelling, subject-oriented business process management.

## 1 Introduction

The Austrian health care system is a typical example for a service sector that comprises various kinds of expert organizations embedded in a very complex, nested environment. Hospitals and in particular university clinics, are expert organizations. In their work setting many different expert groups need to collaborate, while each of them has a very strong identification with a certain profession and expertise. One major challenge of these multi-expert organizations is to overcome differences with respect to professional languages, work practices, problem solving procedures, and work cultures, in order to share goals and develop mutual understanding in their daily business. Furthermore, system dynamics require continuous adaption and systemic change. [1, p. 54ff]

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 43–61, 2012.

So far change management has been of little importance for the professionals compared to their expert knowledge work. Nevertheless, it is still an open question how expert organizations can change in a target-oriented and sustainable way when adapting to changing environments and increasing organizational performance. Organizational development based on Systems Thinking [2] can be applied to address this issue. In this contribution, a framework is introduced facilitating systemic design and management of participative and sustainable organization development projects within complex structures. It requires involving and empowering the different expert groups at hand. Moreover it addresses the intertwining of business processes, communication patterns and task-specific actor- or system interaction. Thereby, business processes serve as reference objects in communication as well as medium for change. When using this novel approach organizational development is led through systemic interventions rather than case-specific project designs. Consequently, it allows the experts involved co-creating organizational future via direct participation.

In the following the case being used to demonstrate the approach is described, namely the Austrian Health Care System (section 2), Then, in section 3, the systemic perspective on organizations and management of change is detailed before the development of the framework is tackled in section 4, and some results of systemic change work are presented in section 5. Section 6 concludes the paper with rephrasing the achievements in the context of the objectives.

## 2 A Case of Complexity – The Austrian Health Care System

In 2007 the Austrian health care system was acclaimed “the best health care system in Europe” [3]. Nevertheless - or perhaps for exactly that reason - it faces some upcoming challenges: [4, pp. 326ff]

- Demographic trends
- Medical-technical improvements and increasing complexity of treatment processes
- Ever more patients requesting health as a “service”
- Exploding expenditures on health because of existing inefficiencies and many health service structures existing in parallel, while at the same time enabling free choice of physicians and therapy within in- and out-patient care
- Rumour about a lack of health professionals and physicians, including nurses, in the near future ...
- … while at the same time issues with overstretched employees
- etc.

For these reasons, the Austrian Government is regularly engaged in developing health reforms, with the ultimate goals of increased efficiency, attaining structural improvements and targeted use of resources. Often, the reforms focus on public non-profit hospitals due to their perceived importance in the Austrian health care system. Particular to the Austrian health care system, responsibilities are split between the Federal Government and its provinces, defined in the Agreement according to Article

15a of the Federal Constitutional Act on the organization and financing of the health care system from 2008-2013 (BGBI.I Nr. 105/2008): "The Federal Government is in charge of defining the legislation for out-patient care (physicians in private practices). Responsibility for in-patient care provided in hospitals is shared between the federal and the provincial levels: the Federal Government lays down the legislative framework, whilst the provinces are in charge of defining legislation on enforcement as well as ensuring implementation. All regulations regarding pharmaceuticals, pharmacies and medical devices, as well as health professions (e.g. education of physicians) and structural policy are the responsibility of the Federal Government. Public health services and administration are jointly provided by federal, provincial and local authorities." [5, p. 6]

This fragmentation may be one of the causes of the lack of effect of past reforms. In brief, the past health reforms targets may be summarised as follows: [6]

- Ensuring universal high quality medical care through social solidarity
- Increasing efficiency and ensuring customer-oriented structures
- Establishing health promotion as a high priority goal

In spite of all reform efforts, a major criticism often levelled is of a lack of clearly formulated objectives for the Austrian health care system, especially on the part of the Federal Government. Consequently, it is difficult to measure if these goals have been achieved. States do partially develop objectives, but at very different level of detail and quality level, and in an uncoordinated manner. According to Josef Probst, Deputy Director General of the Main Association of Austrian Social Security Institutions, this is the greatest barrier to achieving real improvement and cause of the current lack of direction. [7]

In 2006 the Austrian Ministry of Health published for the first time the ÖSG (Österreichischer Strukturplan Gesundheit, Austrian Health Care Structure Plan). The ÖSG includes the coordination of resources across all levels of service provision and sets the framework for the 32 existing care regions.[1] Main goal of the ÖSG is the provision of locally available medical services as well as the concentration of special services into "competence centres" through a certain predefined minimum number of treatments. [8] Each province must provide its own structure plan for providing and committing care according to the overall master plan – a RSG (Regionaler Strukturplan Gesundheit, Regional Health Care Structure Plan). Any further adoption and development has to be coordinated between the individual province and the insurance institution associated with its health care platform. Some of the main problems with these nine individual plans are the different structures, base years, planning horizons and in- or exclusion of in- as well as out-patient care. In order to at least get a comparison and current overview of capacity- and large-device planning within the ÖSG 2010, the Austrian Ministry of Health is now implementing RSG

[1] Since then the ÖSG 2010, the third extended version with a planning horizon until 2020, was published. It contains, in particular, structural and capacity planning for hospitals, a high-technology investment plan and planning guidelines for in- and out-patient care as well as for the rehabilitation sector. [8]

monitoring. Its aim is to outline the current content and results of the different regional structure plans or the provincial hospital plans – as much as is available – through a standard structure. [9]

The local implementation of a provincial RSG is negotiated between the local government and the specific host organizations of the regional hospitals. The next step is to integrate these requirements into the correlating strategies of the host organization. These in turn must be broken down into concrete targets for the individual hospitals.

According to the Federal Constitution Act, Article 15a BGBI.I Nr. 105/2008 also covers an agreement between the Federal Government and its provinces concerning measures to reduce costs, increase efficiency, more specifically supervision of the health care system, as well as evaluation of these measures. In detail, these measures are about: [10]

1. Reducing the amount of hospitalisations/admissions and readmission, as well as optimising clinical day-care treatments.
2. Enforcing new types of organizations within hospitals, such as day or week clinics and interdisciplinary occupancies, while maintaining specialist medical responsibility or similar types of provision.
3. Measures to improve coordination between individual hospitals and primary care and avoid duplication of structures.
4. Taking measures in primary care to ensure balanced, patient-centred care.
5. Improving efficiency in the use of medical devices and drugs.
6. Aligned systems of remuneration.
7. Increasing the percentage of LKF (Landeskrankenhaus Finanzierung) accounted funds incrementally based on LKF evaluation results.
8. Utilising efficiency potential by purchasing via the Federal Procurement Organization (Bundesbeschaffungsgesellschaft).

Nevertheless, critics do not see any leveraging effects for efficiency by means of more detailed planning. They call for higher transparency of information, thus gaining patients as allies and partners in the fight against high costs and poor quality. According to Julian M. Hadschieff, spokesperson for the Austrian Platform for Health Industry, transparency drives change. In order to succeed, patients must be more involved and take higher personal responsibility. This should all help to create stronger solidarity within the system. The overall aim is to intensify competition for quality and reduce bureaucracy in the interest of the patients. Cooperation of all stakeholders within the health care system will be a necessary precondition to this. [11]

This highlights the dynamics and need for change impacting health care institutions, in particular public hospitals. The structural change and modified conditions facing health care systems force organizations to take extraordinary measures to adapt and requires strategic (re)alignment or (re)organization. [1, p. 54ff] In order to implement new strategies, in many cases (change) projects are carried out.

Figure 1 maps the diverse stakeholders and influential groups in an Austrian hospital host organization which is also in charge of, among others, a university hospital. The depicted groups are crucial for both the organizational design (organizational processes and communication systems) and for potential organizational development projects within a certain special clinic. Therefore they impact the whole university hospital as well as the specific project mentioned.

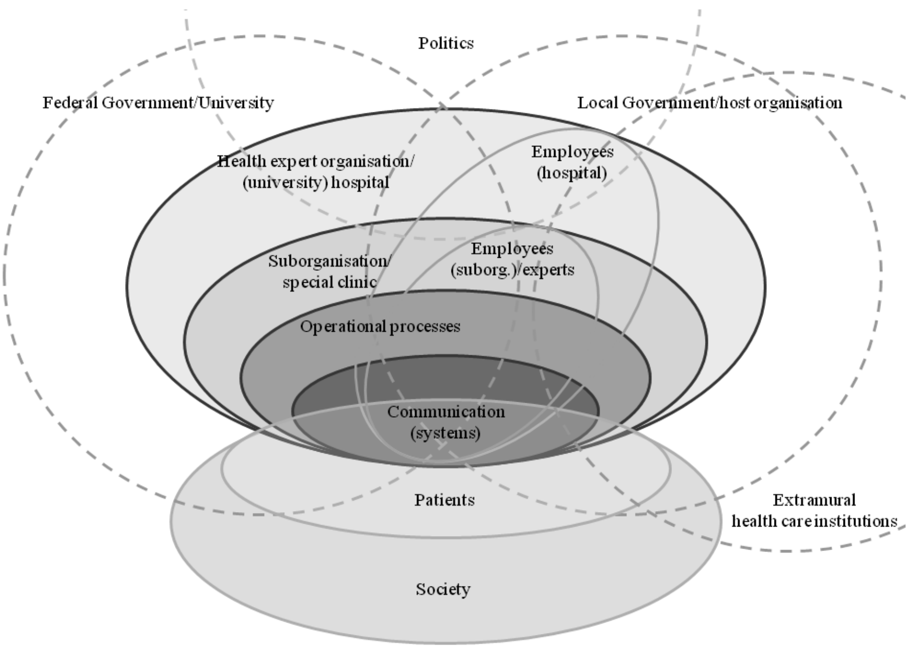



**Fig. 1.** Stakeholder-map of an Austrian university hospital

*Main subject: Research within this context has to challenge relevant questions/problems related to the implementation of target-oriented, effective development projects within expert organizations (e.g. hospitals) that face little transparency and imprecise target-requirements, in order to contribute to an increase in performance.*

"Performance" as used here should be understood from both an economic and a sociological perspective. It denotes a valued contribution for achieving the targets of an organization [12, p. 8] "by an individual, team, organization or process" [13], as well as that specific personal, professional and social competence people are able to activate under certain social and emotional conditions [14, p. 35] Central to this is the consideration of performance in relation to the competence aspect (mentioned above), and accordingly towards action and future orientation. These are regarded as important organizational characteristics by relevant stakeholders. [15, p. 199]

## 3 View on Organizations and Their Changeability

Organizations may be understood as social systems, or to be more precise, as communication systems.[2] For that reason, organizations should not be seen as physical, autonomously existing and acting entities, but as (communication) processes that are and need to be continuously carried out and thus continued. The smallest entity of all social systems is communication which couples two or more actors or their communication acts. At its very heart is not the transfer of messages, but the coordination of actors and their actions. The operational processes that may be observed in an organization, meaning the patterns of action, can be explained as a result of communication. As Fritz B. Simon points out, communication is responsible for the dovetailing of actions of different actors (communication participants). [17, pp. 16ff]

Furthermore, organizations should be understood as living systems. This is highlighted in the process of communication, namely that after previous communication further communication (or the prospect of further communication) follows. Humberto Maturana calls this ongoing auto-reverential process “autopoesis”. The result is a network of internal, self-referential (communication) processes that from an outside point of view are perceived as compound units that differentiate themselves from their environments. [17, pp. 23ff]

In order to establish real change, which means breaking up the existing patterns and processes, the system has to be “perturbed” or irritated. From the systems view this can be seen as “interference” or, positively speaking, as a “stimulus”. Such interference can, for example, be initiated by means of or as a part of an organizational development project. This entails the chance of a new pattern of communication developing (as a result a new pattern of actions) that can be thus, in turn, reproduced. The one thing that cannot be predicted – not even by the system itself – is the mode in which change materialises and becomes effective. In order to avoid this irritation solely being perceived as an external stimulus, it is necessary to involve the actors of the system into a “transformational project” by means of concrete tasks, roles and communication. Thereby, local experts’ knowledge and ideas for improvement can be integrated as soon as possible. Participation subsequently increases the probability of acceptance and sustainable implementation of organizational and technical changes later on. [18, p. 654]

Additionally, the behaviour of the actors of the system realises the new pattern or, in other words, “bring them into communication”. However, only if the behaviour of an individual is given meaning by another person can organizational impact be implied. [19, pp. 97ff]

Representives of “modern” organisation theory (e.g. Johannes Steyrer) assume that traditional concepts of change, such as Kurt Lewin promoted in his three phases of change: “unfreeze” – “moving” – “refreeze”, are no longer adequate in times of

[2] According to Niklas Luhmann communcation systems have to be seen as completely closed systems, consisting of the components information, utterance and understanding (including misunderstanding), that specifies its elements and structures itself. [16, p. 118]

permanent change. Because of this, a "learning organization" that is "chronically unfrozen" has been postulated. [20, pp. 17ff] A learning organization is an organization "… that is continually expanding its capacity to create its future" [2, p. 14].

The goal of continuously working at further developing the organization will not be sustainably successful in the running of externally initiated projects. Crucial to its success is the empowerment of the whole system, so that it can identify when such stimulus is necessary and so it can be designed appropriate to desired targets. Organizational communication and learning is based on the interacting individuals within the organization. Therefore, it is necessary to directly involve the actors of the system und to empower them. Consequently, it needs to be ensured that the design of an organizational development project and the subsequent establishing processes themselves serve as sources of empowerment. It is essential that those capabilities of individuals enable them to master an existing or forthcoming challenge, to meet own needs, solve problems, and obtain the resources required in order to control decisions and actions that impact on their goals are strengthened or activated. [21]

Empowerment must focus on the whole organization. This includes hospitals, in particular, as the requirements for hospital management to act entrepreneurially within the existing conditions have become more complex. One major problem hospitals, as expert organizations, have to cope with is the low relevance of management – even in the minds of leading personnel – and a lacking of change management capabilities within the organization. [22, p. 343] Additionally, hospitals do not only contain one single expert organization, but at heart are composed of many expert organizations with enormous functional differences. Each profession is completely different, is subject to local systems and pursues an entirely different logic. All this is situated within the conflict-prone triangle of cure, care and cost. In this field, physicians, nurses and administrative personnel have to constantly choose between ethical obligation and economic pressure – and priorities differ. Modern process management however does not stop at professional borders, but requires interdisciplinary cooperation. [20, p. 8]

## 4 Developing a Framework – Common Worlds, Common Views, Common Future

Behind this background is the question of how to meet the demand of these very different expert groups with different background knowledge, experience, professional languages, views and mental models jointly developing their own organization. The concept of "boundary objects"[3] offers a possible frame of thinking, analysis and design. Boundary objects enable different groups to share their knowledge and information, as they allow the parties involved flexible interpretation and usage.

---

[3] The concept was first published by Susan L. Star und James R. Griesemer in 1989.

Star und Griesemer define boundary objects as follows: “This is an analytical concept of those scientific objects which both inhabit several intersecting social worlds … and satisfy the informational requirements of each of them. Boundary objects are objects which are both plastic enough to adapt to local needs and the constraints of the several parties employing them, yet robust enough to maintain a common identity across site. They are weakly structured in common use, and become strongly structured in individual-site use. These objects may be abstract or concrete. They have different meaning in different social worlds but their structure is common enough to more than one world to make them recognizable, a means of translation. The creation and management of boundary objects is a key process in developing and maintaining coherence across intersecting social worlds.” [23, p. 393]

Boundary objects allow the translation of different perspectives and thus enable or facilitate communication and cooperation as well as the creation of “shared understanding“ or “common ground“. They are common reference objects of different types within the exchange between individuals or groups. Sometimes they are initially created or get their specific meaning through the interaction of the different actors. If there any conflicts occur in attributing meaning, the conflicts have to be negotiated. But it is exactly this utilisation of a boundary object in discussions or negotiations that determines their function, not their mere existence. [24, p. 6f]

Besides specific people, artefacts such as documents, concepts or rules, (business) processes can also take on the function of boundary objects and thus become central to both communication and action for empowerment. In the case of a hospital a specific clinical pathway (e.g. for colorectal cancer) can be understood as a boundary object. For each health profession, even for management and patients, the pathway has a certain meaning and triggers specific activities, such as deciding on the next steps of therapy, ordering the relevant examinations and documenting the steps in order to inform colleagues also of other professions. Globally, they form the base for interaction.

In order to increase organizational performance, e.g. within a special clinic as an expert organization of a certain hospital organization, on the one hand, (business) processes can become target-oriented by the means of boundary objects. On the other hand, the actors affected have to be empowered to obtain competence in recognising the need for change and in carrying out organizational development projects (change-capability), and therefore lead to become a learning organization.

Boundary objects can serve as a framework for explanation and design. Together with the participation of actors, they turn (business) processes and communication into resources for change.

In order to develop a framework that can deal with complexity and allow derivation of clear targets as well as design elements to facilitate change processes, it is necessary to analyse certain questions such as:

- Which social systems or environments are of relevance for a given organization?
- What impact do they have on the organization or its operational processes and communication patterns?
- Is it possible to identify reference objects of relevance in terms of boundary objects? If so, what special role can be attributed to them?
- What can be done in order to facilitate the utilisation of boundary objects as mediums for empowerment and change?

A framework provides a research scope that includes and focuses on all relevant elements concerning a defined problem or question. In the case of a special clinic in an Austrian university hospital, the first step is to identify, as done in section 2, the relevant stakeholders in terms of social systems.

The stakeholder-map of the larger system (see figure 1) may be seen as a map of "social worlds"[4] which allows the identification of some direct as well as indirect impacts these worlds have on each other (see figure 2). These impacts can serve as possible pertubators for change. The diagram shows two different forms of directional arrows: solid lines are direct impacts while dashed lines are indirect ones. Taking into account the described steering intention of the health care system in section 2 (solid lines), it may be seen that politics set binding requirements for themselves in the role of the Federal Government as well as for the Local Government. Further on, a certain medical university or host organization gets instructions for the allocation and control of resources for a university hospital. The federal and provincial hosting organisations specify these instructions into targets and (partial) strategies. These are binding for the university hospital. Consequently, sub-organizations such as a certain special clinic may face the need for change. Responding to the need for change, either the host organization or the special clinic initiates a concrete organizational change project. Changing the existing patterns and processes determined by communication is crucial to sustainable change within the special clinic as a social system. Therefore, this is the starting point for real change processes. Additionally, communication and operational processes are where direct interaction with patients in terms of a relevant social environment of the special clinic takes place.

Furthermore, there are some indirect impacts that affect top-down impact targets: Expectations of society regarding the "service health" influence politics as well as employees or patients of a hospital as they are part of society. Hospital and patient expectations (the latter by means of their interaction with hospital employees) affect the operational processes within a certain area. And last but not least, employees of other special clinics, over different interfaces, influence expectations of the communication system of this concrete area.

The mapping of the inherent dynamics of the existing system facilitates the conduction of a focused strategic (re-)organization (circles in bold as depicted in figure 2).

---

[4] It should be noted that every single group of stakeholders again consists of different social worlds.

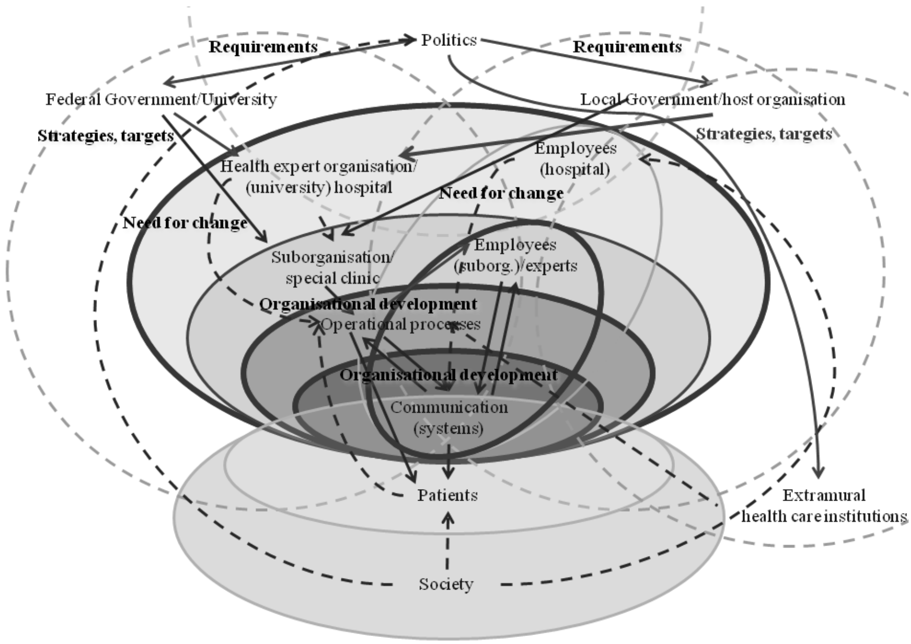



**Fig. 2.** Impacts between different „social worlds" in an Austrian university hospital and derivated project focus

After mapping the general context, we must now consider which social systems contribute direct environments to the special clinic and therefore are of particular importance to (target-oriented) change.

The direct environments of the special clinic are:

- The (university) hospital in which the clinic is embedded
- Its federal and provincial host organizations or employers
- Employees belonging to different professions
- Patients within the catchment area and beyond according to its reputation as a university hospital
- Other special clinics within the university hospital
- Extramural up- and downstream health care institutions such as specialists, family doctors, nursing establishments, ambulance and safety organizations, etc.

As regards the existing complexity and nested structure of the health care system, only the direct impacts of its immediate social environments, that the actors comprehend, affect the special clinic (taking into account that indirect impacts do not lead to less important influences). Most of all, the more or less explicit strategies of the host organization and the hospital (can or should) cause the need for change (reorganization) in processes and communication patterns of the specific clinic. According to the explanations in section 2, the given requirements, strategies or targets are mostly very imprecise or are not charged with any deeper local meaning that leads to local implications. Yet it is important to be aware of the source of certain strategies and targets or requirements and derivations of the health care policy and their relationships, in order to actively make them part of communication. A graphical mapping of the entire system of interaction illustrates this (see figure 2).

Boundary objects serve as important vehicles connecting different social worlds and producing coherent targets. The social worlds of a health expert organization are, for example, its different health professions such as physicians, nurses, therapists, psychiatrists, administrative personnel as well as overall hospital administration. Furthermore, these clusters are characterised by great differences between special disciplines and traditionally strong hierarchical structures.

A properly formulated hospital strategy and derived targets can serve as boundary objects, as they first of all are globally understandable so as to be potentially compatible with all social worlds. Effective implementation, however, requires local attribution of meaning, resulting in two challenges: Firstly, enough meaningfulness for members of a social world, and secondly, an attribution that is adequate (= coherent) as regards the intention of the sender. In order to ensure the expansion of abovementioned strategic goals of the health care system into other social worlds, meaningful targets must be defined. Their main task is to make sense on one hand and foster commitment on the other.

The only possible way to create commitment is by bringing something (the new) into communication and sustainably realising new patterns by according attribution of meaning and repetition. [19, pp. 91ff]

## 5 Systemic Change Work in Practice

In the case of a hospital an excellent target example would be the increase of planning quality in the daily scheduling of physicians. This turns out to be a central bottleneck within the daily routine of a hospital and its special clinics. Taking a deeper look, it may be seen that the necessary processes behind scheduling often produce unnecessary loops in coordination and a lack of transparency in the different tasks in the daily routine. All this ties up a lot of (communication) energy und time in clarifying the real distribution of tasks – especially against the background of many unpredictable emergencies. The solution to this problem also contributes to the globally promoted goals of health care reform, namely the increase in the efficiency and customer orientation of hospitals as well as increased transparency (see section 2 and table 1).

All this puts the focus on the question of how something can get into communication and how it can succeed in coordinating the required actors. Such a process for the realisation of the daily scheduling of physicians can take the role of a boundary object and thus facilitate interprofessional cooperation and modern process management. Most importantly of all, this process has to be visualised in order to make focused communication with it possible. Traditional approaches claim Business Process Management (BPM) and the use of corresponding tools to be effective for modelling and designing processes. However, traditional modelling tools do neither support immediate hands-on experience of processes, nor stakeholder-relevant visualisation. [25, pp. 161, p. 194] Modelling the process using an alternative approach called S-BPM (Subject-oriented Business Process Management) allows subjects to be placed according to their role in social systems, as well as communication to be put to the centre of attention.

**Table 1.** Target structure using the example of "increasing efficiency"

| | |
|---|---|
| Global goal of the health care system | Increase in efficiency, customer orientation of hospitals, increased transparency |
| → Strategy on hospital level | Optimised usage of available resources |
| → Local target | Increased planning quality, keeping to schedules, overview available for all professional groups allowing good planning for their own benefit and for the benefit of patients |

Subject-oriented Business Process Management (S-BPM) makes use of a complete linguistic description of business processes in terms of „subject – predicate – object" (complete sentences). Therefore, the only thing necessary in order to become competent in modelling processes with S-BPM is a command of the natural language. The familiar usage of complete sentences, their close connection to actions and actors exchanging task relevant information, means S-BPM is a method which may be rapidly understood and learnt. Thus an active participation in the organizational development becomes possible. The core elements of a subject-oriented model are those of communication: the subjects (actors or "roles") involved in the modelled process, the interactions and the shared messages between them, as well as their behaviour. [26, pp. 23ff] Figure 3 gives an example of the biannual classification process of physicians as a sub-process of the scheduling process. The grey boxes represent the subjects or roles in interaction. The directional arrows depict interactions to which sent or received messages are attached. This represents the "communication view" of the sub-process. The example of the biannual classification process depicts different subjects such as the clinical director, the director`s office, the office for teaching, the senior managing physician, physicians, etc. and their interactions based on sent or received messages in order to produce the classification of physicians for the upcoming semester.

A "behavioural view" is also drawn for each subject or role, which details the necessary behaviour or workflow of a certain subject in order to fulfil a task. Figure 4 shows the behaviour of a certain physician as soon as he or she receives a teaching request from the office for teaching (in-depth look at the subject "physician" highlighted in figure 3). The light grey boxes represent different functional states the physician moves into, namely a sending or a receiving state. In response he or she conducts different internal tasks and moves to the next state. Messages thereby serve as state transitions.

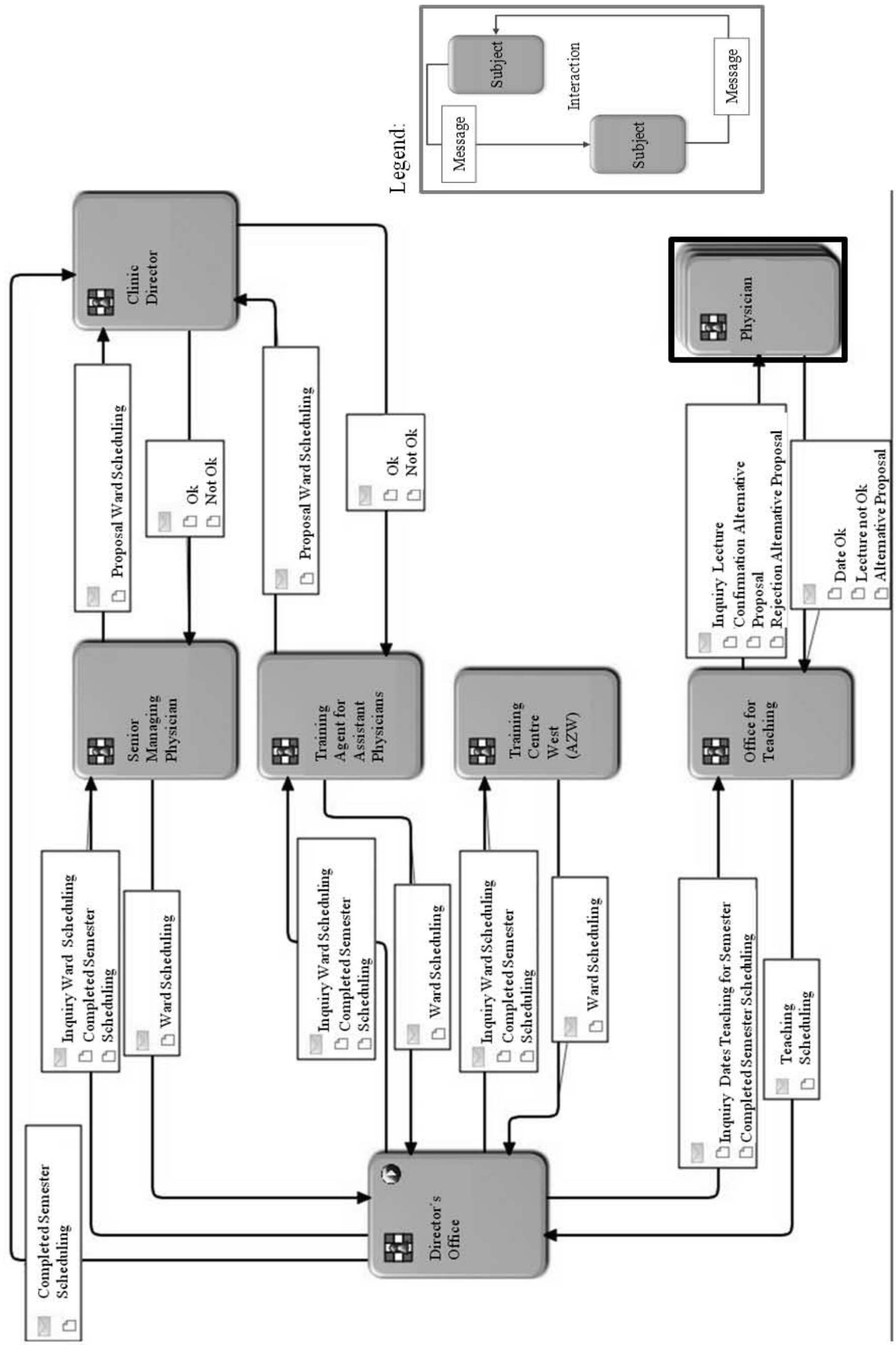



**Fig. 3.** Communication view in S-BPM for the example of the biannual classification process of physicians (illustrated in Metasonic Suite)

Modelling enables us to illustrate an extract of reality in reduced complexity by means of a special medium, while, at the same time, not losing the target-specific relations. Thus it allows participants involved to build a relation associate with their tasks and to express their individual views on it in an effective and efficient way. The description in complete sentences makes the process available for immediate execution by the modelling subjects themselves. Monitoring of the execution and feedback loops facilitate continuous organizational development design. The received models represent the learning process that has taken place over time. [26, pp. 42ff]

The design of the “process for modelling the process” in a subject-oriented way is an essential precondition for success. In this way, it is possible to highlight the required roles regarding the specific requirements and the realisation of the process and to embed them based on communication. [26, p. 41] A conscious design of the process for process modelling is central to an adequate creation and utilisation of the emerging reference objects in terms of boundary objects. This implies a special handling of S-BPM and subsequently facilitates the development of common meaning in the course of process modelling and leads to an empowerment of the participating actors (e.g. via the analysis and validation of the process under development by the means of a “Value Network Analysis” [27]). The direct experience through participation in the development and implementation of the new communication pattern shows up as “two-fold empowerment”: an empowerment both of the participating actors and of the social system as a whole.

Taking our example of the daily scheduling of physicians the communication process, the analysis of the underlying communication patterns can be visualised via S-BPM as shown in two examples above (figures 3 and 4). This also allows process execution by means of certain workflow management systems (e.g. Metasonic Suite[5]), but even an adequate visualisation of the communication relationships (e.g. via Holomapping) allows discussion. Thus common attribution of meaning in the course of negotiation processes becomes possible; as well as the investigation of relationships. Furthermore, changes in the process towards the desired result can be conducted very easily and intuitively. The communication pattern of the analysed “real” and an “optimised” possible process based on the daily scheduling of physicians can be illustrated in a way that is close to natural language and therefore understood by different people. [25, p. 32] This makes it accessible to discussion, validation, negotiation, and change. Necessary preconditions include the participation of actors, focussing on actors and their core processes while taking into account the “big picture” of communication patterns via the framework, integrating targets in common overall validation and rapid prototyping of the new process.

In summary, our example of the realised new common communication pattern for daily scheduling of physicians and its jointly feedback-based optimisation regarding its impact and achievement of the new process can contribute significantly to increased performance of the special clinic and beyond.

[5] See www.metasonic.de

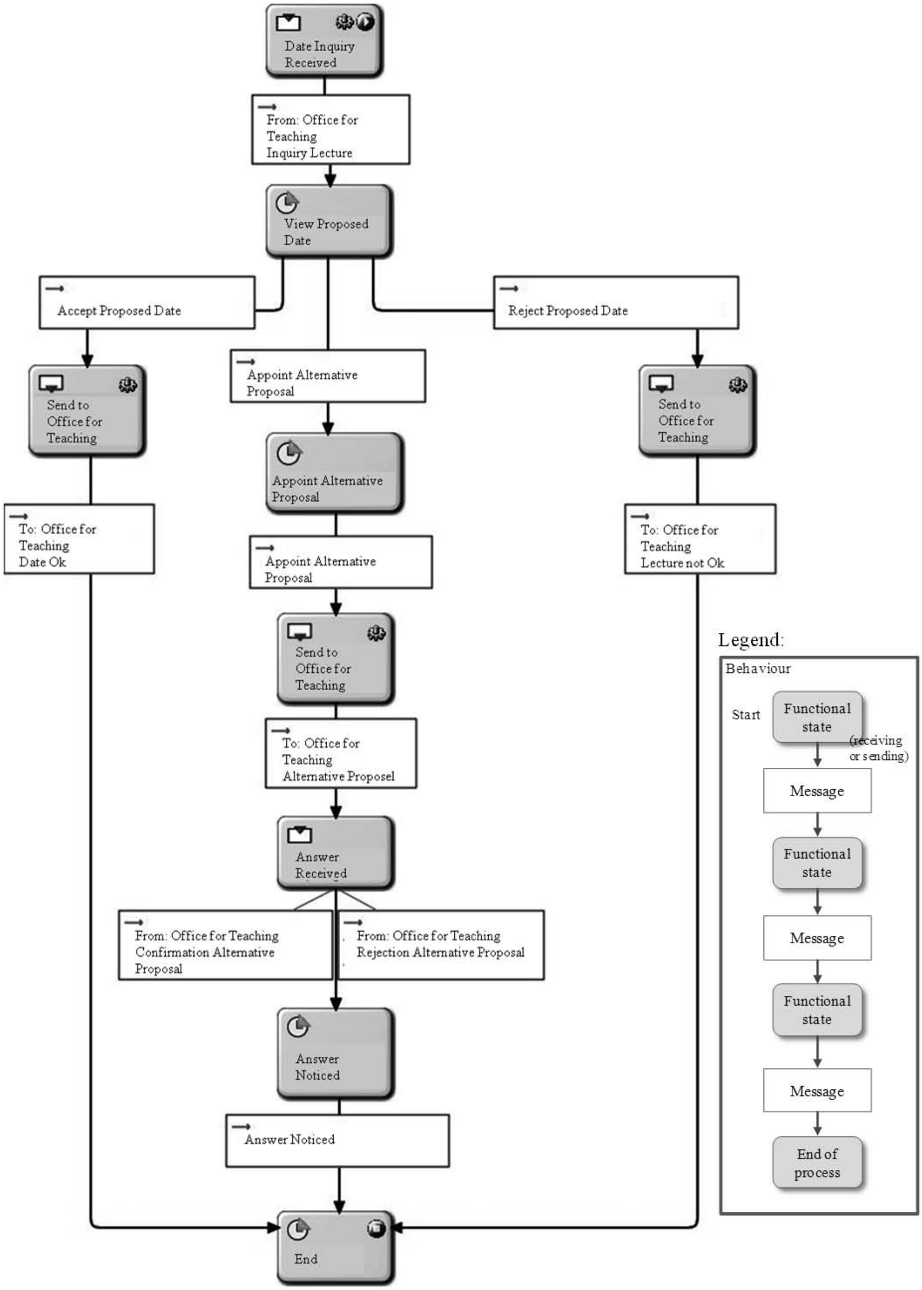



**Fig. 4.** Behavioural view in S-BPM for the example of the tasks of a physician within the process of coordinating the teaching schedule (illustrated in Metasonic Suite)

## 6 Conclusion

Target-oriented change of an expert organization in terms of increase in performance requires (mutual) understanding, a focus on change and binding implementation of new processes. A framework as outlined above allows visualisation and description of relevant social systems and illustrates relationships between the social systems and their direct and indirect impacts on each other. Furthermore, this special view of organizations as communication systems makes actors and their coordination fundamental enablers of and "changing parameters" for change. The systemic analysis of relationships within and between the depicted systems makes specific demands on the design of an organizational development project in an expert organization: namely to integrate different social worlds (expert groups), consider modes of participation and empowerment, and create commitment and coherence of targets.

It is therefore necessary to combine both approaches, the framework and the communication view: If only one of them is taken into account, either the communication view or the visualisation of the system(s) as a framework, then one would not succeed in handling the complexity – perceived as reduced complexity – (pure communication view) or the project would be conducted with inadequate methods and interventions (pure view of the system and its parts). It is the combination of the two approaches that makes a focused analysis of communication relationships possible and manageable(!). Thus a systemic integrated, coherent sub-project (such as the daily scheduling of physicians) can be derivated and conducted and, furthermore, certain processes as well as formulated targets become deployable in the form of boundary objects. The visualisation and coupling of social systems and their communication relationships creates the required, perceivable, but bridgeable boundaries. The example described above of the optimisation of the daily scheduling process of physicians is not regarded as a highly important goal within the larger system for increasing organizational performance and as a focus on organizational development until it is seen as part of the bigger picture of the whole system. Using the framework it may be conversely shown that an optimised scheduling process contributes to the goals of increased efficiency, transparency and customer orientation of the Austrian health care system, as well as to the strategy for improved allocation of resources by the host organizations. At the same time it also satisfies the local need for improved planning for personal as well as quality reasons. The systemic view firstly externalises the underlying relationships and secondly helps the person in charge of conducting a change project to determine where exactly to start. Thus this approach facilitates systems thinking and design of sustainable solutions to problems through graphical visualisation of patterns, communication of individual understanding and design of new patterns in order to intervene in exposed problematic system behaviour.

Boundary objects interlink worlds and therefore are vehicles for, as well as results of, participative coordination processes. They are both flexible enough to adapt to local needs and robust enough to maintain common identity across site. They may have different meanings in different worlds, but are recognisable in others, what is of crucial importance in expert organizations. The one thing necessary to make boundary objects work is to provide a medium to facilitate their entry into communication. S-BPM is a medium to facilitate coordination and communication, as well as the development of an organization as a whole, if embedded properly into the project design and the demands mentioned above are considered. It has been shown that targets for change as well as coherently derived business processes can take on the role of boundary objects and S-BPM can provide a medium for visualisation, modelling and discussion of boundary objects, thus participants establish a common understanding of these targets and business processes. At the same time, S-BPM makes relationships visible and brings them into awareness. Therefore, it is also an adequate medium for externalising analysed relations. S-BPM can help foster acceptance and coherence of organizational developments and change through the management of boundary objects, taking into account the necessary context and relating information across systems. Based on these insights, further analysis and evaluation is needed on whether the management of boundary objects fulfils its promise and results in a coherent, binding impact (e.g. through operable processes in the daily scheduling of physicians) and contributes to an increase in organizational performance.

## References


1. Sens, B.: Wie Pfade wirklich funktionieren. Integrierte Behandlungspfade als Instrument zur Umsetzung der Unternehmensstrategie. In: KU Gesundheitsmanagement, vol. 11, pp. 54–57 (2010)
2. Senge, P.M.: The Fifth Discipline: The Art & Practice of The Learning Organization. Crown Business, Edition 2006 (2006)
3. Health Consumer Powerhouse: Euro Health Consumer Index, Brüssel, Stockholm (2007)
4. Friesdorf, W., Marsolek, I.: Changemanagement im Krankenhaus. Im Mittelpunkt steht der Mensch. In: Behrendt, I., et al. (eds.) Zukunftsorientierter Wandel im Krankenhausmanagement, pp. 323–336. Springer, Heidelberg (2009)
5. Österreichisches Bundesministerium für Gesundheit: The Austrian Health Care System. Key Facts. Springer, Heidelberg (2010)
6. Österreichische Gemeinde-Zeitung: Kann die Gesundheitsreform ihre Ziele erreichen?, `http://www.staedtebund.gv.at/oegz/oegz-beitraege/jahresarchiv/details/artikel/kann-die-gesundheitsreform-ihre-ziele-erreichen.html` (DL September 25, 2011)
7. Schmidjell, R.: Dem Gesundheitssystem in Österreich fehlen Zielvorgaben (October 13, 2008), `http://portal.wko.at/wk/format_detail.wk?AngID=1&StID=429282&DstID=0` (DL September 25, 2011)
8. Bundesministerium für Gesundheit: Österreichischer Strukturplan Gesundheit – ÖSG 2010 (2010), `http://www.bmg.gv.at/cms/home/attachments/1/0/1/CH1071/CMS1136983382893/oesg_2010_-_gesamt,_stand_26.11.2010.pdf` (DL September 25, 2011)

9. Bundesministerium für Gesundheit: Regionale Strukturpläne Gesundheit (RSG) - Monitoring. Status of (July 16, 2011), http://bmg.gv.at/home/Schwerpunkte/Gesundheitssystem_Qualitaetssicherung/Planung/Ragionale_Strukturplaene_Gesundheit_RSG_Monitoring (DL September 25, 2011)
10. Bundesgesetzblatt für die Republik Österreich: 105. Vereinbarung gemäß Artikel 15a des Bundesverfassungsgesetzes über die Organisation und Finanzierung des Gesundheitswesens. Artikel 35 Maßnahmen zur Kostendämpfung und Effizienzsteigerung bzw. Steuerung (2008), http://bmg.gv.at/cms/home/attachments/2/7/1/CH1069/CMS1211801668459/bgbl_15a-vereinbarung_2008.pdf (DL September 26, 2011)
11. WKO (Wirtschaftskammer Österreich): Schelling fordert Paradigmenwechsel im Gesundheitssystem: Einheitliche Gesundheitsziele, mehr Qualitätskontrolle und Prävention (December 02, 2010), http://portal.wko.at/wk/format_detail.wk?angid=1&stid=585854&dstid=8713 (DL September 25, 2011)
12. Hoffmann, O.: Performance Management-Systeme und Implementierungsansätze, 2nd edn. Haupt Verlag (2000)
13. EFQM: The EFQM Framework for Managing External Resources. Driving performance and growth through excellence in managing external resources, p. 35 (2006), http://eipm-alumni.com/sites/default/files/EFQM%20EIPM%20Framework%20for%20Exc%20Ext%20Resources.pdf (DL October 10, 2011)
14. Kreis, R. (ed.): Wörterbuch zur Ethik des Rupert Lay. Lonneburger Texte, Büdingen (2002)
15. Krause, O.: Performance Management. Eine Stakeholder-Nutzen-orientierte und Geschäftsprozess-basierte Methode. Gabler (2006)
16. Luhman, N.: Soziologische Aufklärung 4: Beiträge zur funktionalen Differenzierung der Gesellschaft, 3rd edn. VS Verlag für Sozialwissenschaften (2005)
17. Simon, F.B.: Einführung in die systemische Organisationstheorie, 3rd edn. Carl-Auer-Systeme Verlag (2011)
18. Marsolek, I., Friesdorf, W.: Work Systems and Process Analysis in Health Care. In: Carayon, P. (ed.) Handbook of Human Factors and Ergonomics in Health Care and Patient Safety, pp. 649–662 (2006)
19. Simon, F.B.: Einführung in Systemtheorie und Konstruktivismus, 4th edn. Carl-Auer-Systeme Verlag (2009)
20. Reif, M.: Die Grenzen der Innovation. Oder: Wie viel Veränderung verträgt eine wertebasierte Führung? In: KU Gesundheitsmanagement, vol. 11, pp. 17–19 (2010)
21. Fonds Gesundes Österreich: Empowerment, http://www.fgoe.org/der-fonds/glossar/empowerment (DL October 11, 2011)
22. Reimer, M.: Verhaltensbedingte Hürden als Gegenstand des Changemanagement. In: Behrendt, I., et al. (eds.) Zukunftsorientierter Wandel im Krankenhausmanagement, pp. 337–356. Springer, Heidelberg (2009)
23. Star, S.L., Griesemer, J.R.: Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–1939. Social Studies of Science 19(3), 387–420 (1989)

24. Fischer, G.: Social Creativity: Turning Barriers into Opportunities for Collaborative Design. University of Colorado, Center for LifeLong Learning and Design (2004), http://l3d.cs.colorado.edu/~gerhard/papers/pd04-final-submit.pdf (DL October 11, 2011)
25. Heftberger, S., Stary, C.: Partizipatives organisationales Lernen. Ein prozessorientierter Ansatz. Deutscher Universitäts-Verlag/GWV Fachverlage GmbH (2004)
26. Fleischmann, A., et al.: Subjektorientiertes Prozessmanagement. Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern. Carl Hanser Verlag GmbH & CO., KG (2011)
27. Allee, V.: Value Network Analysis and Value Conversion of Tangible and Intangible Assets. Online version of Final Draft Published in Journal of Intellectual Capital 9(1), 5–24 (2008), http://vernaallee.com/images/VAA-VNAandValueConversionJIT.pdf (DL October 25, 2011)

# E-Learning Support for Business Process Modeling: Linking Modeling Language Concepts to General Modeling Concepts and Vice Versa

Matthias Neubauer

Johannes Kepler University Linz, Department of Business Information Systems – Communications Engineering, Freistaedterstaße 315, 4040 Linz, Austria
matthias.neubauer@jku.at

**Abstract.** Business process models have increasingly gained importance within today's rapidly changing whilst competitive business environments. Process modeling languages provide graphical means to depict, communicate and reflect real-world phenomena. With the advent of business process modeling a range of process modeling languages (such as BPMN, Event-driven Process Chains or S-BPM) as well as training programs for different process modeling languages have been developed. However, appropriate BPM education (cf. [14]) as well as the discussion "what and how to teach modeling" (cf. [8]) are still topics requiring further investigations and developments. In this contribution an e-learning approach towards the education of process modeling is presented based-on requirements derived from related work on educating modeling and e-learning. Furthermore, the developed learning approach will be illustrated within different usage scenarios.

**Keywords:** education, self-directed learning, process modeling.

## 1 Introduction

Models are an integral part of everyday life. They serve as fundamental basis for humans in order to depict, communicate and reflect real-world phenomena of certain domains. Doing so, models provide a basis for learning as well as improving and enhancing existing explanatory models [8]. With the advent of business process modeling a range of process modeling languages has developed spanning from simple flowcharting techniques, languages initially used as part of requirements engineering such as UML, dedicated business-oriented modeling languages such as Event-driven Process Chains | BPMN | S-BPMN, and also formalized and academically studied languages such as Petri nets (cf. [14]) .These modeling languages provide humans graphical constructs to articulate real-world domains for the purpose of understanding and communication (cf. [17])

However, even if more and more training programs teaching different process modeling languages have evolved (cf. [13]), appropriate BPM education (cf. [14]) as

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 62–76, 2012.

well as the discussion "what & how to teach modeling" (cf. [8]) are still topics that require further investigation.

This contribution aims to provide an e-learning approach supporting the education of process modeling competence. Doing so, requirements considering training business process modeling languages will be reconsidered in the first section. Secondly, requirements from e-learning will be discussed in order to provide the conceptual basis for the proposition of a flexible e-learning solution supporting the training of process modeling competence.

## 2 Educating Process Modeling

### 2.1 Requirements Considering the Education of BPM

Following requirements derived from reported experiences in educating modeling as well as BPM curricula development will be presented.

Glinz [8] discusses twelve theses about the Why, What, Where, When, How, and How Much of modeling in Informatics curricula is needed. Glinz identifies the ability to think in models, build models, reflect upon models and understand models as vital for coping complexity. According to Glinz [8] the following aspects are relevant in the context of modeling skill development:

- Students should understand modeling fundamentals and interdisciplinary modeling phenomena
- Students should understand which concepts of modeling approaches represent structural aspects, sequences, behavior, interaction and how they can apply them in different application context
- Empower students to apply modeling knowledge across disciplines
- Empower students to be able to read, understand and create models
- Help students understanding limitations of models and know when to apply which models or modeling approaches
- Knowledge considering modeling tools is not essential - important is conceptual knowledge on modeling. Even if modeling languages and tools embody modeling concepts (such as elements to depict structure, behavior,..) they (especially tools) are usually not as long lasting as the concepts incorporated in modeling languages/approaches. However, learning and understanding modeling concepts without concrete modeling languages is not reasonable because the application and training in concrete settings is not possible.

Besides Glinz [8], Recker and Rosemann [14] provide teaching experiences and recommendations especially from the field of business process modeling. They present a course design incorporating relevant aspects for developing business process modeling skills (compare figure 1).

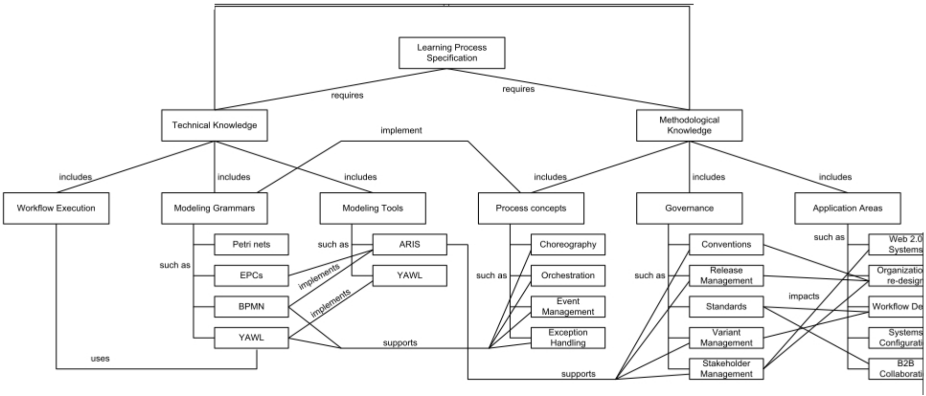



**Fig. 1.** Partial Concept Map for Course Content in Business Process Modeling [14]

Recker and Rosemann [14] divide process modeling knowledge into two strands, i.e. “methodological knowledge of process modeling” and technical knowledge of process modeling”. Methodological Knowledge includes “Process concepts” such as “Exception Handling”, “Event-Management”, “Choreography” or “Orchestration. Furthermore, methodological knowledge includes “Application Areas”, such as Organizational development, and knowledge considering “Governance” of process modeling projects that impacts certain application areas.

From a technical perspective, expertise in modeling processes with state of the Art modeling grammars such as BPMN (cf. [11]) as well as usage experience with respective process modeling tools is required. Knowledge about modeling grammars is interlinked with knowledge on process concepts since certain grammars support the representation of process concepts.

The curricula proposed by [14] incorporates the presented knowledge items in the following teaching process:

- Introduction to basic modeling principles (e.g. abstraction, generalization, association, reduction) and the process modeling discipline.
- Introduction to conceptual modeling (fundamentals of modeling such as method, tool, grammar, notation, governance, purpose, stakeholders are presented)
- Introduction & Comparison of process modeling grammars, e.g. BPMN, Event-driven Process Chains
- Process Architecture Design
- Process Model Governance including modeling governance mechanisms, conventions, variant, and release management
- State of the Art of process modeling and management tool suites
- Workflow execution
- Recent research trends

Within the given teaching process, content is delivered in different forms, i.e. formal lectures, practical workshops, and reflective and formative assessments in the form of weekly quizzes and assignments. In the workshops methodological and technical knowledge is applied to actual process modeling scenarios and cases in order to foster "theoretical" learning through practical application.

The given reports on educating modeling and business process modeling reveal:

- Importance of understanding fundamental modeling concepts
- Importance of understanding concrete modeling languages and their constructs as carrier of certain modeling concepts
- Importance of enabling the comparison of different modeling languages and approaches
- Importance of enabling relating existing (body of) knowledge on modeling with new (or not yet known) modeling approaches
- Importance of applying and reflecting modeling knowledge in certain domains as well as across certain domains
- Role of modeling tools for fostering theoretical knowledge through applying it

### 2.2 Requirements from e-learning

The e-learning paradigm has gained momentum within the last decades. Developments in e-learning design recognize learner control to be essential for self-directed learning processes (cf. [19]). Consequently, platform and content providers need to revisit their interaction facilities and information structures. Selective content consumption as well as a high degree of flexibility when intertwining navigation, presentation, content elements, and communication features seems to be crucial (cf. [7], [16]):

- Content elements should encode didactic quality. For instance, content elements should not only contain, but also visualize metadata, such as definition, for orientation and selection. Navigation facilities have to be adopted properly for exploring content – they have to support accessing information through different categories of content (in addition, e.g., to device adaptation for mobile learning).
- Different learners should be able to trigger and follow individual learning processes. This demand requires the possibility of dynamically selecting content elements according to the type of learner and his/her needs. Some categorization or decomposition mechanism as addressed above should facilitate the recognition and dynamic selection of content elements. User diversity also requires mechanisms to dynamically link discussion entries to content elements, not only in order to keep the transfer process context-sensitive, but rather to implement e-learning as inclusive learning community sharing views on content. Finally, each learner should be able to communicate with other learners as well as coaches.
- For navigation using different learner profiles brings only course- or module-relevant elements for navigation to the front-end device. The same holds for content elements. However, additional mechanisms, such as different levels of detail help to minimize the amount of content to be transferred and displayed, without losing its didactic quality. Another area of concern is the interactive experience with content, both with respect to annotations, and the individual

exploration of information spaces. E-learning environments should not restrict interactivity, rather they should provide various navigation and exploration mechanisms.
- Annotations, i.e. any form of content enrichments (links, comments, markings etc.) should be stored close to the concerned content.
- Fine-granular content elements, such as definitions, should be connected to communication facilities, such as chats, in order to enable the context-sensitive interaction with learn mates and coaches at any time.

This variety of requirements leads to novel approaches in the didactic design of content and its delivery to users. Content elements need not only be didactically relevant, but also flexibly accessible and polymorph when displayed at the user interface. Following, our concept developments for e-learning support are described on the basis of ongoing developments of the learning platform nymphaea (https://nymphaea.ce.jku.at).

## 3 Mapping Requirements to a Learning Environment

Our concept developments aim to put learners in control of the transfer process as well as to allow learners interacting in a context-sensitive way. Schulmeister [16] already lists factors to increase user-acceptance of virtual transfer environments. Influencing factors are for instance (i) the modularization of content, (ii) multimedia content and visualization of information to make these systems attractive, and (iii) the support of mobility and cooperation of users.

Studies on self-regulated learning reveal a variety of variables for positive transfer of knowledge, among them goal-oriented self reflection, an open environment in support of learning, self-instructive learning material, domain-specific integration of content, and multiple intervention based on general and domain-specific content (cf. [21]). When mapping these variables to an Internet-based transfer platform the authors argue for flexible content arrangement and open social spaces for intervention. Social processes should be context-sensitive which requires the binding of conversations to content. Finally, learners with different background and level of skills and experience do not only require features for communication and collaboration, but also the capability to develop individual views on the content (cf. [15]).

Hence, we have considered individualization support of content as a major objective of our developments. It is implemented through an annotation concept, providing textual notes, marking, and multimedia attachments directly in the courseware. Content and navigation are either adapted to learner knowledge (cf. [10]) or actively changed by learners including QoS parameters as claimed by [9], [3], and [20]. Features for individualization should also comprise the possibility for learners to learn mutually, hence sharing individual views or annotations, as suggested by Chang et al. [6].

In our approach, the annotation concept is considered as key for content and navigation individualization (cf. [1]), based on a hypermedia scheme for the flexible arrangement of content elements. It enables learners to (i) mark a specific position in a content element for learning, (ii) post questions, answers or comments, and (iii) additionally link the contribution to a discussion theme from the system's global discussion board. The latter link (being part of navigation) guides users to the adjacent discussion of the course material. In case of real-time online connections, e.g., chats, the questions and answers can pop up immediately on the displays of all connected users (available in a buddy list). In addition, the content elements referred to can be displayed at the same time. The presentation concept does not only support device-sensitive display of content, navigation and manipulation features, but also the decoupling of layout from content elements, thus allowing dedicated look and feel for particular content elements and interaction features.

Besides, we provide learners an associative navigation design that can be used complementary to the classical hierarchic navigation design. Following we will describe both the classical and associative navigation design as well as features provided for learning in the Nymphaea platform.

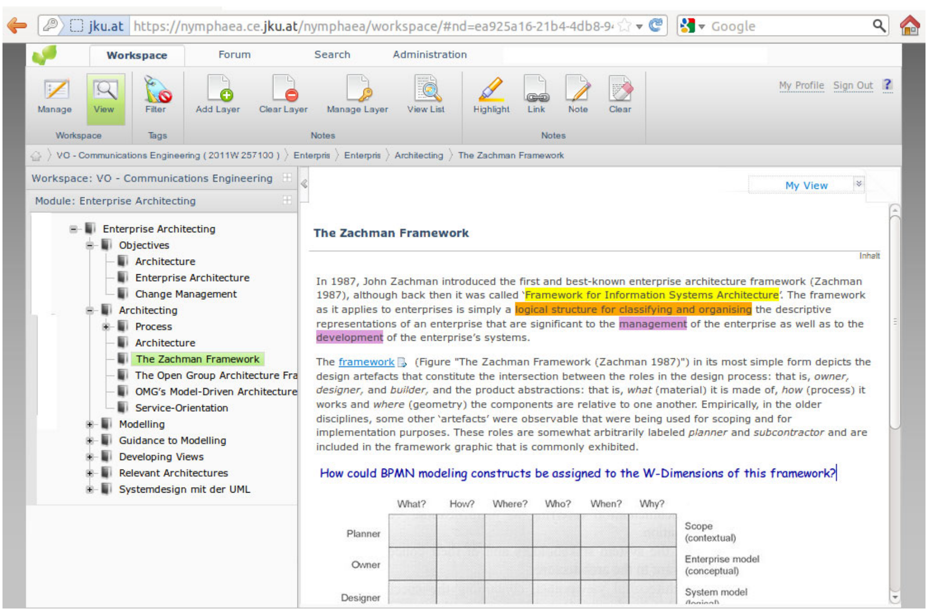



**Fig. 2.** Nymphaea learning platform – classical hierarchic view

When entering the learning environment users can choose different spaces, for instance the might enter the work space, communication space or the office. Within the work space relevant (learning) content is provided for users. In the overall work space a list of "workspaces", for instance for different courses a learner attends, is provided. Learning content within a certain workspace is comprised of modules and elements which are structured exclusively according to educational and domain-inherent metadata, such as ‘definition’, ‘motivation’, ‘background information’, ‘directive’, ‘example’, ‘self test’. They are displayed on the right side on top of each content element (figure 2) and can be used in the course of individualization when filtering content according to metadata. Content of workspaces can be navigated using the tree at the left-hand side.

Annotation support for content is realized using a view concept. As soon as provided content is displayed a view is generated like an overlay transparency. The view is kept for further access and reloaded when the content is accessed again. Within a certain view learners can (i) highlight, (ii) link, (iii) add remarks to content elements (compare “How could BPMN....” in figure 2). The features for the view management (add view layer, delete view layer, share view layer, show available views) as well as those for annotations are located in the ribbon-bar at top, whereas the selection of a certain view is provided at the right hand top of the content area (compare “MyView” in figure 2).

While annotating content learners can add internal and external references to content items. Internal references are links between content and communication items, such as entries in the discussion forum or Infoboard, which support context sensitive discussions. Furthermore, internal links might refer to other elements within the same or a different module. The corresponding features have been included into the annotation icon bar (see figure 1 and 3 – ‘Link’). Editing internal links requires marking a position in the text that should represent the link. After evoking the respective function located in the ribbon bar at the top a tree with the node of the currently addressed module is displayed. It allows users to select the target of the link (e.g. a forum entry or another content item).

Besides the traditional navigation design for the work space, we have developed a navigation design focusing on domain structures that can be used complementary. In order to support an integrated navigation of domain-inherent structures and arbitrary associations, we have developed a didactically augmented concept map solution to graphically organize and represent knowledge for learning.

Concept maps (cf. [12]) are established means to organize and represent information. They can be used to support the process of eliciting, structuring, and sharing knowledge. According to their objective, to enable meaningful learning (cf. [2]), most of the applications of concept maps can be found in education. In educational settings concept maps have been used in a variety of ways, for instance as scaffold for understanding, for consolidation of educational experience or as organizers for information [4]. Concept maps, besides other means to represent information, use concepts as entity to structure items of interest. Concepts might be central terms, expressions or metaphors, as they represent a unit of information for the person(s) using it. Those items are put into mutual context, leading to a network of concepts. Persons express the items of interest and the relationships by means of language constructs, i.e. per se there are no restrictions considering the naming of concepts or relationships.

Compared to the traditional design, the concept map navigation enables domain-specific and cross-boarder relationships. Learning paths can considerably differ when using the concept map approach. Instead of implicit learning paths – via hierarchies of modules, learning units, blocks or via internal/external links –, learning paths using a concept map are oriented towards explicit structure relationships beyond hierarchies and domains, as required for interdisciplinary content.

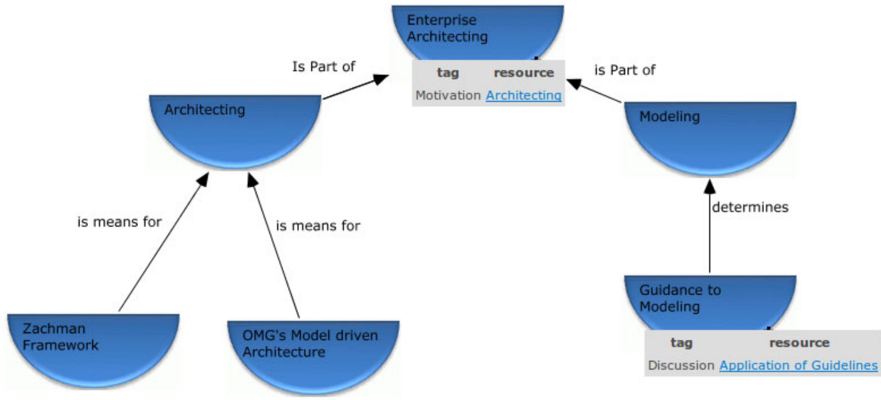



**Fig. 3.** Example for Cmap navigation and content links

Figure 3 depicts a part of a cross-disciplinary concept map for a learning unit on 'Enterprise Architecting'. It can be alternatively displayed for navigating learning contents.

Within the Map domain-specific associations are used for relating concepts. Furthermore, metadata (such as motivation, discussion, etc. - see figure 2) are used to semantically describe links from concepts to information resources (see figure 3). Hence, the associative navigation provides learners additional structural navigation information that shapes learning paths and should guide individual exploration of content.

Links from concepts to resources again can be internal or external. The example in Figure 3 shows that "Guidance to Modeling" is linked with a Discussion in a forum about the "Application of Guidelines". Additionally, "Enterprise Architecting" is linked to an internal resource describing the "Motivation" of "Enterprise Architecting". Available links for concepts are indicated by a '+' sign. If a user clicks or hovers over a concepts with a '+' sign available links are displayed.

Individualization support considering the associative navigation is similar to the hierarchic approach. It is enabled through features like

- Annotating a concept map and its elements
- Editing of a concept map - add individual concepts, relationships etc.
- Sharing individual views on concept maps
- Filtering links to information resources according to didactic content types, such as motivation or explanation

- Filtering links to information resources according to content codality, e.g., text, audio, video
- Individualizing navigation based on user profiles and preferences

Ensuring consistent individualization support for both, the hierarchic tree view and the concept map approach within the learn space, learners can not only choose a preferred navigation design, they can also adapt the respective navigation as well as the content according to their individual needs. Compared to the hierarchic approach, the concept map approach additionally enables to annotate and discuss navigation structures, i.e. concepts, relations, links to resources.

## 4 Applying the Learning Environment

In the previous sections requirements derived from experiences from educating modeling as well as from e-learning have been identified. Besides, features targeted to meet that requirements have been presented. In this section the usage of the presented e-learning features to support developing (S-BPM) modeling competence will be illustrated.

In order to support educating a modeling language respective content needs to be provided within the proposed learning environment. Learning content needs to be enriched with didactic information in order to support self-directed learning processes (cf. section 2 & 3). Auinger et al. [1] present a content authoring procedure that is supported by the introduced learning environment. The authoring procedure aims to provide learning material of high didactic value for a certain domain.

The application of this procedure for a concrete modeling language such as S-BPM includes following phases (cf. [1]):

- ***Preparation Phase:*** In a first step source material for (S-BPM) content development has to be identified and selected for further processing, for instance scripts, presentation slides, books, animations or scientific papers, modeling guidelines, tool manuals or modeling tools
- ***Initial Document Analysis:*** Available source material needs to be analyzed according to the level of granularity, (encoded) didactic principles and content-related orientation and navigation, e.g. Does S-BPM promote a certain didactic principle? What are relationships between provided contents? Results of this phase include: (i) rationale for each document; (ii) the conceptual relationships between the source elements; (iii) generic content types/elements/objects; (iv) ways for navigation or patterns of navigation. The results can be (re)presented using the concept map based navigation design presented in section 3 in order to support structured interviews with domain experts in the next phase.
- ***Structured Interview:*** After the first two phases the appropriation of the available materials needs to be considered. Interviews with coaches or teaching domain experts should be performed (with respect to the identified materials) in order to identify target groups that can be addressed, the learning culture where it might fit

into, its organization, the learning program it is part of, the resources needed, the demands it can meet, the requirements to implement it, and the knowledge as well as competences it addresses. Auinger et al [1] divide the interviews in *Organization*, *Individual Approach to Transfer*, *Knowledge Transfer, Communication* and *Technical support*. *Organization* addresses for instance structuring content, traditional learner profiles, and the organization of the learning environment whereas *Individual Approach to Transfer* aims to clarify the individual approach of content-providers or teaching authors to knowledge transfer. *Knowledge transfer* deals with organizational activities during knowledge transfer activities and the representation (e.g. hierarchic vs. associative navigation) of relevant learning material. *Communication* questions how the communication patterns among coaches and between coaches and learners in the context of knowledge transfer are revealed. *Technical support* addresses the clarification of current and future technical support of the knowledge transfer.

- In-depth ***document analysis*** and ***mark-up of content*** with didactic information. After identifying didactic elements and domain structures within the interviews available materials need to be structured accordingly. Using the presented e-learning environment, content is enhanced with additional metadata on the granularity (presentation slides | text | additional information) as well as didactic information on the type of content (e.g. assignment, example, definition, process). Hierarchical structures within content can be represented using the traditional tree navigation within the platform. In addition, domain structures can be represented using the associative navigation design based on concept maps.
- The actual ***content authoring*** and ***delivery*** to the proposed system.

Using the presented authoring procedure didactically enhanced learning materials for educating business process modeling languages can be developed. Depending on the *Individual Approach to Transfer* different educational strategies will be encoded in the learning material. An educational strategy could be providing conceptual modeling foundations before introducing, applying and comparing particular modeling approaches (cf. [8][13][18]). Following, a scenario will be presented that illustrates the application of the presented learning environment for a lecture on S-BPM according to the previously given educational strategy.

The author of the S-BPM learning material is able to use both, a hierarchic structure using the tree-view or a graph structure using the concept map approach, to encode the structure of the lecture in the learning platform. Following, a concept map including the main parts of a lecture on S-BPM is presented.

After starting the lecture with modeling foundations, the S-BPM modeling approach will be introduced. In order to foster theoretical inputs within the lecture a concrete modeling assignment for a “vacation application process” is given. The modeling assignment is depicted as concept map (cf. link to “Model vacation application” in figure 4) incorporating major activities proposed by the author when analyzing and modeling in a subject-oriented way. Such an assignment map could include following instructions with regard to features of the learning platform:

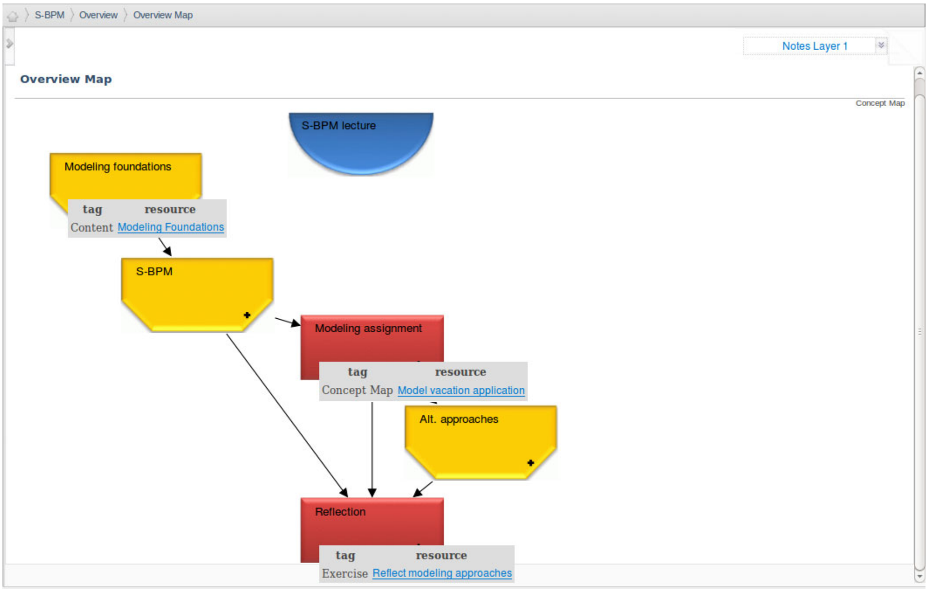



**Fig. 4.** Overview S-BPM lecture using a map

1. Read textual description of modeling scenario
2. Create own view for the analysis of the learning scenario
3. Highlight subjects, predicated and objects within the textual description using different colors and summarize them within the created view
4. Share and discuss your results with other participants
5. Model the scenario using a S-BPM tool
6. Validate scenario using a S-BPM tool
7. Discuss your modeling results within the platform

Using a map for describing assignments, allows authors to provide learners a graphical overview of the "steps to follow" as well as linking proposed activities directly to relevant content items (e.g. textual scenario description, S-BPM tool). When creating maps, authors can additionally use different concept types and assign them certain meanings, for instance in figure 4 concepts with regard to contents and concepts considering assignments are differentiated in terms of color and shape.

So far the provision of concept maps has been considered from a teacher's point of view in order to organize and present information for participants of a lecture. However, concept maps have been successfully applied in various other educational settings such as (i) identification of current understanding and misconceptions, (ii) assessment of learners or consolidation of educational experience (cf. [4]). For this reason the application of concept maps is considered relevant in the course of the reflection assignment. Following the concept mapping method, teachers should provide focus questions as impetus for the creation of "reflection maps", e.g.:

- What are the main concepts of S-BPM?
- Which language constructs of S-BPM depict which kind of aspects (Who/What/Why/Where/When/How)?
- How do the constructs of S-BPM relate to the conceptual modeling ontology of Wand & Weber for comparing/structuring modeling languages constructs?
- How do the language constructs of S-BPM map to general conceptual modeling constructs?
- What are differences between S-BPM, ARIS, BPMN?

On the basis of a certain focus question, learners will then create concept maps representing their current understanding. Doing so, they can explicitly depict relationships between modeling fundamentals and concrete modeling approaches as well as differences and similarities between concrete modeling languages. For example:

- "Subject—represents—>Actor" semantically links a fundamental modeling concept to a concrete modeling language constructs of S-BPM
- "Message Flow (BPMN)—similar to—Transition (S-BPM)" semantically links concepts of different modeling approaches such as S-BPM and BPMN
- "ARIS Business Architect—implements—>Event driven Process Chains" semantically links a modeling grammar and a modeling tool implementing the grammar

Using the annotation features of the platform learners can discuss their maps with others or even the teacher in order to consolidate their current understanding. Furthermore, the teacher is able to assess the understanding of a learner in terms of concepts provided within the created map and valid relationships between them. Within the given setting in figure 4 concept maps are used in terms of a summative assessment procedure. However, concepts maps could also be used during the whole learning process in terms of formative assessment. In formative assessment, the understanding of a learner will be represented as concept map at several stages within the lecture [4].

Assigning learners to author concept maps encourages them to reflect existing knowledge (cf. [5]). Hence, authoring of concept maps is considered a key feature addressing following requirements gathered from the education of business process modeling in section 2:

- Support understanding of fundamental modeling concepts
- Support understanding concrete modeling languages and their constructs
- Support relating modeling foundations and concrete modeling languages
- Support comparison of different modeling languages
- Support reflection of modeling knowledge

Besides, concept maps have been used to organize, present and navigate content in the given scenario. On the one hand they enable representing domain-specific, language-specific and cross-language concepts and relationships. On the other hand concept maps have been used to structure the learning process (compare figure 4). However, results from related work on concept maps for navigation (cf. [4]) indicate that learners should be encouraged to actively process information provided within maps and think in terms of meaningful relationships [4]. In order to encourage learners to rethink and discuss concept maps, teachers can post questions in the forum and link them to the respective maps or annotate the map and add remarks.

## 5 Conclusion

Modeling competence has been identified as an important asset for organizational members within today's rapidly changing whilst competitive business environments. However, educating modeling is still under investigation. This contribution revealed requirements from experiences in teaching modeling as well as requirements from the field of e-learning in order to propose an approach for e-learning support for process modeling competence.

The reports on educating modeling and business process modeling indicate the importance of (i) understanding fundamental modeling concepts, (ii) understanding concrete modeling languages and their constructs as carrier of certain modeling concepts, (iii) enabling the comparison of different modeling languages and approaches, (iv) enabling relating existing knowledge on modeling with new (or not yet known) modeling approaches, (v) applying and reflecting modeling knowledge in certain domains as well as across certain domains and (vi) the role of modeling tools for fostering theoretical knowledge through applying it. These requirements have been addressed within certain learning scenarios (section 4) supported by features of the presented learner-centered environment (section 3).

Learners can choose between an associative and a hierarchic navigation design when learning. The associative navigation design uses concept maps to depict relevant concepts and associations for a certain modeling language. Within the concrete S-BPM scenario concept maps have been used twofold, on the one hand as means for structuring content and depicting proposed content consumption sequences. On the other hand concept maps have been used as means for learners to reflect their knowledge and as basis for assessment of learners understanding.

When using concept maps for navigation, authors of learning materials are able to incorporate didactic and domain inherent metadata into concept maps. However, to foster learning when navigating with concept maps, learners need to be encouraged to actively process and think in terms of meaningful relations [4].

Future work will include the application of the presented learning environment in different lectures on process modeling languages in order to gain further empirical results regarding different educational designs as well as considering authoring and navigation support.

## References


1. Auinger, A., Fürlinger, S., Stary, C.: Interactive Annotations in Web-based Learning Environments. In: IEEE 4th International Conference on Advanced Learning Technologies, ICALT 2004 (2004)
2. Ausubel, D.P.: Educational Psychology, A Cognitive View. Holt, Rinehart and Winston, Inc., New York (1968)
3. Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. Künstliche Intelligenz 4/99, 19–25 (1999)
4. Cañas, A.J., et al.: A Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support. Technical Report from the Institute of Human and Machine Cognition (2003), http://cmap.ihmc.us/publications/ (December 15, 2011)
5. Cassata, A.E., French, L.: Using concept mapping to facilitate metacognitive control in precschool children. In: Cañas, A.J., Novak, J.D. (eds.) Concept Maps: Theory, Methodology, Technology, Proc. of the Second Int. Conference on Concept Mapping, San José, Costa Rica (2006)
6. Chang, S., Hassanein, E., Hsieh, S.-Y.: A Multimedia Micro-University. IEEE Multimedia, 60–68 (1998)
7. Deroin, R.E., Fritzsche, B., Salas, E.: E-Learning in Organizations. Journal of Management 31(6), 920–940 (2005)
8. Glinz, M.: Modellierung in der Lehre an Hochschulen: Thesen und Erfahrungen. Informatik-Spektrum 31(5), 425–434 (2008), doi:10.1007/s00287-008-0273-x
9. Harasim, L.: A Framework for Online Learning: The Virtual-U. IEEE Computer 32(9), 44–49 (1999)
10. Henze, N., Naceur, K., Nejdl, W., Wolpers, M.: Adaptive Hyperbooks for Constructivist Teaching. Künstliche Intelligenz 4/99, 26–31 (1999)
11. OMG: "Business Process Model and Notation, V 2.0" Object Management Group (2011), http://www.omg.org/spec/BPMN/2.0/
12. Novak, J., Canas, A.J.: The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools 2006-01. Florida Institute for Human and Machine Cognition (2006)
13. Recker, J., Dreiling, A.: Does It Matter Which Process Modelling Language We Teach or Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education. In: Toleman, M., Cater-Steel, A. (eds.) 18th Australasian Conference on Information Systems Toowoomba, Australia, December 5-7, pp. 356–366 (2007)
14. Recker, J., Rosemann, M.: Teaching Business Process Modelling: Experiences and Recommendations. Communications of the Association for Information Systems 25, Article 32 (2009), http://aisel.aisnet.org/cais/vol25/iss1/32 (December 15, 2011)
15. Röder, S.: Eine Architektur für individualisierbare computergestützte Lernumgebungen. In: Grundlagen, Modularisierung, und Prototypische Realisierung, Peter Lang, Frankfurt/Main (2003)
16. Schulmeister, R.: Grundlagen hypermedialer Lernsysteme. In: Theorie, Didaktik, Design. Addison-Wesley, Bonn (1996)
17. Siau, K.: Informational and Computational Equivalence in Comparing Information Modeling Methods. Journal of Database Management 15(1), 73–86 (2004)

18. Stary, C.: Evidence-Based (S-)BPM Education. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 3–15. Springer, Heidelberg (2011)
19. Totter, A., Stütz, D., Grote, G.: ICT and Schools: Identification of Factors Influencing the use of new Media in Vocational Training Schools. Electronic Journal of e-Learning 4(1), 96–103 (2006)
20. Vouk, M.A., Bitzer, D.L., Klevans, R.L.: Workflow and End-User Quality of Service. Issues in Web-Based Education. IEEE Transactions on Knowledge and Data Engineering 11(4), 673–687 (1999)
21. Ziegler, K., Hofmann, F., Astleitner, H.: Selbstreguliertes Lernen und Internet. In: Theoretische und empirische Grundlagen von Qualitätssicherungsmaßnahmen beim E-Learning, Peter Lang, Frankfurt/Main (2003)

# From Subject-Phase Model Based Process Specifications to an Executable Workflow

Albert Fleischmann

Metasonic AG
Münchner Straße 29, Hettenshausen, 85276 Pfaffenhofen, Germany
albert.fleischmann@metasonic.de

**Abstract.** In Business Process Management different stakeholders require different levels of abstractions of a process specififcation. Upper management is mainly interested in key perfomance indicators like speed, required ressources, customer complaints etc. Process owners who may be responsible for a process from end to end want to see the involved parties and want to have an overview of the major activities executed in a process. The people responsible for executing a process want to understand the details of the work they have to do in a process. Programmer also need to know the very details of a process to integrate existing application into a workflow system supporting process execution. In S-BPM there are up to now 5 Levels of process abstraction [2]: Goals of process and the related key perfomance indicators, Process architecture showing the process of a process system with their relationships, the active elements of a process called subjects together with the messages they exchange (Subject communication diagrsmm: SCD, the behaviour of each subject (Subject Behaviour Diagramm: SBD) and the implementation for the various subjects. In practical projects it has been shown that between process architecture and subject communication diagramm another view is required especially for process owners. Process owner want to see the involved subjekts with the major activities they execute. In this article a approach is described to specify processes from end to end. This approach allows to give an overview about the dynamic of a process on one page. This apporach has been used in several industrial projects and it has beeen well accepted by process owners as well as by people executing a process.

**Keywords:** process models, process execution.

## 1 Introduction

There are different stakeholders in business process management (see e.g. in [4], [8], [1]). The most important ones are the customers interested in the result of a process, managers responsible for processes (often called process owners) and the parties involved in the execution of a process (Providers). The parties involved in the execution of a process (providers) can be represented by subjects (see [3]. Subjects represent the active elements in a process which communicate

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 77–90, 2012.

with each other in order to coordinate their work producing the required result of a process. Especially middle management wants to get an overview about the process from the event causing the execution of a corresponding process instance (e.g. subject representing the customer of a process) till the results of a process are delivered to the parties expecting it. In the most abstract way upper management is only interested in Key Performane Indicators expressing the efficiency and effectiveness of a process. May be that the delivery of a process result causes the execution of a succeeding process. These succeeding processes use results of the proceeding process as input. Middle management is interested in who is involved in a process and what is their contribution to the process result and what are the cost for each contribution. Managers are not mainly interested in the details how the parties involved in a process execute their tasks, how they communicate with each other in order to coordinate their work and which tools and means they use to do their work. Whereas the parties involved in the execution of a process are exactly interested in these aspects of a process. The parties involved in the execution of a process want to know what work they have to do, in which sequences they have to execute these tasks, including when they have to communicate with whom about what and last but not least which means they use for doing their tasks. On the one hand processes have to be described precisely enough that the parties involved in a process execution know what they have to do in which situation and on the other hand management is primarily interested in more abstract process attributes like key performance indicators and the structure of a process, depending on the management level.

Additionally abstract views on processes are needed if complex process systems have to be defined and a top down approach for describing processes is applied. First designers create an overview of a process system and then this abstract model is refined step by step till a model is created which is good enough for the parties working in a process. In order to define hierarchically most methods and tools for process management use different approaches to solve that problem. In that article an approach is described which allows to get an overview of the involved parties of a process and what are their major contributions to the result of a process. Pracitcal exeprience show that this approach allows to describe the dynamic os a process on one page and the specification is precisely enough to derive executable workflows.

## 2 Related Work

Many approaches exist in order to express hierarchies of processes in business process management. In general three to five levels of process specification are used (e.g. see page 53 in [8], page 52 in [5], page 92 in [9]) These process levels are mainly called process areas, business processes, processes, subprocesses and activities. Sometimes sometimes business processes are classified in kernel processes, and main processes (see [9] page 87. In the following sections the abstraction concepts in the mostly used process modeling approaches are outlined.

In the ARIS (see [7], [10],[9]) ecosystem process chains are used in order to give an overview of a process. A process chain shows the process of a process system and defines the sequence in which these processes in the process system can be executed. There can be several levels of process chains. This means an element representing a process of a process chain can also contain a process chain and so on. In the lowest level each element of a process chain contains a Event driven Process Chain (EPC). The event driven process chain contains the activities executed in a process. On EPC level the activities are connected to subjects who have to execute an activity. This means only at the lowest level a manager can see who is involved in the execution of a process. It is possible that an activity in an EPC is not one single activity. It can again be an EPC. These various levels of abstraction are focused on the activities executed in a process.

BPMN has pools and swim lanes for describing process structures (see [6]). A process consists of one or several pools and each pool can consist of one or several swim lanes. It is not allowed to have pools in a pool or swim lanes in a swim lane. Activities are assigned to swim lanes. This means there are three abstraction layers: Pools, Swim Lanes and Activities. Pools and swim lanes are used to structure the activities in a process. An activity can contain a sequence of activities. This activity type is called subprocess. An activity in a subprocess can also be a subprocess and so on. This means on activity level arbitrary levels of subprocess are possible. In subprocesses pools and swim lanes are not allowed. Each activity is assigned to a party executing that activity. This means that a pool or swim lane does not correspond to a certain doer. Each activity in a swim lane can be executed by a different provider.

This two ways of abstraction are similar to the abstraction used in control flow charts. Only at the lowest level management can see who is involved in a process. The other way around the parties executing the actions have to scan through all the activity sequences in order to find the actions which they have to execute.

In this paper I want to show an approach in order to specify processes on different abstraction levels for the major stakeholders: Management and doer or provider. On one page managers and the actors in a process see the structure of a process (people and activities) and they get an overview who has to execute which action when.

## 3 Subject Phase Matrix (S-PM)

The features of the subject phase matrix will be demonstrated with an example from a real process management project in a small transport company. Figure 1 shows part of the process architecture of that company. There are three processes which built a process chain. There is the process 'Order'. In that process the order of customer will be checked and the transport will be prepared. In process 'Transport' the goods will be transported and in process 'invoicing' the message 'invoice' is send to the subject 'customer' and its payment is controlled.

The execution logic of each of these processes is described with a subject-phase matrics (S-PM). The following figure 2 shows the S-PM of the process order.

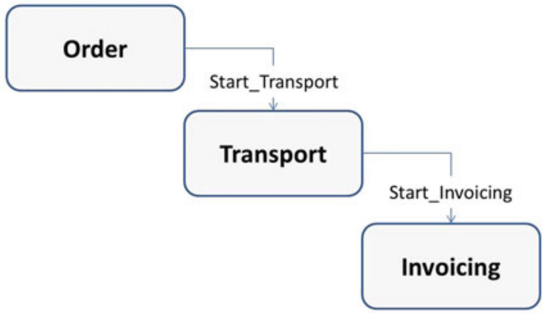



**Fig. 1.** Process-architecture

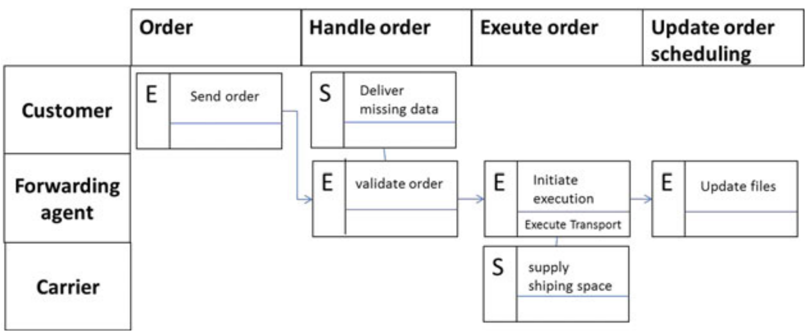



**Fig. 2.** S-PM of process 'Order'

A S-PM shows the subjects of a process in the most left column. Subjects represent the acting parties in a process. Subjects are abstract resources like in S-BPM (see [3]). The subjects in Figure 2 are 'customer', 'forwarding agent' and 'carrier'. An S-PM does not show the embedding of a process into an organization. This is a completely separate step as described in [3]. This means in that article only the process model is considered. The implementation activities are analog to S-BPM (as described in [3]). Process 'order' shown in 2 has the phases

- order
- handle order
- execute order
- update scheduling

A process is executed from the left to the right, but loops back to proceeding phases are allowed. This means a process starts with the most left phase. In the columns representing the various phases of a process the activities executed in that phase are specified. In a phase it is defined which subject executes which major activity set (marked by an E) and which subjects execute supporting activities (Marked by an S) for the executing subject. In figure 2 the subject 'Customer' executes the activity 'send order' in phase 'order'. In phase 'Handle order' the subject and 'Forwarding agent' are involved. The subject 'Forwarding agent' validates the order and the subject 'Customer' supports in that action. In order to get this support the subject 'Forwarding agent' communicates with the subject 'Customer'. In phase 'Execute order' the subject 'Forwarding agent' is supported by the carrier. In that phase subject 'Forwarding agent' starts the process 'Execute Transport' (Lower part of the rectangle representing the major activity in phase Execute order). If phase 'Execute Order' is finished subject 'Fprwarding Agent' executes in phase 'Update Order Schedulin' the activity 'Update Files'. Figure 3 shows the S-PM of process 'Execute Transport' initiated in process 'Order' in phase 'Execute Order' by subject 'Forwarding Agent'.

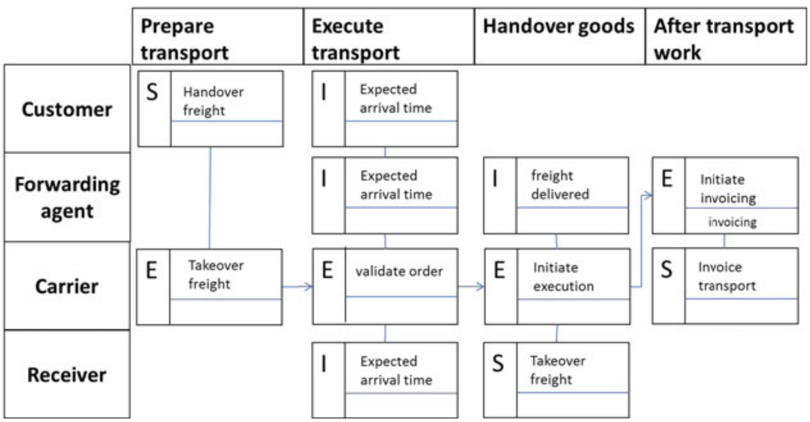



**Fig. 3.** S-PM of Process 'Execute Transport'

In process 'Execute Transport' some activities are marked with an I. This means that the corresponding subjects are only informed. In phase 'Execute process' subject 'Carrier' informs the subjects 'Customer', 'Forwarding agent' and 'Receiver' about the 'Expected arrival time'.

In the S-PM 'Execute Transport' there is also a subject named 'Customer'. This subject is different from the subject 'Customer' in the S-PM 'Order'. Subject names in a S-PM are valid in the corresponding process. This means subjects in different processes with the same name are different subjects. This does not mean that during the execution of a process (process instance) subjects with the same name in different processes are handled by different persons. Which providers (persons or machines) executes the activities of a subject are defined in the organisational embedding. This is not considered in that paper. This is a task of embedding subjects in an organization (see [3]).

S-PMs give an overview about the subjects involved in a process, which activities they execute in which sequence and which relations exists between the various subjects. In many real projects (ISO 9001 projects) more than hundred processes have been specified in that way. It showed up that each process can be structured in phases and processes have between three and six phases only a small number (around 2 percent) have more than six phases. In all cases S-PMs did have more than one page. There has been also the experience that S-PMs are easy to understand also for management and gives the involved parties a first impression what they have to do in a process. In spite of their overview character S-PMs are precise enough that a S-BPM specification can be automatically derived. This will be shown in the following section.

## 4 Conversion of Subject Phase Matrix to Subject Communication Models

The conversion of the Subject-Phase-Matrix into Subject Communication Diagrams (SCD) and Subject Behavior Diagrams (SBC) can be done automatically (see [3]). In general these SCDs and CBCs are executable without any additionally programming (see [3]) but the business objects must be added manually and some internal activities must be specified in more detail. In the following the focus is on the behavior of subjects. Data or business objects are not considered.

In order to transform an S-PM into a subject-communication specification we consider the matrix from the perspective of the involved subjects (see figure 3). Each subject in the matrix corresponds to a subject in the subject

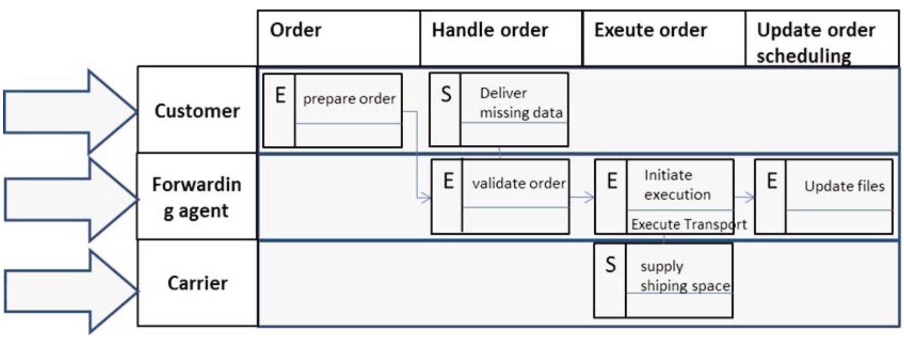



**Fig. 4.** S-PM considered from the subject view

communication diagram. The names of the subjects in the S-PM are extended with an Id for the process. In our case to each subject name the letters AO are added (AO=Accept Order). A subject sends a message to another subject if in the S-PM the E-action in the succeeding phase is executed by a different subject or the subject needs supports from an other subject. If subjects send a message to the subject executing the E-activity in the succeeding phase the message is named 'E-name-of the-succeeding-phase'. If a subject requests support from another subject the message is named 'S-Name-of-the-phase?'. The receiving subject sends the result of the support request back with a message called 'S-Name-of-the-phase!'. Figure 5 shows the communication structure derived from the S-PM shown in figure 2. Subject 'Customer-AO' is the start subject. It is

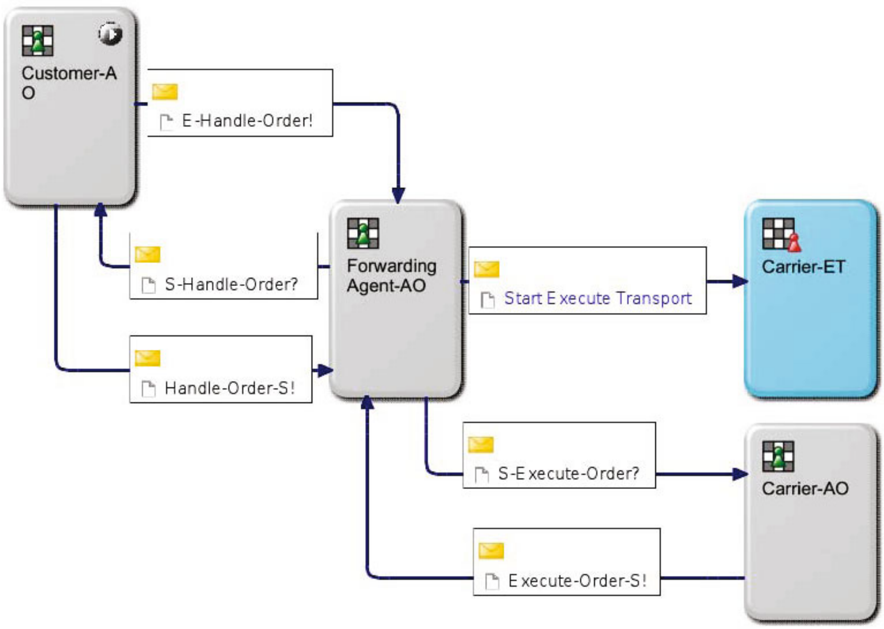



**Fig. 5.** Subject Communication Diagram of S-PM of Process 'Order'

the only subject which executes activities in the start phase of the S-PM Order. The succeeding phase 'Handle order' is executed by subject 'Forwarding Agent' therefore subject 'Customer' sends the message 'E-handle-Order' to subject 'Forwarding agent-AO'. This subject executes the activities in phase 'Handle Order' (see figure 2).

Figure 6 shows the behavior of subject 'Customer-AO'. In order to see the relationship between the phases in the S-PM and the activities in the SBD the activities in the SBD belonging to certain phase in the S-PM have frames marked with the phase name. After the activity 'prepare order' is finished the next phase is started by sending the message 'E-Handle-Order' to the subject 'Forwarding Agent-AO'. Than subject 'Customer-AO' is waiting for the message 'S-Handle-Order?' from subject 'ForwardingAgent-AO'. This is a support request which is sent by subject 'ForwardingAgent-AO' in phase 'Handle order'. The support activity is executed and the result is sent back to subject 'ForwardingAgent-AO'. The following figure 7 shows the more complicate behavior of subject 'ForwardingAgent'. The subject 'ForwardingAgent-AO' receives

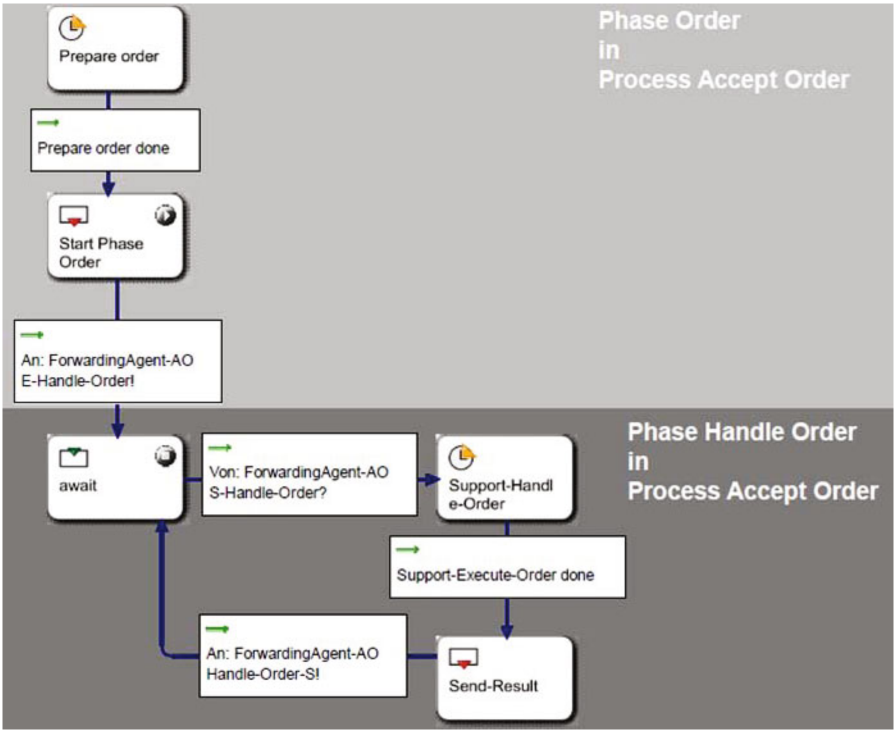



**Fig. 6.** SBD of Subject 'Customer-AO'

the message 'E-Handle-Order' from the subject 'Customer-AO' in phase "Handle Order" in the S-PM. After that message the subject 'ForwardingAgent-AO' checks whether it need some support from the subject 'Customer-AO'. If support is required a corresponding support request message ('S-Handle-Order?') is sent to the subject 'Customer-AO'. After an answer has been received the subject 'ForwardingAgent-AO' continues its work and checks whether some additional support is required or the activities in that phase can be finished. If that phase is finished the subject 'ForwardingAgent-AO' continues its work in phase 'Execute Order'. Because the subject 'ForwardingAgent-AO' is also the excuting subject for phase 'Execute order' it is not necessary to send an E.message to the executing subject of the succeeding phase. In phase 'execute-orde' the subject 'ForwardingAgent-AO' starts also the process 'Execute-Transport' by sending the message 'E-Start-Execute-Process' to the subject 'Carrier-ET' which is a subject in process 'Execute Transport'.

Figure 8 shows the behavior of subject 'Carrier-AO'. In process 'Accept Order' the subject 'Carrier-AO' only supports subject 'ForwardinAgent-AO' in phase 'Execute Order'. This means the subject 'Carrier-AO' receives a support request

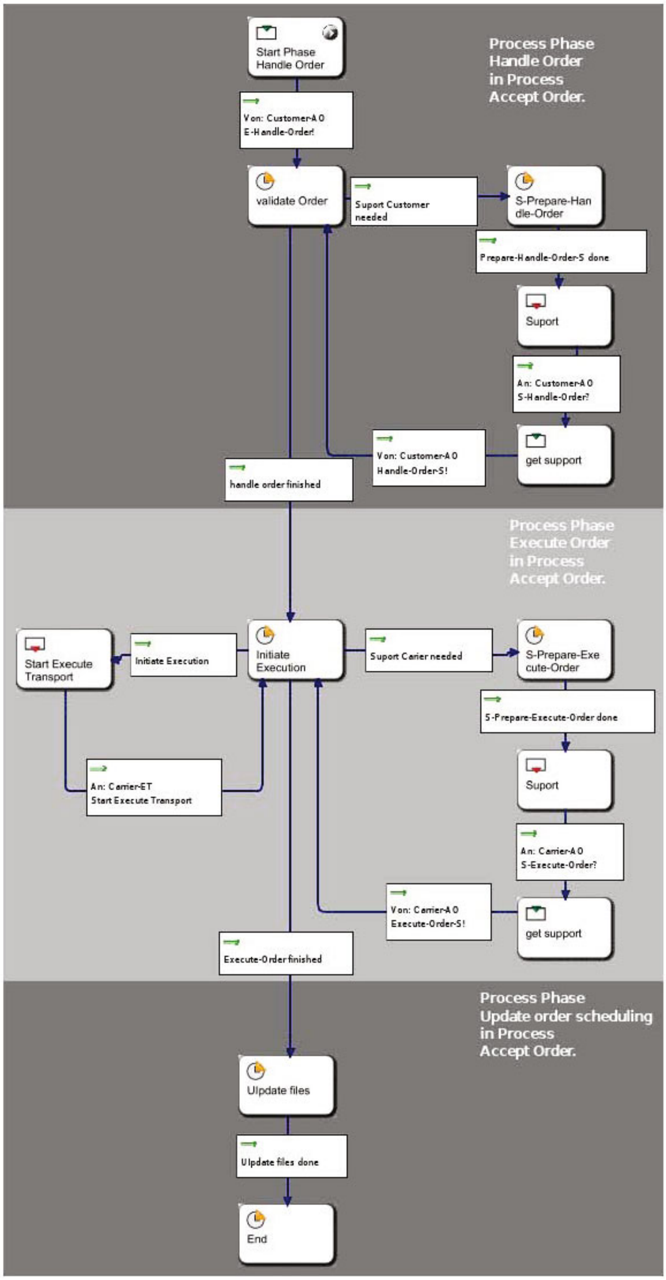



**Fig. 7.** SBD of subject 'Forwarding-Agent' in Process 'Accept Order'

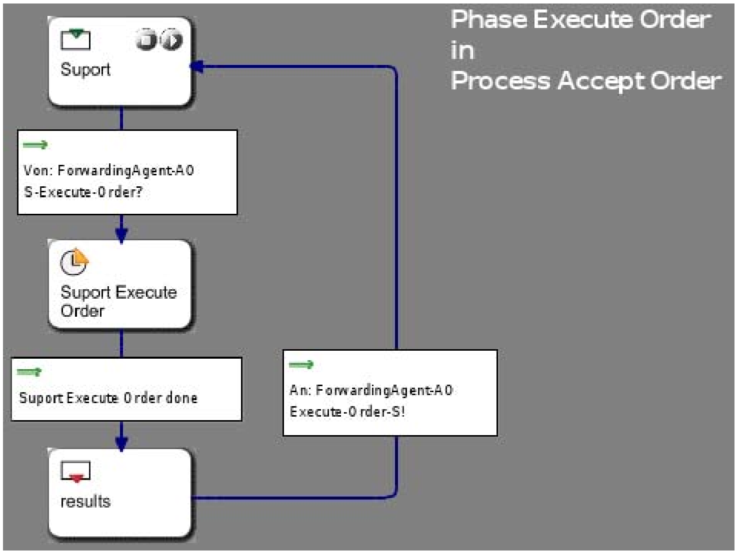



**Fig. 8.** SBD of subject 'Carrier-AO' in Process 'Accept Order'

message 'S-Execute-Order?', excutes the support activities and send the result back to the subject 'ForwardingAgent-AO' by the message 'S-Execute Order!'.

Figure 9 shows the SBD of process 'Execute Process' which is derived from the corresponding S-PM (see 3). The mechanism for getting the communication structure is the same as shown with process 'Accept Order'. The external subject 'ForwardingAgent-AO' represents the starting subject in process 'Accept-Order'.

Figure 10 shows the behavior of subject 'Carrier-ET' for the first two phases of the corresponding S-PM shown in Figure 10. This subject is different from the subject 'Carrier-AO' but it can be handled by the same person. But this is a decision during the embedding of a process into an organizational structure (see [3]).

## 5 Evaluation of the Transformation

S-PMs combine SCD and SBD. It show the subjects as active elements in a process, which subjects communicate with each other and which are the major phases of a process.

The phases can be seen as subprocesses producing some interim results. The author do not know a general proof that all processes can be structured in phases, but many widely used process specification methods use process phases (see section 'related work'). In his practical work the author hasn't yet found a process which could not structured in 3 to 6 phases, sometimes up to 8 phases (around two out of hundred).

The conversion of each process phase into SCD and SBD is based on modeling by restriction (see chapter 6 in [3]). Modeling by restriction is executed in 5 steps:

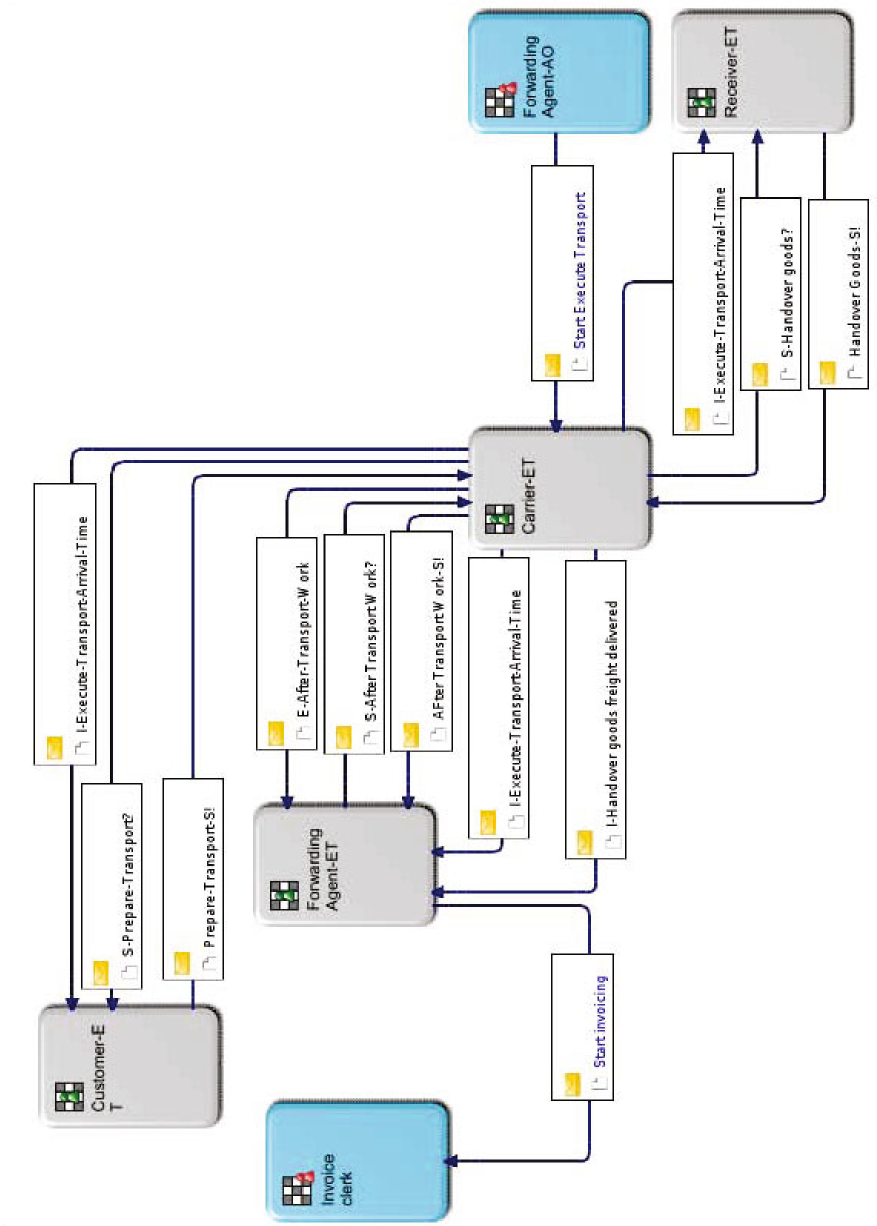



**Fig. 9.** SCD of Process 'Execute Transport'

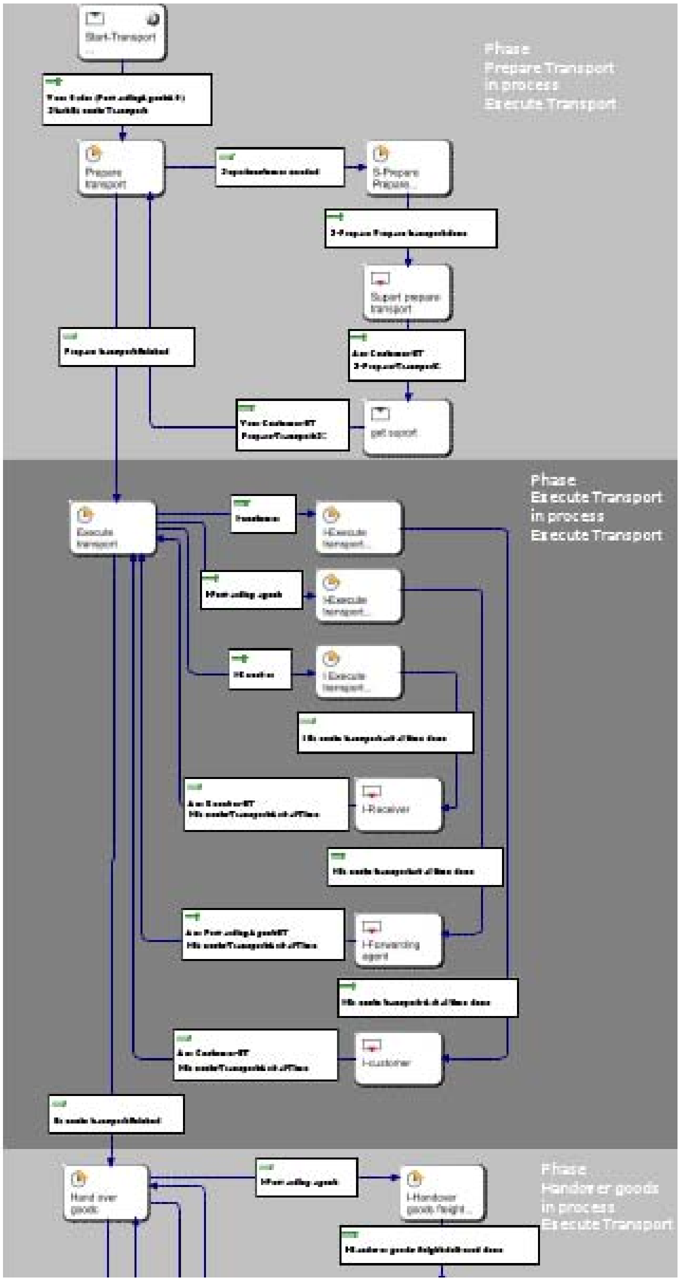



**Fig. 10.** SBD of Subject 'Carrier-ET' in Process 'Execute Transport'

– Identify the number of subjects involved in a process,
– give these subjects names which describe their task in the process,
– remove not allowed communication paths between subjects,
– introduce process specific message names
– adapt communication behaviour to process requirements

An S-PM contains the information to execute the first 4 steps automatically for each process phase.In a phase the E-Subject communicates with the S- and I-subjects in a not very strict way. This means an E-Subject can request a service from a S-subject as often it wants, which may be not correct in the sense of the considered process. Finally an E-Subject decides whether a succeeding phase is started. This means a S-PM is converted in a process consisting of a chain of not very strictly defined subprocesses derived by modeling by restriction from the S-PM.

## 6 Activity and Data Details in S-PM

In S-PM mainly the sequencees of the execution of the major activities are considered. In order to describe data and the details of the activities executed in a process phase a so called Input Execution Output table (IEO-Table) is used. The following figure 11 shows the IEO-table for the S-PM of the process 'Accept Order'.

| | Order | Handle order | Execute order | Update scheduling |
|---|---|---|---|---|
| **Input** | Required transport capacity | Order | Complete order | Transport in execution |
| **Activity** | Prepare order | Key in order; Agree on terms and conditions; | Procure transport capacity; Initiate transport execution | Update transport schedule |
| **Output** | order | Complete order | Transport in execution | Updated transport schedule |

**Fig. 11.** Input Execution Output Matrix of Process 'Accept Order'

The columns represent the phases of a process as in the corresponding S-PM. In the input row the data required in a process phase are specified, in the activity row the activities executed in a phase are listed and the output row contains the results of a process phase. The activities listed in the activity row define the details of the internal activity called 'phase name' or 'S-phase name' in a SBD derived from a S-PM. Based on that information the SBD can be described in more detail. The input and output row can be used to define the business objects used by the various subjects and transported by the different messages. In order to develop a systematic way to hand over that information to more detailed SCDs and SBD some additional work has to be done and experience has to be gathered. Up to now only a very intuitive approach is used.

## 7 Conclusion

S-PMs give managers and subjects involved in process an overview of a process provided that a process can be separated in phases. Practical experiences over more than 10 years show that this is possible without any difficulties. S-PMs are precisely defined which allows to convert them in SCDs and SBDs automatically. This conversion is based on modeling by restriction. This automatic conversion result in a first version of SCDs and SBDs which must be more restricted by the requirements of the considered process and enriched with the required business objects. Details for a S-PM are described in so called IEO-tables which contain details about the required data, the executed activities and the results of a process phase. In further work it has to be investigated how this infomation can be used in automatic conversions.

## References

1. vom Brocke, J., Rosemann, M. (eds.): Handbook on Business Process Management, vol. 2. Springer, Heidelberg (2010)
2. Fischer, H., Fleischmann, A., Obermeier, S.: Geschaeftsprozesse realisieren. Vieweg Verlag (2006)
3. Fleischmann, A., et al.: Subjektorientiertes Prozessmanagement. Hanser Verlag (2011)
4. Fleischmann, A., Stary, C.: Whom to talk to? a stakeholder perspective on business process management. Universal Access in the Information Society (2011)
5. Horvath, Partner (eds.): Prozessmanagement umsetzen. Schäfer, Pöschel (2005)
6. OMG: Business process model and notation (bpmn). Tech. rep. (last access December 2011)
7. Scheer, A.W.: ARIS-Vom Geschäftsprozess zum Anwendungssystem, 4th edn. Springer, Heidelberg (2001)
8. Sharp, A., McDermott, P.: Workflow Modeling. Artech House (2009)
9. Slama, D., Nelius, R.: Enterprise BPM. dpunkt Verlag (2011)
10. Snabe, J., Rosenberg, A., Miller, C., Scavillo, M.: Business Process Management, The SAP Roadmap, 4th edn. Springer, Heidelberg (2001)

# ModelAsYouGo: (Re-) Design of S-BPM Process Models during Execution Time

Robert Gottanka and Nils Meyer

Metasonic AG
Münchner Str. 29 - Hettershausen, 85276 Pfaffenhofen, Germany
{robert.gottanka,nils.meyer}@metasonic.de

**Abstract.** The classic approach of a Business Process Life Cycle, where a process model will be defined, modeled, simulated, deployed and then executed to get monitored, analyzed and finally optimized to start the cycle again, often does not match the needs of new dynamic requirements on Business Process Management Systems (BPMS). Practical use cases for a more dynamic BPM are emergent processes, which occur in situations, where it is necessary to react at execution time on business transactions, e.g. Adaptive Case Management (ACM) and flexible processes. Even S-BPM, an approach which enables BPM especially in processes mostly controlled by human interaction, is working with a static process model that is executable, but not changeable during execution. ModelAsYouGo shows a way to (re-) design a S-BPM process model, whose workflow is not or just partly known at modelling time, while actually executing the process. By exploiting S-BPM's focus on interaction ModelAsYouGo allows process participants to design their process models in a collaborative manner.

**Keywords:** subject-oriented business process management (S-BPM), design by doing, emergent processes, ad-hoc, adaptive case management (ACM), social BPM, collaboration.

## 1 Introduction

"Model as you go" (ModelAsYouGo) is a new approach to model a S-BPM business process by enabling the process actors to record their subject communication and internal behavior, just in time while they execute the process instance. It enables runtime collaboration and dynamic modeling, which are often considered to be the same capabilities in a BPM system; although they are highly related, they are not identical: runtime collaboration is the activity of adding participants to a process instance during runtime who were not part of the original process design and also the activity of additional communication with already existing participants. Dynamic modeling is the activity of modifying the model for a process instance, usually to add one or more new tasks to the process [27]. To introduce the ModelAsYouGo approach, this section explains first an S-BPM project methodology. It shows why there is a need for a modeling method, which enables the process actors to record their communication and behavior in an S-BPM process to keep it available for later use.

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 91–105, 2012.

A project, where a business process will be improved via a Business Process Management System (BPMS) based on an S-BPM methodology, usually starts with its analysis phase. In this phase of the project it is necessary to establish four different roles to handle the project: the facilitators, the governors, the actors and the experts [1]. The governors hold the responsibilities for the project framework with internal and external requirements. The facilitators take up the role of a moderator in the business process project; they support the process actors with the project organization. Typical examples for facilitators are project manager, organization developer, coaches or service desk staff. The process actors execute the business process (the instantiation of the process model) and also should be involved to develop these process models. They are the persons acting in the process and are center of reference in equal measure. After all, the experts support the actors in professional questions, where special knowledge is required to proceed. Each of these four roles has different questions about the process but all together participate on completion of the process image. They define and appropriate the reference points of the project management model, they describe the business process in natural language by identifying subjects, activities and business objects, and finally they model the S-BPM process model with its communication view, the subjects and their messages, and the corresponding internal behaviors. The participants are able to review and decide how a process model should work. At the end, there is an S-BPM process model (diagram) and this process diagram is declared as “complete”.

At this point, a common conflict of the BPM paradigm has its origin. There are many situations, where a process cannot be modeled before its actual execution. This means some process models cannot be considered complete before they actually happened in the real world. The term business process refers here to the spectrum from production workflows to emergent processes [2]. In the latter case the business processes instances are executed in a world with less structure, incomplete information, and unforeseen exceptions, so therefore no BPM process model is really complete. Business Process Reengineering [3] and also similar Total Quality Management [4] (with their simplistic iterative steps scenario design, process execution and analysis continue with the subsequent re-design to begin the next iteration) achieve in the case of unstructured processes their constraints very soon. Also the approach of a top-down BPM Cycle [5] does not match these requirements.

To model a process and its flow first – and execute it after that modeling has many benefits, but not every business process has the potential to be modeled – especially processes where the execution of the process changes on a case-by-case basis. Basic attributes of processes that cannot be modeled are: by strong human interaction driven processes like collaboration and negotiation; content, that is both consumed and produced as part of the process, the participants change the flow, the participant itself changes, an activity changes or every process has an owner[6].

## 2 Structuredness and Ongoing Developments in BPM

Structuredness (that means the state or condition of being structured) of a business process is given, when the way to reach the end state / output is well defined. In this case the whole process is completely and explicitly defined by the follow of its activities. The degree of structuredness increases, depending on the determination of its flow to reach the output on a dictated path. Also the degree of automation can expand with an increasing process structure. The smaller the structuredness of a process is, the more time is necessary to complete it by trend, because the analytical and creative requirements on the executor in process handling increase [25].

BPM is a discipline for continuously improving cross-functional, end-to-end business processes. To accomplish this, business process professionals spend gobs of time and many analyzing and implementing strategies to improve process collaboration, communication, interdepartmental hand-offs, and institutional knowledge. All those activities are basically social, which is just like the natural extension and evolution of collaboration – a fundamental future trend in BPM [26].

This section provides the background information for the ModelAsYouGo approach in this paper. This section first classifies business processes by their structure, which is necessary to explain, for which kind of business processes ModelAsYouGo is applicable. The second part of this section is a summary of current developments in Business Process Management.

### 2.1 Structured, Semi-structured and Unstructured Business Processes

Business process models are differentiated by their structure; structured, semi-structured and unstructured business processes. Structure in an S-BPM context means that the process model is recognizable, observable, has a fixed communication and subject internal behavior. Before the introduction of the use cases it is necessary to classify business processes, to make them comparable.

There are structured business processes that are often core processes, which can be classified alongside the supply chain of a company. Their process model description is strictly pre-defined and has an end-to-end model that includes all process instance permutations. There is no process instance which differs from that model. Structured business processes are usually supported by Workflow-Management-Systems (WFMS) [9] or BPMS or exist directly in a system like SAP.

Semi-structured processes are business or scientific processes whose lifecycle is not completely defined by a formal process model. Often, there is just an incomplete process overview available, where the process is available in the form of a process graph, flow chart or an abstract state diagram, but the execution is not completely controlled by a central entity (such as a workflow or BPMS engine), if at all. Instead, a variety of IT and human centric mechanisms are used, including email, content management systems, web-based forms, custom applications or a combination thereof [10]. Even in an S-BPM environment there are often partial sections, where internal behavior or communication is not modeled, e.g. function states with more behavior on it than described or parts of processes that are not discovered, yet.

An unstructured process is a process that is not predictable. Unstructured processes have a sensitive dependence upon external factors outside of the control of the process context, which is why they cannot be fixed according to their internal state. Synonyms include ad hoc process and emergent process [8].

### 2.2 Collaboration, Social Software and (S-) BPM

There are changes in BPM that are stirring the business process community with approaches like Collaborative BPM, Social BPM, DesignByDoing or Dynamic BPM. All these topics have intersection areas or need each other to exist. This section elaborates, why they are import parts in the puzzle of a complete ModelAsYouGo picture.

A significant change of thinking is observable. BPM is used in organizations due to the increase in collaboration. Process workers should not only access the process tasks and data, they should collaborate with each other via discussions, wikis, documents, etc. A benefit is the complete visibility of the collaboration, provided to all stakeholders. Decisions and actions could be traced to the ad-hoc collaboration and the collaboration can be archived along with the process instance.

Other scenarios are inherently collaborative. End users start working outside of the process, collaborating with each other via email and exchanging information through documents. In effect a shadow process exists outside of the process and the main process has no information about the shadow process, which sometimes has context on the more important audit-worthy decisions. This limits the value BPM Suites can deliver for these types of processes. One of the most compelling promises that social BPM holds is the ability to enable such ad-hoc and collaborative processes, bringing together the structured and unstructured and completing the realm of BPM effectiveness [21].

Collaborative process modeling enables people, technical and nontechnical, to participate in the discovery, modeling, design, implementation, and optimization of a business process. Collaboration during the execution of a an existing structured process in a BPMS allows the participant at any process state to leave the pre-modeled structured process and initiate an ad hoc collaboration with users of their choice in order to accomplish the task at hand. The BPMS allows the gathering of information on how the process instances were executed, allowing them to be considered for future standardization and modeling as structured processes. To provide a dynamic BPMS environment, which can include ad hoc and collaboration scenarios in the context of a more structured business process, allows the participants to use their own best practices and tools, particularly in processes that rely heavily on subjective human knowledge [23].

Social software and Business Process Improvement (BPI) have different focus areas and uses. BPI has mainly focused on structure and efficiency, while social software tends to support more-free-form or unstructured activities. By looking at the intersection of these two areas from technology and discipline perspectives, the emergence of social BPM can redirect the value from both of these views to drive more-effective process performance that meets the needs of not only people doing the work, but also those "experiencing" the process. "Social BPM" extends the traditional BPM and is an approach that describes collaboratively designed and iterated processes. These collaborative processes are driven from the perspective of a “doer” and experienced from a “receiver" perspective to harness the power of continuous learning from "the collective". Social software and BPM are digging out synergy and can work together to enable "design by doing." Social BPM induces two phases - design and iteration. The real value comes from using social BPM to influence existing processes, rather than limiting it to the social aspects of design [24].

Social BPM will leverage social networking tool and techniques to extend the reach and impact of process improvement efforts. Two key areas here are the collaborative process modeling and the collaboration during the execution of a process. So many people from a variety of perspectives – including end users, business analysts and IT – are involved in modeling processes. At runtime collaboration, processes are modified during execution dynamically to include unplanned participants in order to complete the work more effectively [20]. Social will change the way we get work - and processes - done. While it may seem an odd combination, Forrester now sees more BPM teams leveraging social and Web 2.0 components - wikis, mashups, team workspaces - to improve collaboration and engagement throughout process discovery, design, and development [22].

By looking at the intersection of social software and BPM, ModelAsYouGo should redirect the value from both of these views. The concept of DesignByDoing should be implemented within the ModelAsYouGo system. The participant is doing the process instance and modeling after each process step the next step. S-BPM, where the communication and behavior is modeled in the view of the participants, provides the basis for ModelAsYouGo. Where S-BPM works with communication, in ModelAsYouGo it is necessary to work with collaboration. In this concept each participant is just able to model its own internal behavior, to send or wait for messages. If there is more than one participant to the business process involved, collaboration is taking place.

On the one hand there is a need for a collaborative process design, on the other side there is a need for collaboration during the execution of the process, to be able to model unstructured and semi-structured business processes. The ModelAsYouGo concept provides the ability to a collaborative modeling of S-BPM process descriptions during execution time. The ability to create, store and edit S-BPM process descriptions, enables adaptive approaches. The network effect of multiple authors can increase productivity and generate innovative, emergent ideas.

Last but not least is ModelAsYouGo a trial to enable business users to create process models in an easy way. There should be no symbol overload with long taking training. The user is able to model internal function states and also to communicate. Simple text input is necessary to set the subject and the receiver of the message before sending the message.

## 3 Use Cases for the ModelAsYouGo Approach

In S-BPM all processes are a composition of a grammar involving the interplay of subjects, objects and predicates. These elements allow to answer the “who”, “what” and “with what” questions. The subject, thus the participant in a business process instance, is the central focus of S-BPM. The design of a process model and the execution therefore is a grammatical effort accomplished when a subject and another subject communicate [18].

“As organizations continue to embrace BPM to improve business performance during challenging times, this quest is pushing BPM beyond its traditional focus on routine, predictable, sequential processes towards broader, cross-boundary processes that include more unstructured work. Knowledge work is especially complex and unstructured” Janelle Hill, research vice president at Gartner, explained at the Gartner Business Process Management Summit, London 2011. “New BPM technologies will enable the management of unstructured and dynamic processes to deliver greater knowledge worker productivity and competitive advantage.” [7]

There are strong business requirements to manage these unstructured and dynamic processes. ModelAsYouGo is an approach to fulfill these business needs as it tries to enable the modeling of S-BPM process descriptions directly by the actors “as they go”. This section introduces the four use cases, which will be later reviewed. To enable the execution of those use cases in a subject-oriented BPMS, a dynamic modeling of the process models in unpredictable situations is necessary.

### 3.1 Ad Hoc Process – Instantly Created Activities by the Business User

Ad hoc generally signifies a solution designed for a specific problem or task, non-generalizable, and not intended to be able to be adapted to other purposes. Participants involved in an ad hoc process (unstructured process) will experience it as planning and working alternately or at the same time, such that the plan is emerging as the work is proceeding. In many business cases there are situations, where a knowledge worker has to improvise to get the work done [8].

On one side there are new process instances which have never happened in that manner before. So there is no process model existing and the process participant has to start from scratch. There is an open path and the participant has to find its way.

On the other side there are situations, where changes on an existing process model are necessary, when the external context of the process is changing. In this dynamic process scenario the activity sequence should be changeable during runtime. The process instance follows the defined path in the work flow till a certain point, where it has to leave the predefined way. Here is an exit point from the path in the model, where the path steers in a new direction.

In an S-BPM context there is a subject, which wants to start a process with the support of a BPMS, to avoid intransparent email and excel processes. The system is based on the formal model of S-BPM, which enables the business user to give the underlying communication constraints. S-BPM architecture enables the user to instantiate predefined process models, but does not enable to start a process instance from scratch without an existing process description. There should be an agile approach to move quickly and without much preparation and support.

### 3.2 Adaptive Case Management (ACM)

Knowledge workers in today's workforce are individuals who are valued for their ability to act and communicate with knowledge within a specific subject area. They will often advance the overall understanding of that subject through focused analysis, design and/or development. They use research skills to define problems and to identify alternatives [12].

Case management is a process use case that differs from other structured, predefined process use cases. Case management departs from the traditional view of structured and sequential predefined processes. Instead workflows are nondeterministic, meaning they have one or more points where different continuations are possible. They are driven more by human decision making and content status than other factors. Its traits impose unique requirements on support when delivered through business process management suites [11].

Adaptive Case Management (ACM) expands the Case Management by an adaptive component, which supports the knowledge worker to learn, to store and to provide adapted business process case information. Adaptive Case Management fills a gap in the spectrum from predictable to unpredictable processes. It has no fixed process, is goal oriented and collaborative. Adaptive knowledge work is unpredictable in execution and decisions. It is defined by goals and rules, rather than steps and gateways. The planning is part of execution; the participants are added as necessary, not fixed. ACM demands secure, Web 2.0 like collaboration, transparency and auditability [6]. BPM and ACM both involve process and both involve information, but the way is different. In the BPMS, the process is the central theme, which accesses data from various places and sends data to various destinations, but the process remains the persistent aspect. In the ACM system, the case data is known first, and it remains persistent along with information that is collected over the life of the case. In an ACM system, processes are brought into the case to handle work as needed [15].

There already exist approaches to combine BPM an ACM in a system. The case process is the central steering instrument within the case instance. In contrast to classic (subject-oriented) BPMS, this process is not fixed in advance, but will be developed during the case execution [13]. There is a case folder which stores the business information and also the processes. Already known process flow is stored in a template library and is available for use in the case. In conclusion in S-BPM there is a need for an adaptive component, which allows adapting process paths, using this control flow information and storing it after that in the template library for later use. The adapted processes have to be placed around the data and drive the case forward. ModelAsYouGo could help to adapt not yet identified process behavior. Like in ad hoc processes there is a requirement to expand existing process model or react on non-predictable exceptions. An essential part is to enable the participants to collaborate when they model the communication of the subjects.

### 3.3 Flexible Process

Unplanned events and exceptions often head to variations from the preplanned business processes. Exceptions cover cases such as requests to deviate from standard processes, failed tasks incomplete or erroneous information in task inputs and outputs or situations that arise from mismatches between the real processes within the organization and their computerized counterparts. Since business process modelers are not generally capable to predict all possible exceptions and events beforehand and to capture them in the design of a business process model, the BPMS does not always have sufficient knowledge to handle these situations alone. Instead, user involvement is required in order to resolve exceptions and to deal with unplanned events [16].

A flexible process is not stiff. It can be bent during execution but returns back in its origin position, like an elastic strap. The activities in flexible processes can be modified by an actor. Despite to ad hoc processes flexible processes return returns back to the predefined path, after its work around the exception. An approach to allow the instance owner to change the workflow immediately could help to enable this flexible process handling. In S-BPM environment it occurs, that an instance is "hanging" in a state, because a participant is waiting for a final message or the participant wants to send a message, even he has not received the confirmation yet, because someone forgot to send the answer. There are many use cases for exception that in real business occur, when a BPMS will be established. ModelAsYouGo could be a workaround to act just in time on those exceptions and recue the process instance.

### 3.4 Process Template

There are business processes where it not worth to model the full business process, because it is clear from the beginning on, that it is too expensive, to model the whole process with all its cases and possible paths. The 80-20 rule (also known as pareto principle) says that, for many events, roughly 80% of the effects come from 20% of the causes [19]. Even in BPM projects, it is the case, that 80% of the benefit comes from 20% of the efforts. So it should be more cost efficient to model a process template, which represents these 20% of a business process that catches the bulk.

Gartner uses the phrase "process templates" to refer to a broad range of process artifacts available commercially for free and for a fee. Process templates serve as implementation accelerators, while preserving process agility and integrity. Process templates are metadata based software assets that are extensible, allowing one enterprise, division or company to differentiate its business processes. A process template may include prebuilt user interfaces and forms, industry or process-specific rule sets, process design templates, prebuilt configurable process models, industry frameworks, and so on [6].

The development of even complex distributed application systems may reduce to the reuse of pre-modeled process templates from a repository, the customization of these templates, and the insertion of the application components in the style of plug-and-play. To be broadly applicable, however, future WF technology must provide a high flexibility in user assistance and more human-centric approaches that include an integral support for exception handling and dynamic structural changes [16].

In a S-BPM environment are the requirements on a BPMS similar. The ModelAsYouGo approach should be able to record an executable process model of ad-hoc processes, which were started from scratch or based on an existing model. Further, an S-BPM process template, that represents just the 20% best effort, could be established. The process model is expandable by ModelAsYouGo, if there is a need for a new process path.

## 4 ModelAsYouGo for Weakly Structured Business Processes

The discovered use cases show the demand for system that is able to execute process models without a pre-defined model and record the executed process instance in a business process model. This section introduces the principles of ModelAsYouGo by an easy scenario und show how to model with this approach subjects, messages, function, send and receive states.

Considering subjects as the main building blocks of an organization changes the way people see and act in their organizations. Business processes are considered as a set of messages that trigger individual behavior rather than a strictly predetermined way of enacting given steps towards reaching a business goal. Consequently, subject-oriented business process models decouple the description of the flow of work on an organizational level and the description of individual techniques within these flows of work. A S-BPM process model consists of two layers; the communication layer, describing the interaction among subjects, and the subject layer, containing the internal behavior of the subjects. Furthermore, there are approaches for a spatial distributed modeling of interaction and internal behavior, which work in a human-centered point of view [14]. ModelAsYouGo works from the perspective of the process participants, too. It enables the user to execute a new process instance in a communication model, allows him to process function states and to communicate via messages between the subjects of the process instances to other participants. After each process state the instance data will be collected and written to the backend where the descriptions will be created, stored and edited. From the perspective of a subject the process model will expand, while the recorded process information is available for later use.

The communication model works in the same way like used in S-BPM. There is a business process where the participants are represented as subjects and these subjects are able to communicate via messages. The internal behavior of a subject is hidden and has elements to enable function states as well as the states where messages can be sent or received. The following section will first establish a scenario, where the ModelAsYouGo approach will get explained. After that, ModelAsYouGo starts by authentication and authorization of the user. It will be explained, how to start a process instance and to create a subject instance with a function state. Finally the ModelAsYouGo approach with communication (send and receive) will be introduced.

### 4.1 A Simple Scenario

To keep the first scenario simple, there is just a participant A that does some action, before he sends a message to participant B and waits for answer. Participant B logs in the system and is able to receive the message, because he was addressed as receiver. After B received the message the scenario stops. The S-BPM basic modeling elements contain process, subject, function state, send state and receive state. So these five elements should fulfill the basic requirements of the S-BPM modeling notation and the ModelAsYouGo approach in the same manner.

The next subsections introduce how the following steps will be performed: authenticate and authorize the user; create a new process; create a new subject; create a function state; create a send state; transitions and state execution; create a receive state; create a receiver subject; create a receive state in the receiver subject.

### 4.2 Start a New Process Instance and Create a Function State

ModelAsYouGo from scratch starts with a user event. The user wants to initiate a new process instance. First of all the user has to log in to the system. Common credentials are username and password. If he is authorized, he is now able to record his process path by using ModelAsYouGo.

Next the participant gives the process a title, which describes the current process purpose. After that the user has to identify the subject, whose role it wants to play. There is a connection in S-BPM between subjects and the real users in a directory system. In S-BPM there is already a user management approach available, where n users are in n groups are mapped - these groups are assignable to roles. The subjects from the model are connected to a role in a 1:1 relationship. As a consequence the ModelAsYouGo approach provides a similar way.

So after the user identified himself as subject, he is able to describe the first state in the process. This state (State 1) is also the first state in the initial subject. In dependence on S-BPM, where the start state of a subject is also marked, it makes sense to set a flag. After confirmation of the user, the system should now be able to instantiate the process instance. A new unique process instance will be created, identified by an Id and titled by the process title input from the user. After that, the new State 1 will be registered as current state at the subject instance. The State 1 has the type “Function”, the name of the State 1 is like the user described it before. The system identifies all its process instance elements by Ids that the system handles in behind.

At this point the user was able to instantiate the process instance and the subject instance. The system enabled the user to “go” a new process instance till the first step, but this is just the first part of the ModelAsYouGo approach. Thereafter the second part of the system creates the description data, to store the process in a static model. Process, subject and state description will get appropriate Ids. The engine creates a process description name based on the title of the instance and the subject description name based on the identified subject name. Finally a function state (State 1) will be generated, based on the given state type (function) and the state name, e.g. “process start state”.

It is already obvious, that this approach leans strong against the S-BPM methodology to design process models. In summary the system has generated the instance data for State 1 (the first process state) and has written the description data to the backend, where the process model will be stored for later use. To leave a State 1, it is necessary to follow the transition to State 2. If there is no State 2, it is consequently not possible to go further to State 2. The current state of the process instance is still State 1, because there is no transition to follow. Till State 2 is not modeled, the process instance state remains in State 1.

### 4.3 Send Messages

In the next step of the scenario the subject A wants to send a message to subject B. It is assumed that the user, who created the process instance and registered the first function state (State 1), is still logged in to the system. The credentials of the participant are still assigned likewise the corresponding subject has a role assigned, so that the connection of the subject to a user from the user directory is guaranteed. The identity and the data of process and subject instance did not change. Like mentioned in the end of the previous section, the state of the process is still State 1, since there is no following State 2, because there is no transition between possible.

The participant, who owns the subject instance, initiates to send the message. Therefore he has to choose the type of action that he wants to do next. The appropriate state type is the send state. This send state needs a description, too. After that, the sender has to identify the receiver subject of the message. On the one hand it should be possible to reuse still existing subjects; on the other hand it should be possible to create a new subject. After the receiver subject is identified, the physical receiver of the message (the user from the user directory) has to be chosen. There are more ways in an S-BPM system to choose a message receiver. A receiver can be single users from the user database, the user group which contains the single users, all the users which have a preferred role assigned or all those receiver levels in a combined mode. To keep the scenario simple just a single user should receive a message. This receiver is user B and has no subject instance assigned, yet.

ModelAsYouGo has to register the current state Id in the subject instance of sender A and to create a new message. Messages in a S-BPM environment have the same basic principle like the other S-BPM modeling elements. There is the message type and the instance of the message. When modeling S-BPM in the classic approach, first the message type has to be created once; its instances – the messages – can be instantiated as often as required by the business user. After registering the message identifier, the system is able to set a transition between State 1 to this new send state (State 2). ModelAsYouGo is now able to leave State 1. The process instance is now in State 2, not in a State 1 anymore. ModelAsYouGo supports the user by creating the instance data. It uses therefore the user data with its credentials, the process instance and the available subject instance of the sending user. The subject instance of the receiver has to be created separately by the ModelAsYouGo System.

Most of the description data from the previous State 1 can be reused. Process description and subject description are the same, but will be expanded. A new state description, the message type and the description of the receiver subject have to be created. Also the transition description between the first function state and this send state will be created by ModelAsYouGo.

The next step in the scenario is to create a receive state (State 3) for the sender subject, because this participant has to wait for an answer now, to step forward in the process scenario. The user has to choose the next action and registers a receive state and describes it with a matching title like "Wait for answer". The sender of the message is also aware of the sender subject and the name of the sender of the message he is waiting for.

ModelAsYouGo registers now the new State 3 in the subject instance and also registers subject from whom it is waiting for an answer. The system is now able to execute the past send state (State 2) and is sending the message. The flow of the process is now hanging in the receive state (State 3), that the user created just before. The receiving subject A is able to identify the answer for its request by the process instance and also by the sending subject instance. The credentials are still in use, since the user did not log out. The process instance is still the same like in the state before, the subject instance identity is the same, but the subject data will be expanded.

At this point the instance data, the model can be written into the backend. The process and subject description will be reused again; a new state description will be created, the name of the state description is the input given from the user for this State 3. In summary of this subsection the user first created a send state (State 2) and the message. After he modeled the following state (State 3) the transition was runnable. Now the user A is still remaining in State 3, till he models the next state.

### 4.4 Receive a Message

In the scenario at the beginning the receiver B of the message from sender A is not aware, that he is involved in the business process. He logs in to the BPMS and notices that there is a message available. He has the option to accept or leave the message.

There are two options, when a receive state in the receiving subject of a message should be created. First option is when the sender sends the message. Then the subject instance of the receiver will be created and also the receive state. The other option is to create the receive state via ModelAsYouGo, when the receiver accepted the message. In this paper the second option will be examined. After login and accepting the message, the user's subject is involved in the process instance and this user is the current owner of the subject instance. The receiver has to confirm the acceptance of the message by registering a receive state (State 4). The next step is to describe this State 4. If there is more than one subject existing, it is necessary to remember the sender subject and the sender name of the message to send the answer exactly to this participant.

The system supports now the receiving user by registering the current state in the subject instance and registering the sender subject and user identity. After setting the receive state (State 4) the message is receivable. The process description data is now ready to be written to the backend. Process and subject description identity will be used to create the state description. The name for State 4 is the state title given from the user. In consequence there has to be created a new State 5, to be able to proceed from State 4. Next, the user has to choose one of the tree state types and create a state of this type, so that the transition between two states can be set up.

This subsection introduced how to receive a message form another subject. In summary ModelAsYouGo is able all model the five basic elements used in S-BPM. It is possible to model the subjects and the communication via messages between them. The modeling of function, send and receive states was explained as well as the meaning the transitions between the states.

## 5 Conclusion

This paper introduced ModelAsYouGo as a new approach to model an S-BPM business process via enabling the actors to record their subject communication and internal behavior, while they execute the process instance. The S-BPM paradigm works with communication, ModelAsYouGo realizes the same intend by collaboration. In ModelAsYouGo each user of a subject is just able to model its own internal behavior, to send or wait for messages. If there is more than one participant to the business process involved, collaboration is taking place. So ModelAsYouGo enables process actors, to design reliable process models in an easy and social way. Hereby the business world will be considered in the viewpoint of the actor in the first person perspective.

ModelAsYouGo is an enabler technology to realize ad hoc processes, ACM, flexible processes and process templates within an S-BPM driven BPMS system. Collaborative process modeling enables people, technical and nontechnical, to participate in the discovery, modeling, design, implementation, and optimization of a business process. Collaboration during the execution of an existing structured process in a BPMS allows the participant at any process state to leave the pre-modeled structured process and initiate an ad hoc collaboration.

The enabling of S-BPM modeling for process participants in a simple user perspective is powerful enough to design an S-BPM process model as a whole or in parts. A user is enabled to design a business process from scratch without using a common S-BPM designer like Metasonic Build. Emergent processes concerning one or more participants are designable directly by the actors. Thereby it doesn't matter if this emergent process is new or expands an existing process model. If an existing model was expanded, there is a completely new path of the control flow possible.

ACM within a subject-oriented BPMS is realized in the same way as ad hoc processes. In the same way, it doesn't matter, if there is a process model existing or not. ModelAsYouGo is here the tool where the process models will be modeled or adapted by the participants. With ModelAsYouGo the participator in a BPMS driven business process will gain in importance. The actor is processing a new business process instance in the first person perspective, while the system is recording his data to produce an executable process model. Later, the user is able to benefit from the recorded data, if he needs a similar process path in a different process instance. The whole collective of BPMS users will profit from the process knowledge discovery of a single user.

Flexible processes or process templates are handled like ad hoc processes which expand existing process models. ModelAsYouGo is here the tool which allows modifying the pre-defined process path, to come back or not. The work will get done by one or more actors which collaborate socially. The real value comes here from using the social component of the ModelAsYouGo approach, to influence existing processes rather than limiting it to social aspects of design.

We plan to evaluate the ModelAsYouGo approach and system within a case study, examining whether this approach eases process model creation for business users– Based on this finding we plan to evaluate the approach in real use cases in the application domain areas described in this paper.

## References

1. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement. In: Mitarbeiter Einbinden, Motivation und Prozessakzeptanz Steigern, pp. 61–84. Carl Hanser Verlag, München (2011)
2. Dourish, P.: Using Metalevel Techniques in a Flexible Toolkit for CSCW Applications. ACM Transactions on Computer-Human Interaction 5(2), 109–155 (1998)
3. Buhalis, D., Owen, R.: Business Process Reengineering. Business Process Management Journal 6(2), 113–121 (2010)
4. Powell, T.C.: Total Quality Management as Competitive Advantage: A Review and Empirical Study. Strategic Management Journal 16(1), 15–37 (1995)
5. Nakamura, S., Tan, T., Hirayama, T., Kawai, H., Komiyama, S., Hosaka, S., Nakamura, M., Yuki, K.: CGAA/EES at NEC Corporation, Powered by S-BPM: The Subject-Oriented BPM Development Technique Using Top-Down Approach. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 215–231. Springer, Heidelberg (2011)
6. Ukelson, J.: What to do when modeling doesn't work. In: Mastering the Unpredictable. How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done, pp. 29–39. Meghan-Kiffer Press, Tampa (2010)
7. Hill, J.: Gartner Reveals Five Business Process Management Predictions for 2010 and Beyond. Gartner Inc., Press Releases, Egham, UK (2010), http://www.gartner.com/it/page.jsp?id=1278415

8. Swenson, K.D.: Mastering the Unpredictable. How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done, Glossary, pp 314–317, Meghan-Kiffer Press, Tampa (2010)
9. Huth, C., Tas, N., Erdmann, I., Nastansky, L.: GroupProcess: Graphisch interaktives Management von Ad-hoc-Geschäftsprozessen im Web. Groupware Competence Center, GCC (2003)
10. Muehlen, M.z., Su, J. (eds.): BPM 2010 Workshops. LNBIP, vol. 66. Springer, Heidelberg (2011)
11. Keremans, M.: Case Management Is a Challenging BPMS Use Case, Gartner Inc., Gartner Research, ID Number: G00162739 (2008), `http://www.gartner.com/DisplayDocument?doc_cd=162739`
12. Wikipedia, Knowledge worker, `http://en.wikipedia.org/wiki/Knowledge_worker` (page visited October 13, 2011)
13. Herrmann, C., Kurz, M.: Adaptive Case Management: Supporting Knowledge Intensive Processes with IT Systems. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 80–97. Springer, Heidelberg (2011)
14. Oppl, S.: Subject-Oriented Elicitation of Distributed Business Process Knowledge. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 16–33. Springer, Heidelberg (2011)
15. Swenson, K.D.: Historical perspective. In: Mastering the Unpredictable. How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done, pp. 293–302. Meghan-Kiffer Press, Tampa (2010)
16. Reichert, M., Dadam, P.: ADEPTflex – Supporting Dynamic Changes of Workflows Without Loosing Control. JIIS Journal of Intelligent Information Systems (1998)
17. Cantara, M., Hill, J.: Signs That a BPMS Vendor Is Following One or More Technology Evolutionary Paths, Gartner Inc., Gartner Research, ID Number: G00168117 (2009), `http://www.gartner.com/DisplayDocument?doc_cd=168117`
18. Dixon, J., Jones, T.: Hype Cycle for Business Process Management, Gartner Inc., Gartner Research, ID Number: G00214214 (2011), `http://www.gartner.com/DisplayDocument?doc_cd=214214`
19. Wikipedia, Pareto Principle, `http://en.wikipedia.org/wiki/Pareto_principle` (page visited October 13, 2011)
20. Kemsley, S.: Leveraging Social BPM for Enterprise Transformation. In: Social BPM, Work, Planning, and Collaboration under the Impact of Social Technology, Future Strategies Inc., Book Division, Lighthouse Point FL, USA, pp. 77–93 (2011)
21. Das, M., Chow, L.: Next Generation BPM Suites Social and Collaborative. In: Social BPM, Work, Planning, and Collaboration under the Impact of Social Technology, Future Strategies Inc., Book Division, Lighthouse Point FL, USA, pp. 193–205 (2011)
22. Richardson, C., Vollmer, K., Le Clair, C.: Forrester TechRadar™ For BP&A Pros: Business Process Management Suites. In: The Need For Increased Business Agility Drives BPM Adoption, Forrester Research (2009), `http://www.forrester.com/rb/Research/techradar%26trade;_for_bp%26a_pros_business_process_management/q/id/53286/t/2`
23. Kemsley, S.: Enterprise 2.0 Meets Business Process Management. In: Handbook on Business Process Management 1, International Handbooks on Information Systems, pp. 565–574. Springer, Heidelberg (2010)

24. Olding, E., Rozwell, C., Sinur, J.: Social BPM: Design by Doing, Gartner Inc., Gartner Research, ID Number: G00200281 (2010), http://www.gartner.com/DisplayDocument?doc_cd=200281
25. Schwickert, A.C., Fischer, K.: Der Geschäftsprozess als formaler Prozess – Definition, Eigenschaften, Arten, Universität Mainz, Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Arbeitspapiere WI Nr. 4 (1996)
26. Richardson, C.: Foreword. In: Social BPM, Work, Planning, and Collaboration under the Impact of Social Technology, Future Strategies Inc., Book Division, Lighthouse Point FL, USA, pp. 5–8 (2011)
27. Kemsley, S.: Runtime Collaboration and Dynamic Modeling in BPM: Allowing the Business to Shape Its Own Processes on the Fly. Cutter IT Journal 23(2) (2010)

# Modeling Business Objectives for Business Process Management

Matthias Lohrmann and Manfred Reichert

University of Ulm
Databases and Information Systems Institute
James-Franck-Ring, 89069 Ulm, Germany
{matthias.lohrmann,manfred.reichert}@uni-ulm.de

**Abstract.** For application scenarios such as the management of business process variants or business process quality, business objective models assume the role of formal requirements definitions as in software engineering. However, effective concepts in this area still constitute a gap in the presently available array of business process management methods. To address this issue, this paper develops and shortly evaluates a refined business objective modeling approach. Our approach builds on use case-based effectiveness criteria, and on insights gained from assessing the state of the art. It derives required constructs and interrelations from application scenarios, and integrates these into a business objective meta-model. As an initial validation of our concept, we model a sample scenario and match the results against effectiveness criteria.

**Keywords:** business objectives, business goals, business process modeling, business process quality.

## 1 Introduction

Definitions for the term *business process* range from business process reengineering approaches [3,7] to trade associations [26] and current business process management (BPM) research [25]. They generally comprise a notion of *business goals* or *business objectives*, which can be viewed as an integral aspect reflecting the utilitarian nature of *business* processes. It is therefore interesting that formal modeling of business objectives is not covered by common business process modeling approaches such as BPMN [23], EPCs [19] or Workflow Nets [1].

To further illustrate the potential benefits of concise business objective modeling, we consider its possible use cases. In general, business objective models could assume the role of a formal requirements definition for business process design, implementation and enactment. Accordingly, application scenarios and benefits are generally comparable to broader requirements engineering [8,11].

From an a priori perspective, business objective models add a layer of abstraction to business process models. They allow discussing and documenting business objectives independently from a concrete implementation, and assessing the effectiveness of implementation options. From an ex post perspective, they can be

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 106–126, 2012.

used to determine whether a business process instance (or a set of instances) has terminated in a state being consistent to the organization's business targets. In addition to these general use cases, we point out three more specific exemplary scenarios that are closely related to present directions of BPM research: automated derivation of business process models, management of business process variants, and business process quality management.

**Scenario 1 (Automated Derivation of Core Business Process Models).** An available business objective model would comprise formalized information on the things a respective business process should create or alter including decision criteria, ordering constraints, etc. On that basis, it would be possible to formally derive minimum control flow implementation requirements. In addition, it would be possible to integrate the approach with process mining techniques [2]. For instance, real-world cases might be used to analyze the probability of control flow decision criteria given in a business objective. This would allow for automated or even continuous control flow optimization.

**Scenario 2 (Management of Business Process Variants).** The management of business process variants has emerged as an important BPM issue [6]. However, determining whether two process instances are variants of the same business process remains a "missing link" in this respect, especially when considering the mining of process variants or reference processes [12], or the refactoring of model repositories [24]. Formally modeling business objectives could contribute to closing this gap, as it would enable to assess the "equivalence" of process variants with respect to a business objective.

**Scenario 3 (Business Process Quality Management).** Business process quality management constitutes another emerging area of BPM research. In [14], a definition framework in this regard was developed. The concept of *efficacy*, i.e., whether a business process achieves its business objective, is crucial in this respect. Thus, formal modeling of business objectives constitutes an important prerequisite to effectively assess business process quality in design, implementation and enactment.

**Scenario 4 (Subject-oriented Business Process Management).** Subject-oriented business process management (S-BPM) constitutes an approach to address critical practical issues in BPM adoption, thereby significantly broadening its appeal [5]. The concept is based on shifting the paradigm of BPM away from formalizing tasks to be executed to the roles and interactions of subjects or stakeholders. As this potentially takes out some of the implicit formality of, e.g., strict control flow, it is all the more important to employ concise business objective models to ensure effectiveness, i.e. making sure that a process still achieves what it should.

Our research has shown that present approaches to business objective modeling do not yet effectively support the application scenarios set out above. This paper therefore seeks to develop a refined solution which is methodologically well-founded as well as readily applicable in future research and for practical purposes. The remainder of the paper is structured as follows: Section 2 presents our methodology including process examples to illustrate our ideas, basic terminology, and effectiveness criteria to evaluate results. Sections 3, 4, and 5 implement our methodology, from a review of available approaches to a refined business objectives meta-model. Section 7 concludes the paper and gives an outlook on future research.

## 2 Methodology and Preliminary Considerations

A business objective model constitutes a goal-bound artificial construct. We therefore apply the principles of design science [20,15]. Accordingly, our methodology consists of the following steps:

1. Define *effectiveness criteria* to assess the utility of design artifacts in the field of business objectives modeling (cf. Section 2.3).
2. Assess the *state of the art* based on the defined effectiveness criteria to determine gaps and obtain pointers towards a refined solution (cf. Section 3).
3. *Build* required terminology for business objectives based on effectiveness criteria and research into available approaches (cf. Section 4).
4. *Build* a meta-model for business objectives (cf. Section 5).
5. *Evaluate* our solution with respect to effectiveness criteria (cf. Section 6).
6. Discuss implications and further steps to leverage our results (cf. Section 7).

The remainder of this section presents two sample business processes we use to illustrate and evaluate our results, discusses required preliminary terms, and develops effectiveness criteria to evaluate present approaches as well as our work.

### 2.1 Sample Processes

Our first sample process stems from the field of accounting: *invoice checking and approval* constitutes a typical example of an administrative process which is often supported by workflow management systems. [1]

To ensure that our concepts are also applicable to domains where workflow applications are not as common, we include a second sample process from the field of healthcare. Figures 1 and 2 show the models of the two sample processes in terms of BPMN [23] flow charts.

*Example 1 (Sample Process A: Invoice Checking and Approval).* The business process starts with the receipt of a supplier invoice (activity A1). The invoice is then compared to the respective purchase order (A2). If deviations exist, these are subject to approval. In practice, this is often the case when, for instance, price

[1] Incoming invoices processing was used to illustrate the concept of business process reengineering by both Davenport and Hammer [3,7].

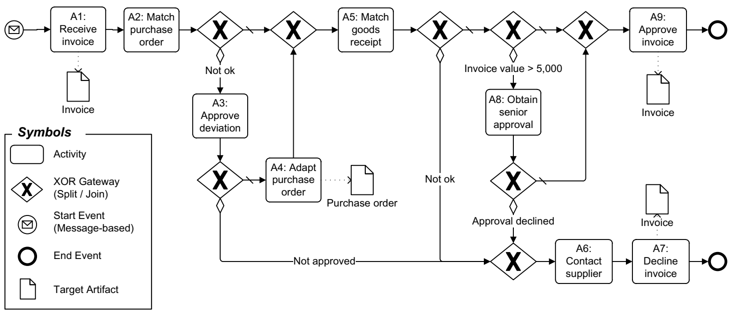



**Fig. 1.** Sample Process A: Invoice checking and approval

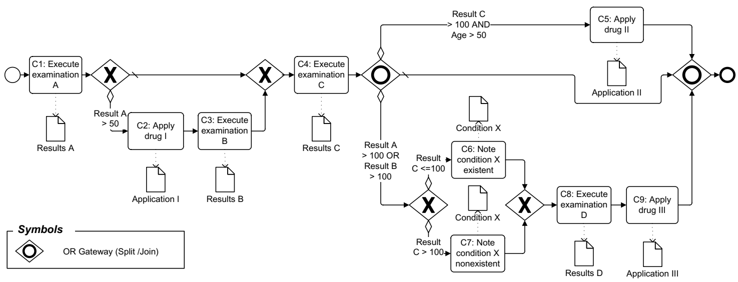



**Fig. 2.** Sample Process B: Medical examinations

data have not been maintained or when no purchase order has been entered into the ERP system. If the deviation is approved (A3), the purchase order is created or adapted (A4). Otherwise, the invoice is declined (A6, A7). In the next step, the invoice is matched against goods receipt (A5) and, depending on the result, either declined (A6, A7) or passed to the next check, which is based on the invoice value. For a value of more than 5,000, senior management approval (A8) is required. If this is granted, the invoice may finally be approved (A9).

*Example 2 (Sample Process B: Medical Examinations).* In our alternate example from the healthcare field, a medical examination A is performed (B1). Based on its result, a drug is applied (B2) and a second examination B (B3) is performed or not. A third examination C, which may only be carried out once examination A is completed, should follow in each case (B4). Thereafter, another drug is applied depending on the result of examination C (B5) and the age of the patient. In parallel, further steps are performed depending on the results of examinations A and B: First, the existence or non-existence of condition X is noted dependent on the result of examination C (B6, B7). Then, a fourth examination D is performed (B8). After completing examination D, application of a drug is required (B9).

### 2.2 Basic Terminology

Business processes constitute artifacts in the sense of design science [20] which operate within an *affecting and affected outer environment*. The outer environment of a business process consists of *target artifacts* and *resources*, i.e. things the process strives to create and alter, and things required to properly do so. Note that this perspective differs in some regards from the classic BPM concepts of process input and process output as it includes things usually not considered (e.g. capital goods). This topic was discussed in more detail in [14].

In the field of BPM, business objectives represent the targets an organization aims to achieve with a business process. As illustrated in Example 3, this can be understood on a strategic, collective operational or transactional level.

*Example 3 (Semantic Business Objective Levels).* As another exemplary business process, consider the handling of job applications in an enterprise. On a *strategic level*, the business objective of this process may be understood as providing the organization with the right "human resources". On a *collective operational level*, the business objective may be understood as properly handling the overall occurring cases of job applications. Depending on the required service level, the business objective may, for instance, be fulfilled if 90% of cases are managed correctly. On a *transactional level*, it may be understood as properly handling an individual application.

For the purpose of business objectives modeling, we define the term business objective on the transactional level to achieve consistency with common business process modeling approaches: In business process modeling, models are generally defined on a process instance [26] level without considering the cardinality of cases or instances. This means that a task that occurs many times for the business process, but one time per process instance is modeled as an individual activity, not as a set of activities.

Moreover, remember that an affecting environment may determine what actually needs to be induced to fulfil a business objective, for instance when considering decision processes (cf. Example 4).

*Example 4 (The Affecting Environment of Business Objectives).* Again, consider the job application process from Example 3. In this case, the business objective cannot be achieved by simply approving or disapproving an application. Rather, the respective hiring criteria are to be considered. Thus, they constitute the affecting environment of the business objective. As another example, consider medical treatments. In many cases, tests are required to find out which drugs are required. In this case, the test results are part of the affecting environment of the business objective.

Note that, when considering business objective levels as well as the affecting environment, the *organizational target* as the business objective on strategic level may differ from business objectives on lower semantic levels. This occurs when the affecting environnment restrains the business process from achieving the original organizational target (cf. Example 5). In other words, there may be states of the affecting environment where the business objective of the process is fulfilled while the corresponding organizational target has not been achieved.

*Example 5 (Business Objective Levels and the Affecting Environment).* When handling incoming job applications, the strategic organizational target will be to fill the respective positions. However, the business objective on more operational levels may well pertain to decline an applicant if her qualifications (as part of the affecting environment) are not sufficient.

In summary, this leads us to the following basic definition for business objectives to be further elaborated in our modeling approach:

**Definition 1 (Business Objective).** *A* ***business objective*** *in the sense of business process management constitutes a refinement of organizational targets to the transactional level. It pertains to an affecting and affected environment. The affecting and affected environment represent the things to be considered and the things to be manipulated to achieve the business objective. The business objective relates each state of its relevant affecting environment to a set of aspired states of the affected environment.*

### 2.3 Effectiveness Criteria

Considering the scenarios lined out in Section 1, business objective models as requirements definitions for business processes will generally be used to

- determine what needs to be done to achieve a business objective (e.g., as a starting point for structured business process design, or as in Scenario 1 from Section 1),
- assess whether a modeled business process enables to achieve its business objective (e.g., to evaluate design options, or as in Scenario 2), and
- assess whether a concrete business process instance has actually achieved its business objective (e.g., in testing, or as in Scenario 3).

Accordingly, the notion of an *achieved* function reflecting whether an aspired state of the affecting and affected environment of a business objective is reached is central to business objectives modeling.

Recapitulating the terms introduced in Section 2.2, business objectives are *achieved* by propagating target artifacts to an *aspired state*. However, which target artifacts need to be created or altered, and which states are considered as aspired may depend on other elements of the affecting environment.[2] Thus, business objectives cannot be recorded solely in terms of attributes of targets artifacts, but in terms of a set of consistency rules to be satisfied in respect to the entire environment. This set of rules must be complete and free of overlaps to ensure conformance can be assessed for each state of the outer environment.

[2] Note that the affecting environment of a business objective may differ from the affecting environment of an associated business process – the affecting environment of an efficacious business process will encompass, but possibly not be limited to, the affecting environment of its business objective (cf. [14]).

**Table 1.** Effectiveness Criteria for Business Objective Modeling Approaches

| | |
|---|---|
| *SR1* | *Consideration of the affecting environment:* Whether a business objective is achieved or not must be determined in terms of target artifacts *and* additional properties of the outer environment; e.g., in Sample Process A (cf. Example 1), the approved or disapproved invoice as a target artifact and the defined conditions for invoice approval as additional properties of the outer environment. |
| *SR2* | *Varying target environment:* The set of target artifacts to be created or altered as well as the concrete operations to be carried out on them may vary; e.g., in Sample Process A (cf. Example 1), the purchase order may have to be adapted or not, but the invoice must always be approved or disapproved. |
| *SR3* | *Order constraints:* There may be constraints regarding the order in which the activities of a process need to be executed in conformance with the business objective. Consider, for instance, Sample Process B from Example 2: drug application and examinations must occur in a specific order. It is important to note that these constraints actually represent constraints with respect to target artifacts manipulation, because, by definition, executing activities cannot constitute a business objective. |
| *UC1* | *Semantic interdependencies:* The approach should be apt to transparently capture semantic interdependencies between elements of the outer environment like mutual exclusivity or correlation. As an example for mutual exclusivity, consider the approval or disapproval of invoices in Sample Process A from Example 1 (cf. "pragmatic quality" in the sense of comprehension in [9]). |
| *UC2* | *Model compaction:* The approach should lead to a compact result in the sense of avoiding unnecessary content which might "hide" the relevance of model elements. For instance, in Sample Process A, it would be obstructive to model the effect of senior management approval for invoices below a value of 5,000 (cf. "semantic quality" in the sense of validity or relevance to the problem in [9]). |
| *UC3* | *Knowledge externalisation:* The approach should leverage implicitly available knowledge of the modeler (cf. "physical quality" in the sense of externalisation in [9]). |

Table 1 summarizes *effectiveness criteria* towards business objectives modeling. The *semantic requirements* SR1 to SR3 are based on the issues discussed above. They reflect the semantic content an approach needs to address to properly model business objectives. In addition, an effective modeling approach will also fulfill *usability criteria* UC1 to UC3 to support both modelers and users. The usability criteria are based on the considerations on model quality in [9]. Since we work on a meta-model level instead of the model level addressed in [9], we place special regard to the quality types of "physical quality", "semantic quality" and "pragmatic quality".

## 3 State of the Art

Models for business objectives or goals[3] have been proposed by Kueng and Kawalek [10], Neiger and Churilov [17], Soffer and Wand [21], and Lin and Sølvberg [13]. Markovic and Kowalkiewicz [16] proposed a business goal ontology as part of the SUPER project on semantic BPM (cf., e.g., [18]). For comparison, we include an approach by Engelman et al. towards goals modeling in enterprise architecture. Table 2 matches the approaches against semantic requirements SR1 to SR3. For reasons of brevity, usability criteria UC1 to UC3 are not considered.

[3] In the field of BPM, the terms are generally used as synonyms.

**Table 2.** Available Business Objective Modeling Approaches

| | *Evaluation against semantic requirements* (cf. Table 1) | | |
|---|---|---|---|
| *Source / focus* | *SR1* | *SR2* | *SR3* |
| *Kueng and Kawalek* [10]: Goals-based modeling, design evaluation | *Not fulfilled:* No formal measurable definition of goals | *Not fulfilled:* Goals are discussed on an abstract level only | *Not fulfilled:* Goals are discussed on an abstract level only |
| *Neiger and Churilov* [17]: "Value-focused thinking" to structure objectives | *Not fulfilled:* No formal measurable definition of objectives | *Not fulfilled:* "Functional objectives" on a more abstract level | *Not fulfilled:* "Functional objectives" on a more abstract level |
| *Soffer and Wand* [21]: Formalizing processes' contribution to "soft goals" | *Not fulfilled:* Business goals as any possible process termination state, goal achievement only pertains to target artifacts | *Partially fulfilled:* Implicitly considered: only one relevant process path required per target artifact | *Partially fulfilled:* Order constraints implicitly considered via consistent process paths |
| *Lin and Sølvberg* [13]: Goal ontology for semantic annotation in distributed environments | *Partially fulfilled:* Goals are seen as states of activities or artifacts, but no specification of respective artifact states | *Not fulfilled:* Goals are defined for activities instead of processes, no concept of goals changing with the environment | *Partially fulfilled:* Constraints are comprised in the meta-model, but not further specified as state of activities or the environment |
| *Markovic and Kowalkiewicz* [16]: Integrating goals into business process modeling | *Not fulfilled:* No concise definition of when a goal has been achieved | *Not fulfilled:* No notion of goals evolving with the environment | *Not fulfilled:* No notion of order constraints |
| *Engelsman et al.* [4]: Enterprise architecture goals modeling language | *Not fulfilled:* Hard goals concept, but no formal notion of goal achievement | *Not fulfilled:* No affecting environment concept | *Partially fulfilled:* Goal aggregation might be extended to include ordering |

Since the related approaches discussed generally aim at amending process models with a descriptive goals perspective and not necessarily at using business objectives as a formal requirements definition in a BPM context, it is not surprising that additional work is needed to develop a *business objectives meta-model* to fully address the criteria set out in Table 1.

## 4 Extended Business Objective Modeling Terminology

According to semantic requirement SR1 in Table 1, an effective approach to business objectives modeling must relate aspired states of target artifact properties to conditional states of additional properties of the outer environment. In the following, we will refer to the respective environmental properties as *elements of the target environment* (or, in short, *target elements*) and *elements of the conditional environment* (or, in short, *conditional elements*). Both sets of *environmental elements* may overlap, i.e., an environmental element may constitute a target element, a conditional element or both. We may conceive of environmental elements as "metering points" of sufficient semantic relevance to determine, in their totality, whether a business objective has been *achieved*. Note that the conditional elements correspond to the additional properties of the outer environment cited in semantic requirement SR1 in Table 1. The relevant "metering points" may be expressed as *binary state determinants*.

**Definition 2 (Binary State Determinants).** *A **binary state determinant** (BSD) is the combination of an environmental element with an absolute or relative state condition that is relevant to a business objective and that may or may not be **fulfilled**. **Conditional BSDs** and **target BSDs** refer to conditional elements and a target elements, respectively.*

On that basis, it would be possible to list all BSDs with respect to conditional elements, enumerate the possible states and relate them to the corresponding set of aspired states of the target elements, represented by target BSDs. This approach would link aspired target states to the affecting environment as demanded by semantic requirement SR1 (cf. Table 1). However, there would still be major issues regarding our effectiveness criteria, as presented in Table 3.

To address these topics, we introduce a business objectives modeling approach that (i) reflects distinct types of target BSDs, (ii) sets out with target BSDs instead of conditional BSDs, and (iii) avoids redundancies in its modeling of both the target and the conditional environment. To this end, we employ a number of terms summarized in the remainder of this section.

**Definition 3 (Target BSD types).** ***Target BSDs** are constituents of the business objective. To achieve a business objective, all respective target BSDs must assume **target values**. Dependent on the range of target values, we discern various target BSD types.*

*To achieve the business objective, **monovalent target BSDs** must assume a "true" value (target BSDs that may only assume a "false" value are to be rephrased accordingly). There is no condition attached. Note that target BSDs subject to order constraints must include "false" in their value range.*

**Table 3.** Basic Modeling vs. Effectiveness Criteria

| | |
|---|---|
| *SR2* | As all target BSDs are enumerated for each conditional state, the potentially limited relevance of individual target artifacts is "hidden". |
| *SR3* | Order constraints are not addressed, and still require an additional construct. |
| *UC1* | Semantic interrelations between elements of the outer environment, such as mutual exclusivity or correlation, are not captured. |
| *UC2* | For an individual target BSD, only a (typically small) part of the conditional environment is relevant. Hence, a relation matrix between conditional and target BSDs would only be sparsely populated. For instance, in Sample Process B (cf. Figure 2), the age of the patient is not relevant to examination B. This characteristic is not utilized which leads to a unnecessarily bloated model. |
| *UC3* | From a modeler's perspective, it is much easier to determine (e.g. by discussion with stakeholders) what the prerequisite conditions for a target BSD are than which target BSDs are determined by a conditional element, let alone *a priori* enumerating relevant conditional BSDs. Moreover, semantic interrelations or relevances (cf. UC1-2) are not addressed. Capturing available knowledge is thus impeded. |

**Table 4.** Target BSD Types

| | | *Aspired target BSD states* | | |
|---|---|---|---|---|
| ***Target BSD types*** | ***Condition states*** | Not fulfilled | Fulfilled | ***Example*** |
| *Monovalent* | n/a | | X | Examination A in Sample Process B |
| *Fully determinate bivalent* | Not fulfilled | X | | Invoice approval in Sample Process A |
| | Fulfilled | | X | |
| *Partially determinate bivalent* | Not fulfilled | X | X | Senior management approval in Sample Process A |
| | Fulfilled | | X | |
| *Trivalent* | Only 1st condition fulfilled | | X | Marking of condition X in Sample Process B |
| | Only 2nd condition fulfilled | X | | |
| | No condition fulfilled | X | X | |
| | Both conditions fulfilled | May not occur | | |

*To achieve the business objective,* ***fully determinate bivalent target BSDs*** *may assume either a "true" or a "false" value. We thus require only one condition attached to either "true" or "false".*

*To achieve the business objective,* ***partially determinate bivalent target BSDs*** *may assume either a "true" or a "don't care" value ("false" target BSDs are to be rephrased).*[4] *"True" is bound to a respective condition.*

*To achieve the business objective,* ***trivalent target BSDs*** *may assume a "true", a "false", or a "don't care" value. Trivalent target BSDs differ from bivalent ones as there are two conditions attached to "true" and "false". The conditions are mutually exclusive, but not comprehensive (i.e. one or none of the two can evaluate to "true" at the same time).* □

Table 4 provides an overview on the various target BSD types and the state they must assume to enable achieving the business objective depending on the state of their relevant conditional environment.

Note that trivalent target BSDs can also be understood as two partially determinate bivalent target BSDs referring to the same target element. However, modeling a trivalent target BSD as two bivalent target BSDs results in a loss of semantics because the two respective bivalent target BSDs' mutual exclusivity is not visible in the model.

**Definition 4 (Conditional propositions).** *Conditions attached to target BSDs can be expressed as* ***conditional propositions*** *consisting of conjunctively and / or disjunctively interlinked* ***conditional BSDs****. Unlike target BSDs, the value range of conditional BSDs is confirmed to "true" and "false". A target element may also act as a conditional element within one business objective.*

***Absolute conditional BSDs*** *compare one conditional element to an absolute value range.* ***Relative conditional BSDs*** *compare two conditional elements to each other.*

[4] "Don't care" implies that the business process needs to do nothing – consider, for instance, the target BSD "Purchase order value = invoice value" from Sample Process A in Figure 1, where we either need to adapt the purchase order or simply leave it as it is. Semantically, this represents the characteristic that the set of relevant target artifacts may change with the conditional environment.

**Table 5.** Order Constraints Modeling

| ***Order constraint:*** *the conditionally dependent target BSD must be fulfilled. . .* | ***Modeled conditional propositions*** | |
|---|---|---|
| | *"Father" target BSD* | *Conditionally dependent target BSD* |
| . . . before | Dependent target BSD only | Shared conditional proposition only |
| . . . after | Shared conditional proposition only | "Father" target BSD only |
| . . . at any time (no order constraint) | Shared conditional proposition only | Shared conditional proposition only |

*Target BSDs are considered as* ***conditionally equivalent*** *if the attached conditional propositions are equivalent or if, for fully determinate bivalent target BSDs, the attached conditional propositions are a negation of each other. Target BSDs are considered as* ***conditionally dependent*** *on another if a BSD's conditional proposition comprises the value another target BSD has assumed or should assume by way of a relative conditional BSD.* □

We identified the treatment of order constraints as a requirement towards business objective modeling (see semantic requirement SR3 in Table 1). To address this issue, we consider a number of characteristics of conditional propositions as specified in Definition 4:

- As shown in Example 6, a conditionally dependent target BSD shares conditional proposition of the "father" BSD.
- A conditional dependency exists for any two target BSDs where an *order constraint* applies; i.e., the dependent target BSD must be fulfilled before, after or at the same time as the "father" BSD.
- From a modeling perspective, it does not make a difference which BSD is the "dependent" one, because both are required to achieve the business objective.

We therefore introduce a convention to model conditional dependencies and order constraints as described in Table 5.

Note that conditionally dependent target BSDs to be fulfilled *at the same time* should be merged with their "father" BSD (i.e., the two underlying target elements should be treated as one as they must be manipulated concurrently anyway) or resolved into two (or more) sequences as appropriate.[5]

*Example 6 (Order Constraints Modeling).* Consider Sample Process B in Figure 2. The activity pairs B2 / B3 and B8 / B9 reflect that examination B has to be prepared by applying a drug while another drug is required after examination D. The applications of both drugs thus become elements of the target environment which are conditionally dependent on the respective examination.

In the first case, application of drug I is dependent on whether examination B *shall happen*. In the second case, the application of drug II is dependent on whether examination D *has happened*. Regardless of the requirements with respect to the order of activities, both drug applications are semantically dependent on the relevant examination and thus share the examination's conditional environment. However, they differ in terms of their order constraint in regard to

[5] Note that this issue is also not addressed in common process modeling approaches.

the respective examination. Nevertheless, both are part of the business objective, which – given the respective conditional environment – cannot be fulfilled unless the drugs are applied properly.

The considerations set out above enable to define business objective achievement on the basis of target BSDs and conditional propositions, thus addressing semantic requirement SR1 in Table 1.

**Definition 5 (Business objective achievement).** *A business objective is achieved iff each target BSD comprised in the business objective has assumed a state reflecting its conditional propositions. Thus, a business process has to approve or disapprove each conditional proposition and manipulate target artifacts accordingly.* □

Based on Definition 5 and our convention on the modeling of order constraints in Table 5, control flow in business processes is generally oriented at approving and disapproving conditional propositions. Optimizing control flow thus amounts to optimizing the approval and disapproval of conditional propositions that are, in turn, composed of conditional BSDs.

Accordingly, control flow in business processes will be designed based on the business objective's conditional propositions. It thus makes sense to refine the business objectives meta-model to represent properties of conditional propositions which are relevant to approval or disapproval. To this end, we discern necessary and sufficient sub-conditions as possible constituents of conditional propositions. In case of multiple conditional BSDs comprised in a conditional proposition, necessary and sufficient sub-conditions generally occur in pairs. In case of one conditional BSD in a conditional proposition, the conditional BSD amounts to the sole necessary and sufficient sub-condition.

**Definition 6 (Necessary and sufficient sub-conditions).** *For conditional proposition $CP := NC_1 \wedge NC_2$, $NC_1$ and $NC_2$ constitute **necessary sub-conditions**. Any part of a conditional proposition that is conjunctively linked to the entire remainder of the conditional proposition (e.g. any subterm in a conjunctive normal form) constitutes a necessary sub-condition. If any one necessary sub-condition is not fulfilled, the conditional proposition is disapproved.*

*For conditional proposition $CP := SC_1 \vee SC_2$, $SC_1$ and $SC_2$ constitute **sufficient sub-conditions**. Any part of a conditional proposition that is disjunctively linked to the entire remainder of the conditional proposition (e.g. any subterm in a disjunctive normal form) constitutes a sufficient sub-condition. If any sufficient sub-condition is fulfilled, the conditional proposition is approved.* □

Sufficient and necessary sub-conditions can be identified by building minimal conjunctive and disjunctive normal forms for each conditional proposition (e.g., by way of a Karnaugh-Veitch diagram). The respective subterms provide us with minimal ways to either approve or disapprove a target BSD. As they are relevant for any business process implementation of a business objective, we include them in our semantic business objectives meta-model.

**Table 6.** Semantic Target BSD Interrelations

| | Type of commmon sub-condition X | | | |
|---|---|---|---|---|
| | Target BSD A | | Target BSD B | |
| **Semantic interrelation** | Necessary | Sufficient | Necessary | Sufficient |
| Common conditional branch | $X$ | | $X$ | |
| Mutually exclusive (fully determinate bivalent only) | $X$ | | $\overline{X}$ | |
| B is correlated to A | $X$ | | | $X$ |
| *[No proposition]* | $X$ | | | $\overline{X}$ |
| Possibly common approval | | $X$ | | $X$ |
| *[No proposition]* | | $X$ | | $\overline{X}$ |

*Note:* $\overline{X}$ *refers to an inversed sub-condition*

To fully capture the semantic content of business objectives either formally or based on *a priori* knowledge, we also consider semantic interrelations between target BSDs beyond conditional equivalence or dependency (cf. Definition 4). Target BSDs may be semantically correlated or mutually exclusive. In our context, semantic correlation e.g. for two target BSDs infers that if the first BSD is required to achieve the business objective, the second BSD will be required as well.[6] Mutual exclusivity implies that the business objective cannot be fulfilled if two respective target BSDs are both fulfilled (i.e., $TargetBSD_1 \Rightarrow \neg TargetBSD_2$ and $TargetBSD_1 \Rightarrow \neg TargetBSD_1$). This is caused by "overlaps" in the conditional environment, i.e. conditional BSDs that are relevant for multiple target BSDs or in themselves correlated or mutually exclusive. Table 6 summarizes the possible semantic interrelations between two fully determinant bivalent target BSDs that occur with common sub-conditions.

Besides common sub-conditions, mutual exclusivities and concurrencies may also be caused by semantic interdependencies between conditional BSDs. Beyond the simple case of non-overlapping value ranges for conditional BSDs referring to a common conditional element, it is, however, not practical to capture these characteristics in business objective modeling. Accordingly, an effective semantic business objective model will reflect multiple occurrences of individual necessary or sufficient sub-conditions in various conditional propositions linked to target BSDs as well as mutually exclusive and concurrent conditional BSDs referring to a common conditional element.

## 5 Business Objective Meta-model

The semantic concepts discussed in the previous section can be integrated into the RML[7] meta-model presented in Figure 3.

[6] Note that temporal concurrency would be an even more strict requirement, as it would demand that target BSDs are fulfilled *at the same time*.

[7] We use the Referent Model Language (RML) notation defined in [22] because it provides a concise graphical notation for set theory constructs.

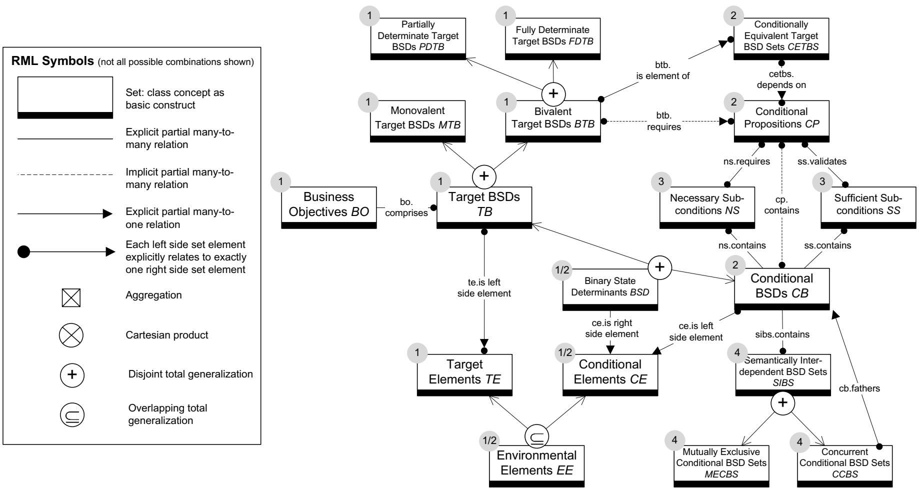



**Fig. 3.** Business Objective Meta-model

The following modeling steps illustrate how a business objectives model which is compliant to the presented meta-model can be obtained. Following these steps is a possibility we suggest with regard to criterium UC3 from Table 1. The numbering included in Figure 3 reflects the order of modeling steps. Relevant explanatory notes regarding modeling concepts and their interrelations are comprised as well. Capital letters represent sets of constructs where all elements are of the same type. Section 6 will apply the respective steps to a sample process.

**Step 1 (List Target BSDs).** Based on the Business Objective's target artifacts, all relevant Target BSDs including their types are listed. The respective Conditional Propositions may be modeled in a later step to make use of implicitly available knowledge on the business process first and limit modeling effort.

A Business Objective *bo* comprises a set of Target BSDs $TB_{bo}$. The Business Objective is achieved iff all comprised Target BSDs have assumed a target value.

According to Definition 3, a Target BSD might be a Monovalent Target BSD $mtb \in MTB$ or a Bivalent Target BSD $btb \in BTB$, i.e.,

$$TB = MTB \mathbin{\dot\cup} BTB$$

A Bivalent Target BSD might be a Fully Determinate Target BSD $fdtb \in FDTB$ or a Partially Determinate Target BSD $pdtb \in PDTB$, i.e.,

$$BTB = FDTB \mathbin{\dot\cup} PDTB$$

Note that we choose to model trivalent Target BSDs as two Partially Determinate Target BSDs as described in Section 4. Target BSDs and Conditional BSDs $cb \in CB$ are Binary State Determinants $bsd \in BSD$, i.e.,

$$BSD = TB \mathbin{\dot\cup} CB$$

A Binary State Determinant $bsd$ consists of a left-side element $leftside_{bsd}$, a right-side element $rightside_{bsd}$, and a relation $rel_{bsd}$, i.e.,

$$bsd = \big(leftside_{bsd},\ rightside_{bsd},\ rel_{bsd} \in \{=, <, >, \ldots\}\big)$$

Each Target BSD $tb$ refers to a Target Element $te \in TE$, and each Conditional BSD $cb$ to a Conditional Element $ce \in CE$ as its left-side element, i.e.,

$$leftside_{bsd} \in \begin{cases} TE & \text{if } bsd \in TB \\ CE & \text{if } bsd \in CB \end{cases}$$

Each Binary State Determinant $bsd$ refers to a Conditional Element $ce \in CE$ or to an absolute value range $vr \in VR$ as its right-side element, i.e.,

$$rightside_{bsd} \in CE \mathbin{\dot\cup} VR$$

Target Elements $te \in TE$ and Conditional Elements $ce \in CE$ are Environmental Elements $ee \in EE$. A Target Element may also be a Conditional Element, i.e.,

$$EE = TE \cup CE$$

A BSD $bsd \in BSD$ is fulfilled iff its relation $rel_{bsd}$ holds between its left-side element $leftside_{bsd}$ and its right-side element $rightside_{bsd}$, i.e.,

$$fulfilled(bsd) := leftside_{bsd}\ rel_{bsd}\ rightside_{bsd} \qquad \square$$

**Step 2 (Normalize Bivalent Target BSDs).** To "normalize" Bivalent Target BSDs, we build conditionally equivalent sets. To limit modeling effort, normalization can initially be conducted based on implicit knowledge without formally considering Target BSDs' Conditional Propositions.[8]

According to Definition 4, Fully Determinate Target BSDs are "rephrased" (i.e. negated) to join a conditionally equivalent set if the respective Conditional Proposition is a negation of a set's joint Conditional Proposition. Note that this is semantically not possible for Partially Determinate Target BSDs. Each Bivalent Target BSD $bsd \in BSD$ is an element of one Conditionally Equivalent Target BSD Set $cetbs_{btb}$ sharing one Conditional Proposition $cp_{cetbs_{btb}}$. Conditional Propositions are then made explicit as logical expression of Conditional BSDs considering the convention for order constraints in Table 5. A Conditional Proposition $cp$ is fulfilled iff its logical expression is fulfilled, i.e.,

$$fulfilled(cp) := \begin{cases} true & \text{if the logical expression for } cp \text{ is fulfilled} \\ false & \text{else} \end{cases}$$

On that basis and according to Definition 5, a Business Objective $bo$ is fullfilled iff the states of its Target BSDs and the respective Conditional Propositions are

[8] As an example for implicit available knowledge, consider Sample Process C: a physician will know that examination B requires drug I without modeling conditions first.

coherent considering Target BSD types, i.e.,

$$\begin{aligned} achieved(bo) := \forall mtb \in MTB_{bo} :\ & fulfilled(mtb) \wedge \\ \forall fdtb \in FDTB_{bo} :\ & fulfilled(cp_{fdtb}) \Leftrightarrow fulfilled(fdtb) \wedge \\ \forall pdtb \in PDTB_{bo} :\ & fulfilled(cp_{pdtb}) \Rightarrow fulfilled(pdtb) \end{aligned} \qquad \square$$

**Step 3 (Resolve Conditional Propositions).** Conditional Propositions are resolved into Necessary and Sufficient Sub-conditions according to Definition 6. Each Conditional Proposition can be decomposed into a set of Necessary Sub-conditions $NS_{cp}$ and a set of Sufficient Sub-conditions $SS_{cp}$, i.e.,

$$\begin{aligned} fulfilled(cp) &\Leftrightarrow \forall\ ns \in NS_{cp} : fulfilled(ns) \\ &\Leftrightarrow \exists\ ss \in SS_{cp} :\ fulfilled(ss) \end{aligned}$$

Each Necessary Sub-condition $ns$ and each Sufficient Sub-condition $ss$ contain a set of least one Conditional BSD $CB_{ns}$ or $CB_{ss}$. A Necessary Sub-condition $ns$ is fulfilled iff at least one of its Conditional BSDs is fulfilled, i.e.,

$$fulfilled(ns) \Leftrightarrow \exists\ cb \in CB_{ns} :\ fulfilled(cb)$$

A Sufficient Sub-condition $ss$ is fulfilled iff all of its Conditional BSDs are fulfilled, i.e.

$$fulfilled(ss) \Leftrightarrow \forall\ cb \in CB_{ss} :\ fulfilled(cb)$$

Necessary and Sufficient Sub-conditions are modeled in consolidated form, i.e., equivalent sub-conditions for multiple Conditional Propositions are modeled only once. The decomposition of Conditional Propositions into sub-conditions can also be used to identify conditional equivalences not recognized yet. □

**Step 4 (Consolidate Conditional BSDs).** To consolidate Conditional BSDs, we identify Semantically Interdependent BSD sets. A Semantically Interdependent BSD Set $sibs$ comprises a number of Conditional BSDs $CB_{sibs}$ and may either be a Mutually Exclusive Conditional BSD Set $mecbs$ or a Concurrent Conditional BSD Set $ccbs$. Each Mutually Exclusive Conditional BSD Set comprises at least two Conditional BSDs with:

$$fulfilled(cb) \mid cb \in CB_{mecbs} \Rightarrow \nexists\ ecb \in (CB_{mecbs} \setminus cb) :\ fulfilled(ecb)$$

Each Concurrent Conditional BSD Set comprises at least one Conditional BSD and refers to one Conditional BSD $cb_{father_{ccbs}}$ which "fathers" the set:

$$fulfilled(cb_{father_{ccbs}}) \Rightarrow \forall\ ccb \in CB_{ccbs} :\ fulfilled(ccb)$$

Mutual exclusivity of Conditional BSDs propagates to Necessary Sub-conditions that consist of just the one Conditional BSD, rendering the respective Conditional Propositions and hence Target BSDs mutually exclusive as well. Semantic correlation propagates to Sufficient Sub-conditions that consist of just the one

**Table 7.** Sample Target BSDs

| Target BSDs | Target BSD types |
|---|---|
| Result A available | Monovalent |
| Drug I applied | Fully determinate bivalent |
| Result B available | Fully determinate bivalent |
| Result C available | Fully determinate bivalent |
| Drug II applied | Fully determinate bivalent |
| Condition X marked | Partially determinate bivalent |
| Condition X not marked | Partially determinate bivalent |
| Result D available | Fully determinate bivalent |
| Drug III applied | Fully determinate bivalent |

Conditional BSD, rendering the respective Conditional Propositions and hence Target BSDs semantically correlated as well. [9]

Mutual exclusivity and semantic correlation are most obvious if the respective BSD set relates to the same Conditional Element. In that case, mutual exclusivity is caused by non-overlapping value ranges, and correlation is caused by partial quantity relations in value ranges. However, this is not a strict prerequisite.

Being aware that not all interdependencies in the outer environment are generally known to the modeler, note that this modeling step will usually lead to a partial result reflecting best knowledge. □

## 6 Sample Validation

This section presents an initial validation of the approach presented in Section 5 through application to the medical examination process from Figure 2 followed by a short evaluation against our effectiveness criteria.

### 6.1 Sample Application

We retrace the steps presented in Section 5 for Sample Process B:

**Step 1 (List Target BSDs including types).** For Sample Process B in Example 2, note that "Examination C executed" is not monovalent due to order restrictions (Examination C can only be executed after Examination A). Moreover, we assume that medical examinations as well as medications are not arbitrary, i.e. they should only be executed in case of a clear indication. Note that the originally trivalent Target BSD "Condition X marked" is deconstructed into two Partially Determinate Target BSDs. Results are presented in Table 7. □

**Step 2 (Normalize Bivalent Target BSDs).** There are no Conditionally Equivalent Target BSD Sets containing more than one Target BSD in our example, as illustrated in Table 8. For comparison, we also show how the normalized Target BSD sets would change when not considering order constraints. □

**Step 3 (Resolve Conditional Propositions).** Table 9 shows the resolution of Conditional Propositions into Necessary and Sufficient Sub-conditions. □

[9] See Table 6 for semantic relations caused by common sub-conditions.

**Table 8.** Sample Normalization of Target BSDs

Result with consideration of order constraints:

| $CETBS_{bo}$ | BSD types | Conditional Propositions |
|---|---|---|
| Result A available | Monovalent | none |
| Drug I applied | Fully determinate bivalent | [Result A > 50] |
| Result B available | Fully determinate bivalent | [Drug I applied] |
| Result C available | Fully determinate bivalent | [Result A available] |
| Drug II applied | Fully determinate bivalent | [Result C > 100] AND [Age > 50] |
| Condition X marked | Partially determinate bivalent | ([Result A > 100] OR [Result B > 100]) AND [Result C ≤ 100] |
| Condition X not marked | Partially determinate bivalent | ([Result A > 100] OR [Result B > 100]) AND [Result C > 100] |
| Result D available | Fully determinate bivalent | [Result A > 100] OR [Result B > 100] |
| Drug III applied | Fully determinate bivalent | [Result D available] |

Alternative result without consideration of order constraints:

| $CETBS_{bo}$ | BSD types | Conditional Propositions |
|---|---|---|
| Result A available, Result C available | Monovalent | none |
| Drug I applied, Result B available | Fully determinate bivalent | [Result A > 50] |
| ... | ... | ... |
| Result D available, Drug III applied | Fully determinate bivalent | [Result A > 100] OR [Result B > 100] |

Figure 4 presents a graphical notation of the results up to now based on the exemplary content for Sample Process B. The format is simplified as it presents either Necessary or Sufficient Sub-conditions (in case of only one Conditional BSD comprised in a Conditional Proposition, the differentiation is unnecessary). Because modeling is executed in a consolidated form, there is just one "column" for each Conditional BSD or Sub-condition comprised in Figure 4. Conditional Elements which are also target Elements (this is the case for all Conditional Elements except the patient's age) are comprised in the "line" of the respective Target BSD. The figure is to be read as follows: to achieve the Business Objective,

- the monovalent Target BSD set must be fulfilled,
- all elements of Bivalent Target BSD sets for which we modeled Necessary Sub-conditions must be fulfilled if all Sub-conditions for the set are fulfilled, and
- all elements of Bivalent Target BSD sets for which we modeled Sufficient Sub-conditions must be fulfilled if at least one Sub-condition for the set is fulfilled.

Note that circular relations between Target BSDs (i.e., one Target BSD as Conditional Element of another which is also a Conditional Element of the first Target BSD etc.) must not occur, because in that case the Business Objective could not be achieved by any business process. Figure 4 can thus be read from the top down.

**Step 4 (Consolidate Conditional BSDs).** In our sample case, we only consolidate on the basis of Conditional Elements shared between Conditional BSDs,

**Table 9.** Sample Target BSDs with Resolved Conditional Propositions

| | Conditional Propositions | |
|---|---|---|
| $CETBS_{bo}$ | Necessary Sub-conditions | Sufficient Sub-conditions |
| Drug I applied | [Result A > 50] | [Result A > 50] |
| Result B available | [Drug I applied] | [Drug I applied] |
| Result C available | [Result A available] | [Result A available] |
| Drug II applied | ([Result C > 100])<br>([Age > 50]) | ([Result C > 100] AND [Age > 50]) |
| Condition X marked | ([Result A > 100] OR [Result B > 100])<br>([Result C ≤ 100]) | ([Result A > 100] AND [Result C ≤ 100])<br>([Result B > 100] AND [Result C ≤ 100]) |
| Condition X not marked | ([Result A > 100] OR [Result B > 100])<br>([Result C > 100]) | ([Result A > 100] AND [Result C > 100])<br>([Result B > 100] AND [Result C > 100]) |
| Result D available | ([Result A > 100] OR [Result B > 100]) | ([Result A > 100])<br>([Result B > 100]) |
| Drug III applied | [Result D available] | [Result D available] |

i.e., we assume no further semantic interrelations between Conditional BSDs. Consolidation results can thus be easily derived from Figure 4, as we only need to consider line by line:

– Concurrent Conditional BSD Set: [Result A > 50] ⇒ [Result A available]
– Mutually Exclusive Conditional BSD Set: [Result C > 100] ⇔ ¬ [Result C ≤ 100]

Accordingly, application of Drug I and Examination C are correlated, and marking Condition X is mutually exclusive with application of Drug II and (obviously) not marking Condition X. □

## 6.2 Evaluation against Effectiveness Criteria

To evaluate our results, we consider the criteria defined in Table 1:

– *SR1:* The approach builds on target and conditional elements. Accordingly, both relevant aspects of the outer environment are covered effectively.
– *SR2:* The relevance of Target BSDs is determined considering the conditional environment. Together with Partially Determinate Bivalent Target BSDs, this enables target artifact sets varying with the conditional environment.
– *SR3:* Order constraints can be modeled via a convention (cf. Table 5).
– *UC1:* Semantic interdependencies are captured via the normalization of Target BSDs and conditional consolidation. Necessary and sufficient sub-conditions can directly be used to optimize control flow via approval / disapproval strategies.
– *UC2:* The resulting model is compact and apt for graphic presentation (cf. Fig. 4). Imagine, for comparison, full enumeration of the conditional environment and the related aspired states. There are no redundant model elements.
– *UC3:* By setting out with target elements, modeling is, in our opinion, intuitive and less prone to errors of omission. The approach also allows capturing available semantic knowledge before formal modeling. Available modeler knowledge could be captured through the "guided" modeling steps – however, this topic is obviously subject to individual preferences.

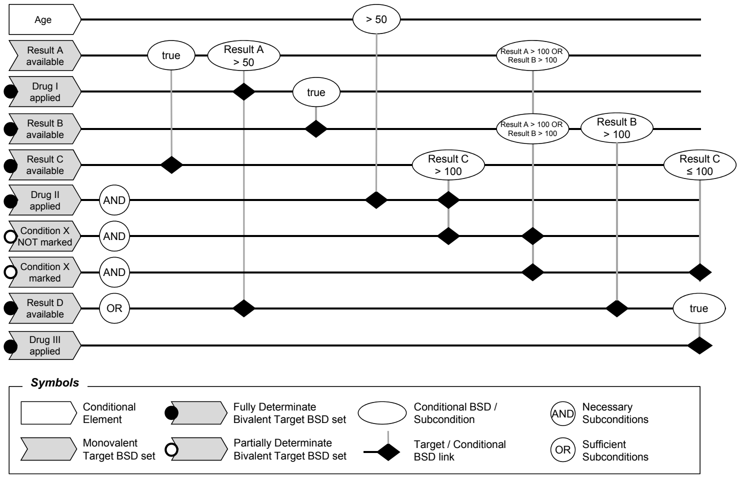



**Fig. 4.** Sample Conditional Consolidation

## 7 Conclusion

We developed an approach to business objective modeling by deriving a semantically enriched meta-model and a corresponding modeling methodology. The approach fulfills semantic requirements deducted from typical application scenarios as well as additional effectiveness criteria for practical adoption. Most prominently, and in contrast to related work, it addresses both the affecting and the affected environment of business objectives. We intend future work in this area to focus on promising application scenarios facilitated by our approach to business objectives. As an example, consider automated ongoing optimization of control flow from Scenario 1. Leveraging the concept of necessary and sufficient sub-conditions might be very beneficial in this respect. Beyond the use cases lined out already, we aim at exploring additional areas of application such as formal control of business process chains in functionally structured organizations or in service-oriented architectures.

## References

1. van der Aalst, W.M.P.: The application of Petri nets to workflow management. J. Circ. Syst. Comput. 8(1), 21–26 (1998)
2. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: a research agenda. Comput. Ind. 53(3), 231–244 (2004)
3. Davenport, T.J., Short, J.E.: The new industrial engineering: Information technology and business process redesign. Sloan Mgmt. Rev. (4), 11–27 (1990)
4. Engelsman, W., Quartel, D., Jonkers, H., van Sinderen, M.: Extending enterprise architecture modelling with business goals and requirements. Ent. Inf. Sys. 5(1), 9–36 (2011)

5. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
6. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. J. Sw. Mnt. Ev. Res. Pract. 22(6-7), 519–546 (2010)
7. Hammer, M.: Reengineering work: don't automate, obliterate. Harvard Bus. Rev. 68(4), 104–112 (1990)
8. Hull, E., Jackson, K., Dick, J.: Requirements Engineering, 3rd edn. Springer, Heidelberg (2011)
9. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. Europ. J. of Inf. Sys. 15, 91–102 (2006)
10. Kueng, P., Kawalek, P.: Goal-based business process models: creation and evaluation. Bus. Process Manag. J. 3(1), 17–38 (1997)
11. van Lamsweerde, A.: Goal oriented requirements engineering: a guided tour. In: Proc. 5th RE. IEEE (2001)
12. Li, C., Reichert, M., Wombacher, A.: Mining business process variants: Challenges, scenarios, algorithms. Data & Knowl. Eng. 70(5), 409–434 (2011)
13. Lin, Y., Sølvberg, A.: Goal Annotation of Process Models for Semantic Enrichment of Process Knowledge. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 355–369. Springer, Heidelberg (2007)
14. Lohrmann, M., Reichert, M.: Understanding Business Process Quality. In: Advances in Business Process Management. Springer, Heidelberg (2012) (accepted for publication)
15. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decis. Support Syst. 15(4), 251–266 (1995)
16. Markovic, I., Kowalkiewicz, M.: Linking business goals to process models in semantic business process modeling. In: Proc. 12th EDOC, pp. 332–338. IEEE (2008)
17. Neiger, D., Churilov, L.: Goal-Oriented Business Process Modeling with EPCs and Value-Focused Thinking. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 98–115. Springer, Heidelberg (2004)
18. Pedrinaci, C., Domingue, J., Alves de Medeiros, A.K.: A Core Ontology for Business Process Analysis. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 49–64. Springer, Heidelberg (2008)
19. Scheer, A.W., Thomas, O., Adam, O.: Process Modeling Using Event-Driven Process Chains. In: Process-Aware Information Systems, pp. 119–145. Wiley (2005)
20. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press (1996)
21. Soffer, P., Wand, Y.: On the notion of soft-goals in business process modeling. Bus. Process Manag. J. 11(6), 663–679 (2005)
22. Sølvberg, A.: Data and What They Refer to. In: Chen, P.P., Akoka, J., Kangassalu, H., Thalheim, B. (eds.) Conceptual Modeling. LNCS, vol. 1565, pp. 211–226. Springer, Heidelberg (1999)
23. The Object Management Group: Business Process Model and Notation: Version 1.2 (2009), http://www.omg.org/spec/BPMN/1.2
24. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Comput. Ind. 62(5), 467–486 (2011)
25. Weske, M.: Business Process Management. Springer, Heidelberg (2007)
26. WfMC: Workflow Management Coalition terminology & glossary 3.0, document reference WFMC-TC-1011 (1999), http://www.wfmc.org

# PCA-C: A Process-Centric Approach for Integrating and Managing Cloud Services

Matthias Kurz[1], Gunnar Billing[2], Karl Hettling[3], and Holger von Jouanne-Diedrich[4]

[1] University of Erlangen-Nuremberg, Information Systems II
Lange Gasse 20, 90403 Nuremberg, Germany
matthias.kurz@wiso.uni-erlangen.de
[2] Atos, Werner-von-Siemens-Str. 60, 91052 Erlangen, Germany
[3] Atos, Otto-Hahn-Ring 6, 81739 München, Germany
[4] Atos, Lyoner Straße 27, 60528 Frankfurt am Main, Germany
{gunnar.billing,karl.hettling,holger.jouanne-diedrich}@atos.net

**Abstract.** Cloud computing allows enterprises to consume computational power and IT services as if they were commodities. This enables businesses to respond to new market demands faster and more cost-efficiently. However, integrating new IT services in the existing service portfolio is not simply a matter of technically integrating these services. Instead, the business users have to be enabled to control how cloud services are used in their day-to-day work. Based on real-world projects of a leading global IT service provider, this contribution suggests a process-oriented architecture that enables business users to directly participate in the integration of cloud services.

**Keywords:** BPM, BPM 2.0, SOA, cloud computing, IT service management, ITSM, ITIL.

## 1 Introduction

Cloud computing allows enterprises to consume computational power and IT services as if they were commodities. This enables businesses to respond to new market demands faster and more cost-efficiently. However, integrating new IT services in the existing service portfolio is not simply a matter of technically integrating these services.

Empowering the business users to control and adapt how cloud services are used in their day-to-day routine is a critical success factor for the use of cloud services, as a lack of direct influence requires expensive, time-consuming and error-prone coordination of business users and IT experts about the business processes the services are used in [1, 2].

This contribution suggests PCA-C, a process-oriented approach for integrating cloud services into the existing service portfolio. PCA-C comprises a technical architecture as well as a corresponding procedure model. The approach is the result of

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 127–144, 2012.

the experience gained during a vast number of customer projects related to cloud computing that have been undertaken by Atos - one of the top-10 global IT outsourcing service providers. PCA-C is based on TANGO which is the service automation solution of Atos [1, 2].

## 2 TANGO: BPM Meets SOA

Over the last two decades, the IT services industry has undergone a drastic change. The way of delivering IT services has seen a major shift from building bespoke software systems for individual customers to offering a portfolio of standardized services from which the customer selects a set of pre-defined, off-the-shelf solutions for his purposes. In this sense, services have turned into products which are developed, sold, and managed in a retail-like fashion. Processes have changed accordingly, with a focus on the automation of previously manual tasks. The driving force behind this development are the customers and their increased demand for having IT services delivered and adapted as fast as their own businesses are changing. In particular cloud computing as the upcoming delivery model for IT services focuses on these demands by:

- Flexible sourcing and combination of services from different internal and in particular external service providers
- Faster integration of services, i.e. optimization of time to market
- High automation degree for provisioning processes to enable a real on demand offering
- Pay per use

Within this period of substantial change, Atos has undergone various efforts to improve and automate its methodology and toolset of defining, implementing, deploying, and delivering IT services. These include:

- The *reference process architecture* (RPA), a central top-level process model for all business processes, in particular the processes from the IT infrastructure library (ITIL).
- The *operational framework*, a common set of tools and concepts to provision, bill, and report IT services based upon the RPA processes.

Although the resulting methods and tools have been successfully introduced to the company's operation, experience has shown further need for improvement. Two observations turned out to be especially significant: First, time-consuming centralized business process management lifecycles find little acceptance in operational departments. Second, the complexity of IT systems makes it difficult for operational departments to adapt service processes relying on these systems and thus limits their influence on automating them.

Based on these experiences, three major requirements were identified:

1. Operational departments should have more and direct influence on the design of their services and on the respective IT support.

2. The complexity of IT systems needs to be hidden from the operational departments and the related development activities have to be kept simple.
3. The building block principle underlying service-oriented architectures has to be leveraged to reduce the overall complexity of using IT systems.

As a consequence, Atos has developed the TANGO approach as the next-generation operational framework for IT services which combines the essences of model-driven architecture, service-oriented architectures, and business process management into a unified automation platform for IT services. The goal is to maximize the influence of operational departments on the automation of their services, with the effect of significantly speeding up the design and implementation of IT services.

# 3 Foundations

The PCA-C approach is built upon two foundations: Self-organization and business process management (BPM). By applying the concept of managed self-organization to BPM, the BPM 2.0 approach is derived. This BPM 2.0 approach is in turn applied to the integration of cloud services. The TANGO approach provides the technical foundation of PCA-C.

## 3.1 Business Process Management

Although BPM is a widely utilized concept for organizing work spanning multiple organizational units, it provides only limited support for flexible business processes. In this contribution, the term *flexibility* refers to the ability of an organization to swiftly adapt its processes to new challenges.

In the predominant BPM concepts, the design of business processes (BP) is assigned to highly specialized personnel [3]. This is inspired by Taylor's principle of separating planning and execution of work [4]. These specialists create formal plans describing business process execution by using modeling methods like ARIS (Architecture of Integrated Information Systems) [5] or modeling languages like BPMN (Business Process Modeling Notation) [6].

There are two fundamental approaches for improving business processes [7]: (1) *Business process reengineering* (BPR) is a well-known approach that radically changes organizations or vital processes without focusing on existing structures [8]. (2) *Continuous process improvement* (CPI) is the constant, incremental improvement of processes [7]. This concept is based on the incremental concepts KAIZEN [9] and the Deming-Cycle [10]. In practice, these approaches are widely used for improving BP. However, they offer only limited support for highly flexible BP.

Both BPR and CPI rely on the separation of planning and execution. However, the necessary thorough analysis and planning rely on constant environmental conditions, which in many cases is not true anymore [11]. This limits the benefit of detailed ex ante analysis and planning of business processes. Conducting BPR projects proves to be inefficient for processes with few instances due to the effort required for such projects [12].

Designing business processes requires both knowledge about a businesses' structure as well as a comprehensive understanding of the market and customers. For operational business processes such know-how is available in the departments interacting with customers. Yet it is only rarely utilized [13]. This is especially true for stakeholders who are not part of the department which is responsible for the respective business process.

### 3.2 BPM 2.0

The *BPM 2.0* concept presented in [13, 17] aims at offering a comprehensive BPM methodology which incorporates self-organization [14, 15, 16]. BPM 2.0 provides a procedure model encompassing all BPM lifecycle phases. The suffix *2.0* indicates that integrating employees who execute business processes as part of their day-to-day tasks in the improvement of these processes is the core idea of BPM 2.0. Thus, we suggest the following definition of BPM 2.0:

> *BPM 2.0 is a business process management approach which encourages employees to improve "their" business processes. Web 2.0 technologies are leveraged to enable contributions from employees with little BPM expertise.*

Classical BPM lifecycle models like [25] or [12] distinguish between designing, implementing, executing, and analyzing BP. As BPM 2.0 encourages a larger audience to contribute in process innovations (PI), integrating these contributions is vital for BPM 2.0. Innovation management lifecycle models like [26] exhibit a similar structure: Innovations are identified, selected, implemented, and finally evaluated. The main difference between management lifecycle models for innovations and BP is that the former assume that a larger quantity of individuals is involved in creating innovations than in creating business process improvements. Therefore, review mechanisms for innovations have to cope with a larger number of candidates.

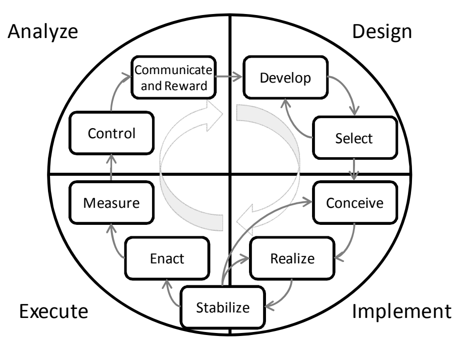



**Fig. 1.** BPM 2.0 lifecycle [13]

Fig. 1 depicts the BPM 2.0 lifecycle which combines the innovation [26] and BP management lifecycles [25] previously mentioned.

During the *design* phase, employees involved in the respective BP create and refine (*develop*) PI. By using a web based platform, alternatives are discussed in discussion boards, documented in wikis, and formalized as graphical process models. Promising PI are selected and submitted for a formal review which serves as a bridge between self-organization and classical hierarchical decision-making. The *implementation* phase is about realizing PI both organizationally and technically. Once implemented, the revised processes are *executed* by the employees who can contribute their experience and best-practices to the process modeling environment. That way, new process innovations may be triggered. The performance of PI is assessed and rewards for successful PI are granted in the *controlling* phase.

### 3.3 Cloud Computing

The term *cloud computing* is defined in various ways [27]. While [27] describes cloud computing as distributed systems which are virtualized and presented as "one or more unified computing resources" [27], [28] emphasizes the pay-as-you-go aspect that comes with cloud computing. Instead of having to invest large amounts of money in order to set up a scalable infrastructure for providing services, businesses pay the cloud service providers to run and maintain services depending on the actual resource consumption [28]. This allows businesses to swiftly meet peaks in resource consumption as well as scaling back the costs of the provided services when the demand lessens.

The notion of having computing resources available whenever needed – similar like electrical power – can be considered to the technical answer to Carr's well-known article [29] about the commoditization of information technology. The economies of scale of large datacenters and prevalent broadband internet connections are the major drivers for this commoditization. Following the idea of commoditization, in this contribution, the companies integrating cloud services into their existing service portfolios are designates as customers.

Typically, three cloud service layers are distinguished [30]:

- On the *infrastructure layer*, the cloud service provider offers basic computing infrastructure functions like a virtualization solution. Within this infrastructure, customers can install own software, services, or platforms. This is also known as *infrastructure as a service* (IaaS).
- On the *platform layer*, a runtime environment and a complementing framework for running and developing own cloud services are provided. Customers develop own cloud services that may run within this runtime. This is also known as *platform as a service* (PaaS).
- On the *application layer*, services are provided by an external supplier. The customer integrates these services in his own infrastructure. This is also known as *software as a service* (SaaS).

As this contribution strives to increase the degree of control business users exercise over the integration of cloud services into the companies' respective service portfolio, it focuses on the application layer which provides complete cloud services. The remaining two layers explicitly address IT experts and thus are not considered here.

There are two basic ways of integrating application layer cloud services in the companies' service architecture:

- When integrating cloud services on the *presentation level*, cloud services are provided as web applications. Hyperlinks to these applications provide an easy way of linking these applications with existing company portals of the customers.
- The integration on the *logic level* is more complex. In this case, the cloud service offers an interface that can be accessed programmatically by customers.

While the presentation level integration is trivial to implement, this approach suffers from the limitation that data cannot be synchronized automatically between the customers' and the cloud service providers' systems. The logic level integration allows a substantially more comprehensive integration. However, due to the complexity of tackling with IT systems, the logic level integration limits the commoditization of cloud services by reducing the amount of control business users without a technical background can exercise over the integration of the cloud services.

## 4 Process-Centric Cloud Services Integration

The process-centric perspective of PCA-C allows business users who integrate cloud services to focus on the day-to-day operation of services instead of programming languages. While workflow management systems (WFMS) have a similar process-centric perspective on the automation of processes that are necessary to provide services, they require too much expertise to develop executable services. Furthermore, they provide little explicit support for integrating cloud services.

The process-centric cloud services integration approach (PCA-C) reduces the complexity of integrating application layer cloud services in company-specific services by distributing the technical complexity among three modeling levels. As in most scenarios, only little technical expertise is required, business users can easily create and improve services which rely on cloud services. Wrappers ensure that cloud services can be integrated into the service portfolio without having to tackle the technical complexity associated with invoking external services. Such complexity stems from technical details like proprietary authentication mechanisms, proprietary data models, or the need for complex data structures.

### 4.1 Process-Oriented Services

In the process-centric perspective illustrated in Fig. 2, *services* consist of three components. The activities required for providing a service are defined by a *process*. These activities are executed by either humans or electronic *resources*. All final or temporary results like paper-based forms or entries in electronic databases are referred to as *artifacts*. The process defines which tasks the resources have to fulfill and which artifacts are the result of the respective tasks.

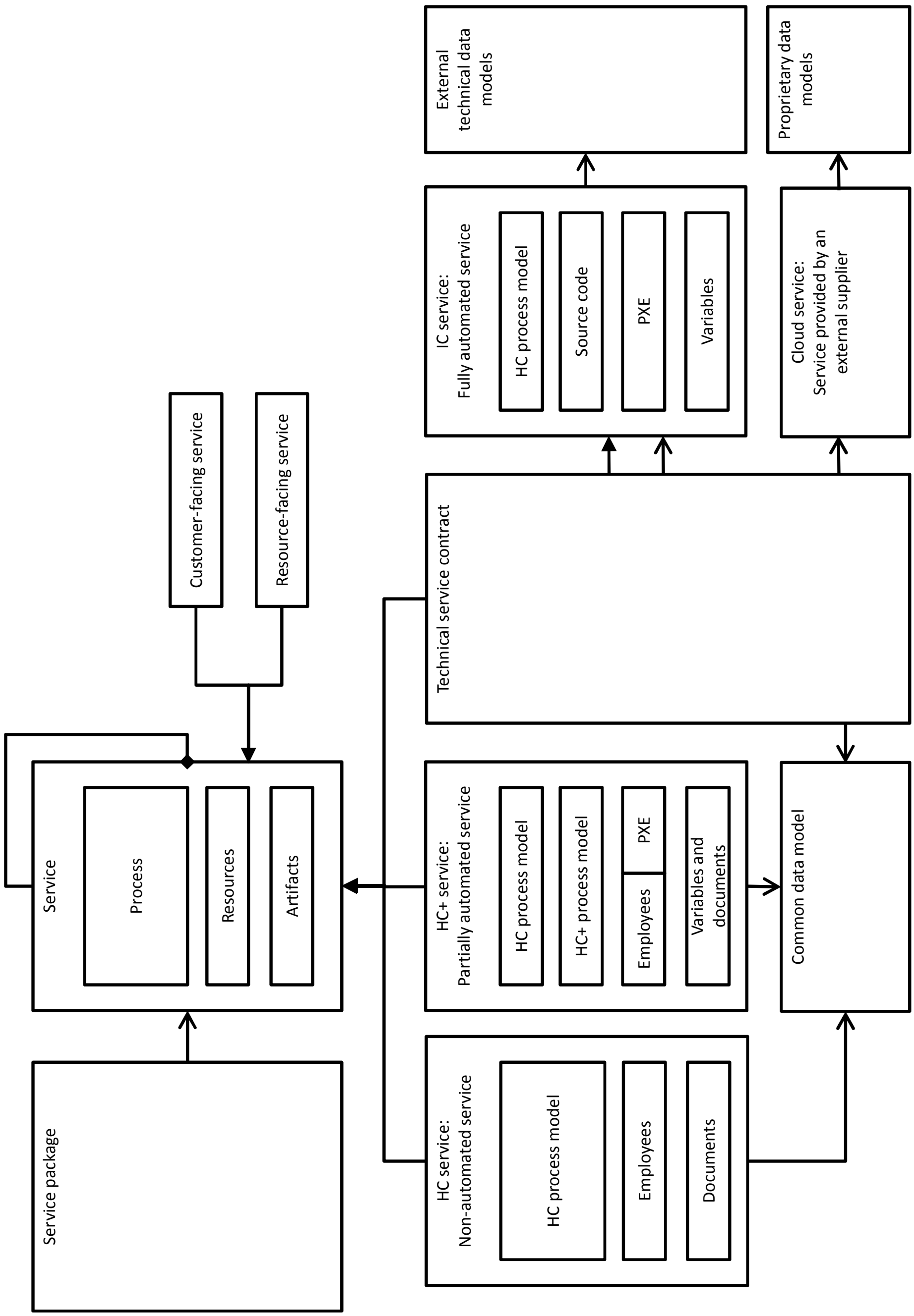



**Fig. 2.** Process-centric service classification

Services can be grouped by the way they are provided [32]. Fig. 2 contains the four primary classes of service provision.

- *Human-centric services* (HC) are services that are not managed by a single IT system. Typically, electronic or paper-based documents are created or modified using word processors or structured data is processed using process-centric application systems. The service provision tasks are described by a HC process model from a business perspective.
  An example for HC services is repairing stationary desktop computers after an electronic order has been issued.
- *Extended human-centric services* (HC+) are human-centric services which are augmented by the ability for partial automation. In order to achieve this, the business-oriented HC process models are enriched by technical details and therefore the resulting HC+ process models can be executed by the services execution engine (SXE).
  Due to the technical enrichment it is possible to modify structured data like order or customer data by the SXE without human intervention.
  For example, the steps of the above-mentioned repairing service can be orchestrated by the SXE completely. While single tasks like billing can be automated completely (e.g. by a cloud service), the actual physical repairing remains a non-automated task.
- *Integration-centric services* (IC) are fully automated services that are developed by IT experts using classical programming languages. HC process models are only the template for the development of such services.
  Typically, IC services are used for automating singular process fragments of HC and HC+ services. An example for such an IC service is a fully automated billing service which may be invoked by HC or HC+ services.
- *Cloud services* (CS) are a derivate of IC services which are provided by an external partner. As CS are considered to be black boxes with fixed interfaces, they essentially are IC services which are provided externally.
  For example, the above-mentioned billing service may be provided by a company specialized in encashment.

Services can form hierarchies, as services may depend on other services. This is especially true for HC+ services. By including fully automated IC services into the process model of HC+ services, single tasks of HC+ services can be automated by IC services. This process-centric perspective on services allows for an incremental increase of the automation degree of the services and empowers non-technical departments to swiftly integrate or replace cloud services without needing to initiate time-consuming IT projects. Furthermore, changing the HC+ service processes requires no advanced IT knowledge. This enables businesses to respond with more flexibility to new challenges when compared to the prevalent IT-centric approach using classical programming languages.

Providing an adequate automation degree is a core task of ITSM. Typically, two service classes are prevalent: ITSM suits like BPMC Remedy offer an instrument for automating a large number of IC services. Less frequent or knowledge-intensive services typically are the domain of HC services. The introduction of HC+ services suggests a new solution for services which cannot be economically automated with IC services due to frequent changes the service process or high flexibility requirements during the service execution.

Another key benefit of HC+ services is the possibility of swiftly introducing cloud services into process-centric services and thereby combining multiple external services to a new value-adding service.

The smooth cooperation of internal IC services and external cloud services requires a compatible data model. This can be achieved with comparably little effort within a company. Existing standards like the TM Forum Information Framework (SID) [33] provide a starting point for defining such a data model. As these standards are not commonly used in today's businesses, intermediaries for translating between the internal and external data models have to be introduced. In many cases, such intermediaries are also required for internal services that are provided by ERP systems which rely on proprietary data models.

This intermediary task is fulfilled by *technical service contracts* (TSC). TSC are service wrappers that map external or proprietary external data models to the company standard. This transformation ensures that the technical aspects of the external data models are hidden from the business-focused perspective of the HC+ service process models. Thus, less technical details have to be considered when integrating cloud services.

### 4.2 Service Process Design

With processes being the central instrument of the process oriented integration of cloud services, the design and improvement of processes is the most important aspect of designing and improving services. Modeling the services' processes allows business users to change and adapt partially automated HC+ services and integrate cloud services without having to start an IT project.

For this purpose, a multi-level procedure model derived from the model-driven architecture (MDA) is being used. The key idea of the MDA approach is to hide the complexity of the technical implementation from business users by utilizing successive model transformations that add technical detail in each step. MDA distinguishes three model types [40]: (1) Computation-independent models (*CIM*) contain requirements from the domain of a software system, but no technical details. (2) Platform-independent models (*PIM*) detail the system and its architecture, yet contain no information about the underlying platform. (3) Platform-specific models (*PSM*) enrich the PIM with information about the underlying platform and can be transformed to source code or source code skeletons.

For HC+ services, a HC level process model representing a CIM is developed and then refined to a HC+ level process. HC+ level process models are not tied to a specific platform and therefore are classified as PIM. As the HC+ level process models contain all information in order to be executable, it can be used to directly generate source code and compile it to a package that can be loaded and invoked by the SXE. As there currently is only one SXE, it is not necessary to create a PSM.

With IC services being developed using classical software development tools, this development is not part of the PCA-C approach. However, as HC+ services may invoke IC services, the PCA-C procedure model for the development of HC+ services details the coordination of service development on the HC+ and IC levels.

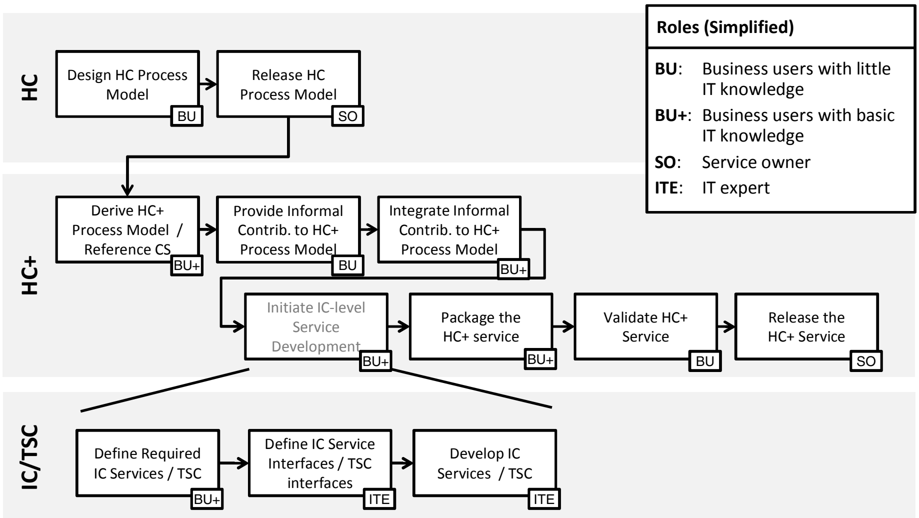



**Fig. 3.** Procedure model for developing and adapting a HC+ service process (based on [1, 34])

Fig. 3 summarizes the steps required for the development and adaptation of a HC+ service process.

First, a HC level service process model is designed by business users with little IT knowledge (BU). For this purpose, there exist two interaction types: (1) Informal contributions like post-it style comments in the process model or postings in the discussion board allow business users to provide suggestions to the process model. (2) A web-based editor allows BU with more experience in process modeling to transform informal contributions into graphical process models. If the service owner (SO) considers new HC process model as acceptable, he or she releases this model.

Business users with basic IT knowledge (BU+) extend a copy of the newly created HC model and extend it by adding technical information that is required for execution by the SXE. This HC+ process model references externally provided CS and / or internally provided IC services. As the HC+ meta model is a superset of the HC meta model and both modeling levels share the same notation, BU can understand this HC+ process models that are derived from HC process models. This passive understanding of the derived HC+ process models is further assisted by the fact that the HC and HC+ process models share a similar structure. By using the informal instruments mentioned before, BU can provide informal feedback to the HC+ process model which the BU+ will integrate later on.

For complex scenarios, it is necessary to develop IC services which provide complex functionality that cannot be implemented economically or efficiently on the HC+ level. If new cloud services are to be integrated, the corresponding TSC typically have to be developed in order to make the CS accessible from the HC+

level. A key objective the TSC is to map the proprietary data model of the CS to the companies' standard data model.

In these scenarios, the IC service or TSC requirements are first defined by the BU+. In the next step, the corresponding interfaces are proposed by the IT experts (ITE). Finally, the ITE develop the underlying IC services or TSC.

Once the HC+ process model and – if necessary – the supporting IC services and / or TSC have been finalized, the HC+ service package is created. This package contains all necessary components in order to be executable by the SXE.

Before finally releasing this HC+ service, the service is validated by the BU. This ensures that the new HC+ service meets its requirements. For this purpose, the execution platform has to provide a separate validation environment that allows executing HC+ services without changing the operational data.

## 4.3 Prototype

The PCA-C prototype comprises three major components: (1) The technical foundation of the TANGO platform, (2) a modeling environment for HC services, and (3) a modeling environment for HC+ services.

With TANGO realizing a classical MDA approach [2], it consists of modeling, development, and execution environments. In order to ensure that the new services do not have to replicate elementary functionality, TANGO already provides a large number of commonly used services. This functional architecture forms the technical backbone both of TANGO and PCA-C.

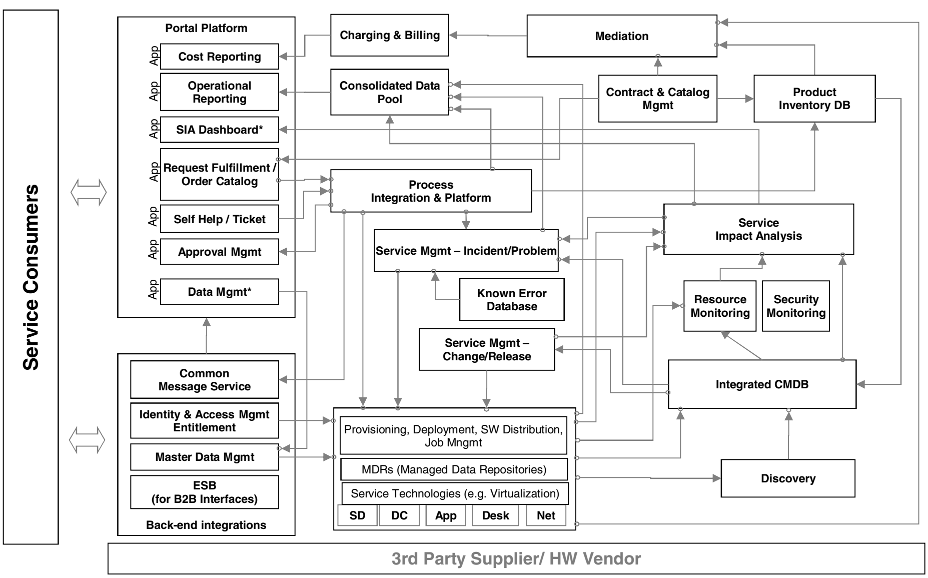



**Fig. 4.** Functional architecture of the underlying TANGO platform

Fig. 4 shows the functional architecture of the TANGO platform on which PCA-C relies as well. With ITIL [5] being the methodical foundation of IT operations at Atos, the platform has to provide typical ITSM functions for *contract*, *service*, *incident*, and *service management*. The integrated *configuration management database* (CMDB) serves as a data warehouse of all configuration information that is relevant to TANGO. An important part of the TANGO platform are the two types of interfaces to the customer: (1) The *portal platform* provides a web application that provides the user interfaces for starting and executing services. (2) The back-end integration allows integrating the customer IT systems on a technical level.

While the TANGO functional architecture is the mature and well-used foundation, the HC and HC+ service modeling environments currently are academic prototypes based on Activiti [36] and the portal solution Liferay Portal [37]. Fig. 5 shows the modeling environment for HC services. This environment primarily consists of a process modeling component for HC level processes. Post-it style comments and a discussion board allow BU without methodical BPM knowledge to contribute their operational expertise. The services designed in the HC modeling environment can be swiftly copied to the HC+ modeling environment. Here, the HC level process model is enriched by technical information like invoked services, branching conditions, role mapping, and forms. In typical cases, the service models can be deployed with a mouse click to the execution environment and executed instantly.

Functions for a managing data types centrally and mapping proprietary data models to the company standard are not yet supported within the prototype.

## 5 Potentials and Challenges

The proposed approach addresses the requirements laid out in section 2:

1. It provides business users like business-oriented consultants with a web-based software as well as a methodology to directly contribute to the automation of internal services and the integration of cloud services Thereby, operational departments can exercise more control over the way their services are automated and swiftly adapt existing services or develop new ones.
2. By wrapping the technical complexity of CS with the help of TSC, business users are not confronted with unnecessary technical details. Once a TSC has been developed for a CS, this CS can be easily used without having to involve IT experts. On the other hand, using the classical software development on the IC level as a fallback mechanism ensures that the approach can cope with complex scenarios as well.
3. The PCA-C approach builds upon the service-oriented architecture of TANGO and therefore allows for leveraging the building block principle. Besides IC services, CS and TSC, the HC+ services orchestrating other services are reusable services as well.

All in all, the approach promises to fulfill the previously defined requirements. However, the approach will have to cope with a number of issues.

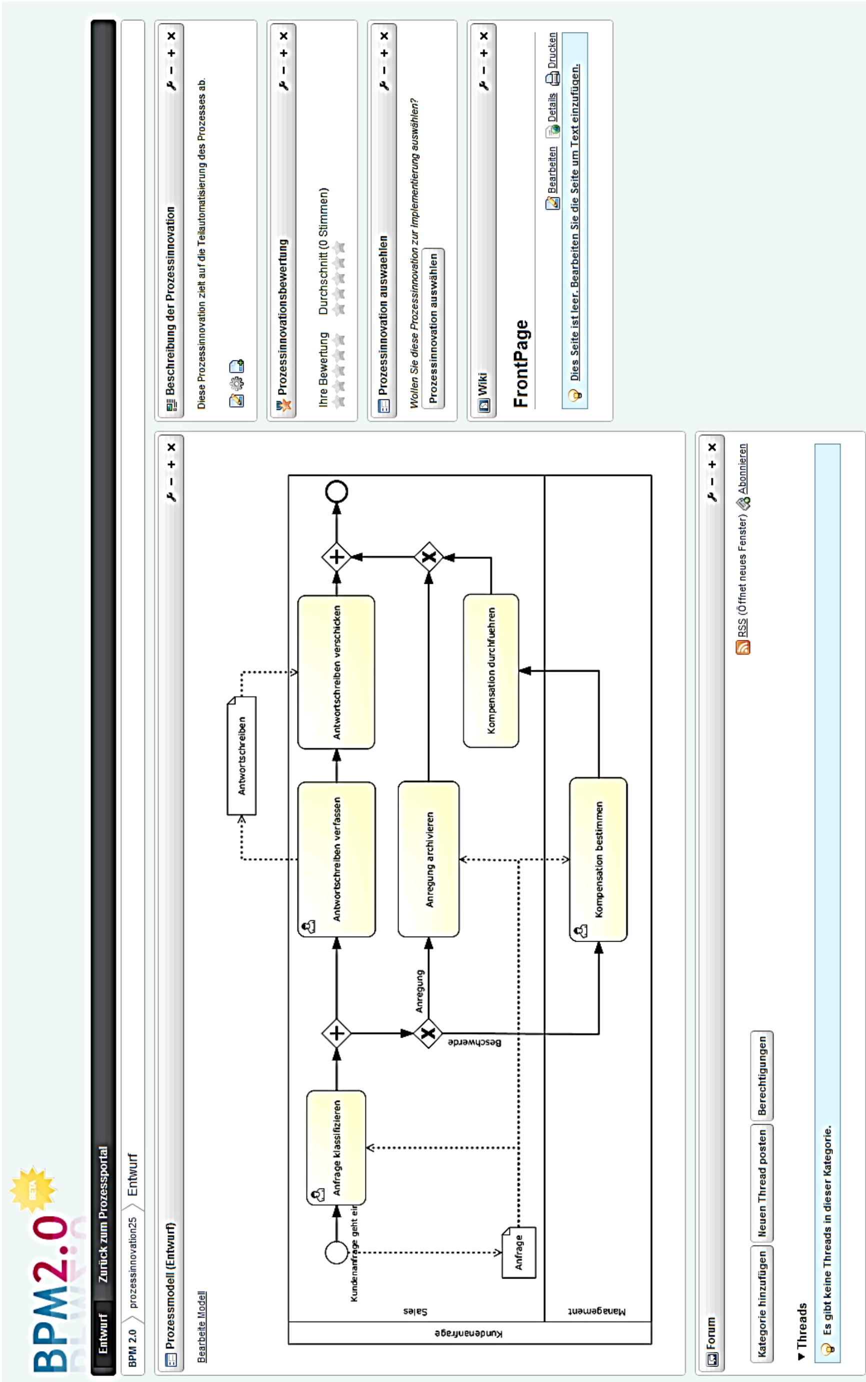



**Fig. 5.** Modeling environment for HC services [38]

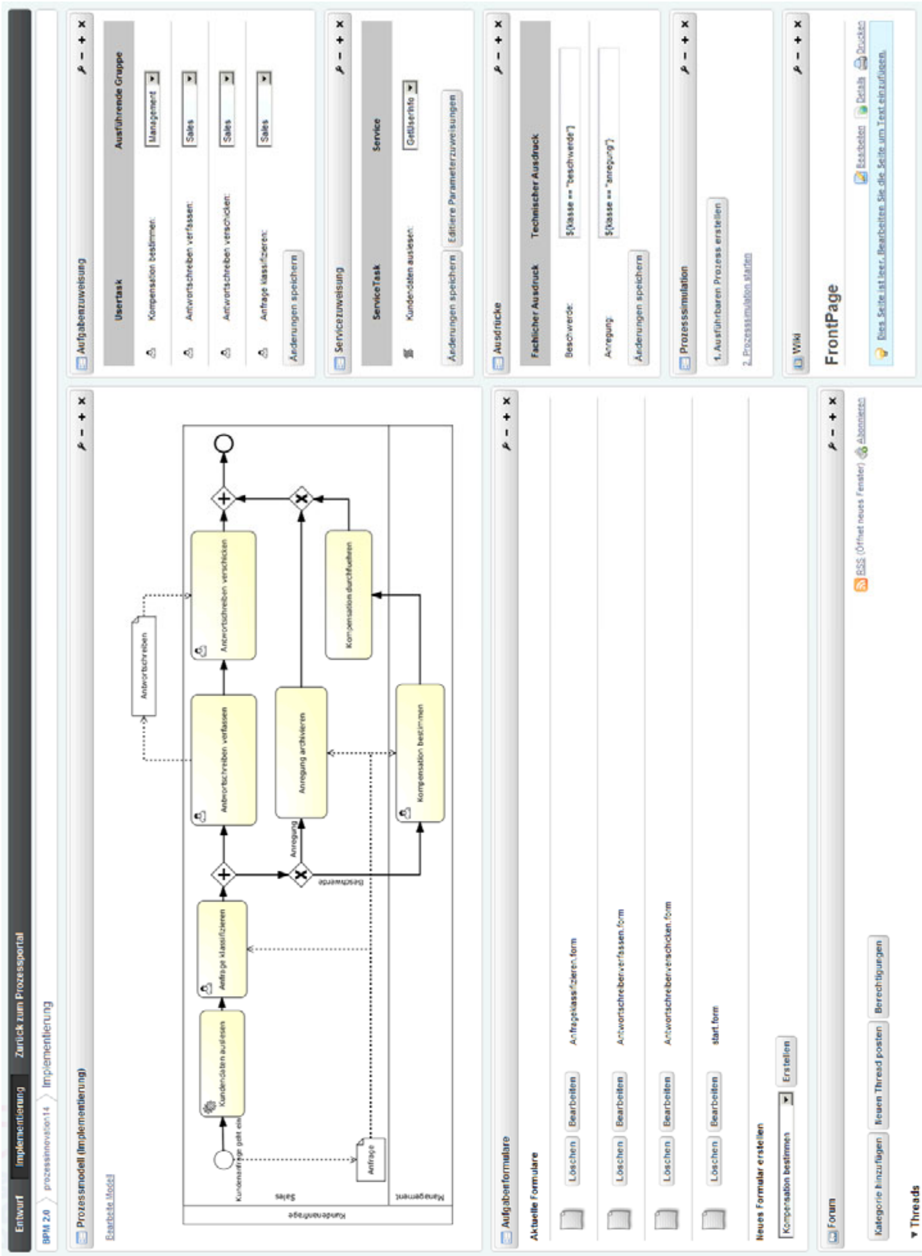



**Fig. 6.** Modeling environment von HC+ services [38]

Currently, the development of TSC requires the consultation of IT experts and therefore limits the degree of autonomy business users have over the automation of their services. However, on the long run, standardized data models like the TM Forum Information Framework (also known as TM Forum SID) can be used for standardizing the data models across multiple service providers.

The prototype will have to be substantially extended before conducting case studies for checking the feasibility of PCA-C in real-world projects. The current version only serves as an instrument for demonstrating the concept.

Experience shows that the strongest obstacles are not at the technical but at the human level. Future users will have to develop the willingness to take responsibility for the automation of their services. However, such responsibility is frequently met with skepticism by employees who primarily focus on completing work instead of planning it [39]. The fear of becoming dispensable contributes further to this unwillingness.

## 6 Related Work

According to [43], incompatible service interfaces and data models are the key challenge for integrating CS. [43] summarizes the approaches for overcoming this issue: (1) Introducing an intermediate layer between internal and external services allows wrapping the actual interfaces of the CS and provide a unified interface. In PCA-C, the TSC fulfill this function. Using standards (2) for interfaces as well as (3) the data models simplifies integrating services significantly. As there currently is no widely used standard, this is primarily a long-term vision.

Already available commercial products like WebSphere Cast Iron Cloud Integration (WSCICI) [41] or Pervasive Data Integrator (PDI) take an approach to integrating cloud services in a process-oriented way similar to PCA-C. However, while the focus on PCA-C is on defining complete and (partially) automated business processes, WSCICI and PDI use technically focused processes to transform data so it can be exchanged between CS and internal systems.

In order to simplify this transformation, WSCICI provides templates for integrating external applications. This works well for use cases that are covered by the templates. Essentially, these templates serve a similar function like TSC, but omit to translate between different data models. In order to overcome this, WSCICI allows data models to be mapped graphically on the attribute level within the integration process. While this is a more intuitive approach than developing IC-level TSC, this approach cannot hide the technical complexity of the proprietary data models (like technical identifiers or authorization information).

In the classification of modeling levels (HC, HC+, IC/TSC) of PCA-C, WSCICI and PDI essentially provide solutions for integrating CS on the IC level using an easy-to-understand graphical representation of a technical transformation process while PCA-C relies on HC+ level integrations that comprise less technical detail, as the technical complexity like data model transformation is hidden by TSC. Instead, the HC+ processes describe the service provision processes primarily from a business perspective. Wrapping this complexity is vital for the PCA-C approach as this empowers business users to define the automation of their services as well as the integration of CS.

## 7 Conclusion

This contribution suggests a process-centric approach for integrating new cloud services. By distinguishing between three carefully aligned service levels (HC, HC+, and IC), the PCA-C approach allows business users to directly contribute to the adaptation of the service automation solution and minimizes the involvement of the IT department. With existing solutions targeting IT experts, PCA-C offers a new approach that enables business users to integrate cloud services into their daily work.

Special emphasis is put on empowering business users to include new cloud services in existing services. With PCA-C bringing the concept of managed self-organization to the integration of cloud services, it provides a similar level of flexibility to the management of cloud services as BPM 2.0 provides for BPM.

## References


1. Billing, G., Kurz, M., Hettling, K., von Jouanne-Diedrich, H.: Applying BPM 2.0 in IT centric environments. In: Rosemann, M. (ed.) Case Studies in Business Process Management. Springer, Heidelberg (2011) (accepted paper)
2. Kurz, M., Lederer, M., Billing, G., Hettling, K., von Jouanne-Diedrich, H.: BPMS 2.0. Übertragung der geleiteten Selbstorganisation auf die Prozessautomatisierung. Information Management und Consulting 25, 67–72 (2011)
3. Schreyögg, G.: Organisation. Grundlagen moderner Organisationsgestaltung Gabler, Wiesbaden (2008)
4. Taylor, F.W.: The principles of scientific management. Dover, Mineola (1998)
5. Scheer, A.-W.: ARIS - business process modeling. Springer, Berlin (2000)
6. Silver, B.: BPMN method and style. Cody-Cassidy Pr., Aptos (2009)
7. Neuman, S., Probst, C.W.C.: Continuous Process Management. In: Becker, J. (ed.) Process Management. A Guide for the Design of Business Processes, pp. 233–250. Springer, Berlin (2003)
8. Hammer, M., Champy, J.: Reengineering the corporation. A manifesto for business revolution. Collins Business Essentials, New York (2006)
9. Imai, M.: Gemba Kaizen. A commonsense low-cost approach to management. McGraw-Hill, New York (1997)
10. Deming, W.E.: Out of the crisis. Massachusetts Institute of Technology, Cambridge (1998)
11. Picot, A., Reichwald, R., Wigand, R.T.: Die grenzenlose Unternehmung. Information, Organisation und Management Gabler, Wiesbaden (2001)
12. Schmelzer, H.J., Sesselmann, W. (eds.): Geschäftsprozessmanagement in der Praxis. Kunden zufrieden stellen, Produktivität steigern. Wert erhöhen Hanser, München (2008)
13. Kurz, M.: BPM 2.0. Kollaborative Gestaltung von Geschäftsprozessen. In: Schumann, M., Kolbe, L.M., Breitner, M.H., Frerichs, A. (eds.) Multikonferenz Wirtschaftsinformatik 2010, pp. 729–740. Univ.-Verl. Göttingen, Göttingen (2010)
14. Yates, F.E.: Self-organizing systems. The emergence of order. Plenum Press, New York (1987)

15. Camazine, S., Deneubourg, J.-L., Franks, N.R., Sneyd, J., Theraula, G., Bonabeau, E.: Self-organization in Biological Systems. Princeton Univ. Press, Princeton (2003)
16. Haken, H.: Synergetics: An introduction. Nonequilibrium phase transitions and self-organization in physics, chemistry and biology. Springer, Berlin (1983)
17. Kurz, M.: BPM 2.0. Organisation, Selbstorganisation und Kollaboration im Geschäftsprozessmanagement, `http://www.forflex.de/uploads/AB/forflex-2009-001.pdf`
18. Surowiecki, J.: The wisdom of crowds. Anchor Books, New York (2005)
19. Kotter, J.P., Schlesinger, L.A.: Choosing Strategies in Change. Harvard Business Review 86, 130–139 (2008)
20. Leiba, S., Hardy, C.: Employee Empowerment. A Seductive Misnomer? In: Hardy, C. (ed.) Managing Strategic Action. Mobilizing Change - Concepts, Readings and Cases, pp. 256–271. Sage, London (1994)
21. McAfee, A.P.: Enterprise 2.0. The Dawn of Emergent Collaboration. MIT Sloan Management Review 47, 20–28 (2006)
22. Lai, L.S.L., Turban, E.: Groups Formation and Operations in the Web 2.0 Environment and Social Networks. Group Decision and Negotiation 17, 387–402 (2008)
23. Komus, A., Wauch, F.: Wikimanagement. Was Unternehmen von Social Software und Web 2.0 lernen können Oldenbourg, München (2008)
24. Neumann, G., Erol, S.: From a Social Wiki to a Social Workflow System. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. LNBIP, vol. 17, pp. 698–708. Springer, Heidelberg (2009)
25. Weske, M.: Business Process Management. Concepts, Languages, Architectures. Springer, Heidelberg (2007)
26. Reichwald, R., Möslein, K., Huff, A.S., Kölling, M., Neyer, A.-K.: Service Innovation. Leipzig (2008)
27. Buyya, R., Yeo, C.S., Venugopal, S.: Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. In: 2008 10th IEEE International Conference on High Performance Computing and Communications, pp. 5–13. IEEE (2008)
28. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A.S.I., Zaharia, M.: Above the Clouds. A Berkeley View of Cloud Computing (2009)
29. Carr, N.: IT Doesn’t Matter. Harvard Business Review 81 (2003)
30. Badger, L., Grance, T.P.-C.R., Voas, J.: DRAFT Cloud Computing Synopsis and Recommendations (2011)
31. Lacy, S., Macfarlane, I.: ITIL Service Transition. ITIL® (IT Infrastructure Library) The Stationery Office, Norwich (2007)
32. Kurz, M., Duschinger, S.: BPMS 2.0. Übertragung der Selbstorganisation auf die Automatisierung von Geschäftsprozessen. In: Bartmann, D., Bodendorf, F., Ferstl, O.K., Sinz, E.J. (eds.) Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse, pp. 217–240. University of Bamberg Press, Bamberg (2011)
33. tmforum: Information Framework (SID), `http://www.tmforum.org/InformationFramework/1684/home.html`
34. Kurz, M., Lederer, M., Billing, G., Hettling, K., von Jouanne-Diedrich, H.: BPMS 2.0. Übertragung der geleiteten Selbstorganisation auf die Prozessautomatisierung. In: IM Die Fachzeitschrift für Information Management und Consulting (2011)

35. Kurz, M., Schaller, T., Reichelt, D., Ferschl, M.: Project4Sure X: Accelerating Implementation Projects for the Microsoft Dynamics Suite with BPM 2.0. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 139–154. Springer, Heidelberg (2011)
36. Activiti: Activiti BPM Platform, http://www.activiti.org/
37. Liferay: Enterprise open source portal and collaboration software, http://www.liferay.com
38. Kurz, M., Duschinger, S.: BPMS 2.0. Übertragung der Selbstorganisation auf die Automatisierung von Geschäftsprozessen. In: Bartmann, D., Bodendorf, F., Ferstl, O.K., Sinz, E.J. (eds.) Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse, pp. 217–240. University of Bamberg Press, Bamberg (2011)
39. Randolph, A.W.: Re-thinking Empowerment. Why Is It So Hard to Achieve? Organizational Dynamics 29(2), S. 94–S. 107 (2000)
40. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1 (2003), http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf
41. IBM: WebSphere Cast Iron Cloud Integration (2011), http://www-01.ibm.com/software/integration/cast-iron-cloud-integration/
42. Pervasive: Pervasive Date Integrator (2011), http://www.pervasiveintegration.com/products/Pages/data_integrator_data_integration_software.aspx
43. Dillon, T., Wu, C., Chang, E.: Cloud Computing: Issues and Challenges. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 27–33. IEEE (2010)

# Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes

Dominik Wachholder and Stefan Oppl

Johannes Kepler University Linz
Department of Business Information Systems - Communications Engineering
Freistaedterstrasse 315, 4040 Linz, Austria
{dominik.wachholder,stefan.oppl}@jku.at

**Abstract.** Subject-oriented business process management (S-BPM) constitutes a new approach that focuses on acting subjects rather than events or data. Consequently, elicitation of business process knowledge can occur in cooperative settings, driven and directly conducted by involved individuals. This specific setting not only is in need of a collaborative modeling environment, but also requires tool support in order to allow people to focus on the work processes to be represented. This paper presents an approach for the collaborative modeling of subject-oriented business processes with the aid of an interactive, distributed platform and introduces concepts for information awareness and tool supported development of cooperative work aspects for effectively supporting modelers. A report on the conducted exploratory user study to elicitate user requirements illustrates potentials for further usability and user acceptance enhancements as well as extensions towards the modeling functionality of the tool set.

**Keywords:** collaboration, subject orientation, business process modeling, knowledge elicitation, tabletop interface, distributed interaction.

## 1 Motivation

Work environments involving a group of people require specific cooperation attitudes [26], especially when facing more complex situations and problems [27]. Decided, explicit alignment interactions are vital to successfully accomplish a common understanding on activities and work processes. The perception of each involved individual impacts the identification as well as the final execution of such interactions [11]. The theory of mental models [14] provides a conceptual approach to describe and reflect upon individual perceptions of work, two activities that are considered necessary for the development of a common understanding [16]. Individuals' actions triggered by the involvement in business processes can therefore be explained by the means of mental models [14]. Research in this area has identified a set of methods that aid externalization [13] and alignment of mental models [2]. Diagrammatic models, such as structuring techniques and concept mapping, have shown the ability to support the externalization of

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 145–162, 2012.

mental models and accelerate reflection and communication processes [13]. Once stakeholders are involved, the externalization is necessary to establish a common understanding to be able to communicate people's perspectives and share information [20].

The modeling tool set *Comprehand* picks up these approaches and facilitates a modeling environment designed for developing collaborative cooperation interactions represented by diagrammatic models [19]. Physically placeable elements allow for the collaborative creation of models on a computer-augmented interactive tabletop modeling surface and enable simultaneous alterations additionally supported by the tabletop interface [23]. The combination of the tabletop interface with a semantically open modeling language possesses a flexible environment for developing conceptual models from the users' viewpoints [18].

Semantically open conceptual models allow involved people to externalize their perception of work in their language. Such models do not force individuals to adapt to a pre-specified notation, thus preventing an additional mapping step in the course of externalization [9]. Additionally, semantically open notations allow to not only capture the model of work itself, they can also capture the domain concepts that people use to describe their work. A different vocabulary becomes evident and can be resolved in this way [25].

Conceptual models, however, cannot only be used to aid communication about work in the course of alignment of work processes. They can also be used to configure interactive systems that support the alignment process (e.g. via in-place simulation and validation) and the actual work process itself (e.g. via a workflow engine) [1]. Although these usage scenarios also support cooperation, they require a more formalized way of representing the conceptual models in order to make them interpretable for computers [3]. Dori (ibid.) has termed this situation as the "apparent human-machine language orientation dilemma".

Supporting the process of developing formalized models from less structured, semantically open models is a recognized research issue. Different approaches to solve this problem have been examined, ranging from explicitly representing vagueness in models [12] to providing a unified notation for informal and formal models [28] [7]. These approaches, however, focus on language properties rather than providing methodological support for the externalization of more formalized models. By extending the Comprehand approach, the research presented in this paper aims at methodologically bridging the gap between the way people describe their work and the formal models necessary to interactively support these work processes.

Subject-oriented business process management (*S-BPM*) [4] is an approach for modeling business processes from a stakeholder perspective. It explicitly distinguishes between one's individual work (internal behavior) and the communication (interaction behavior) among involved people that is required to successfully accomplish a process. This separation of concerns reasonably supports the elicitation of distributed work process knowledge [20] and allows for a step-by-step integration of different views on cooperative work processes. Tools for direct simulation, validation, and enactment of created models are available

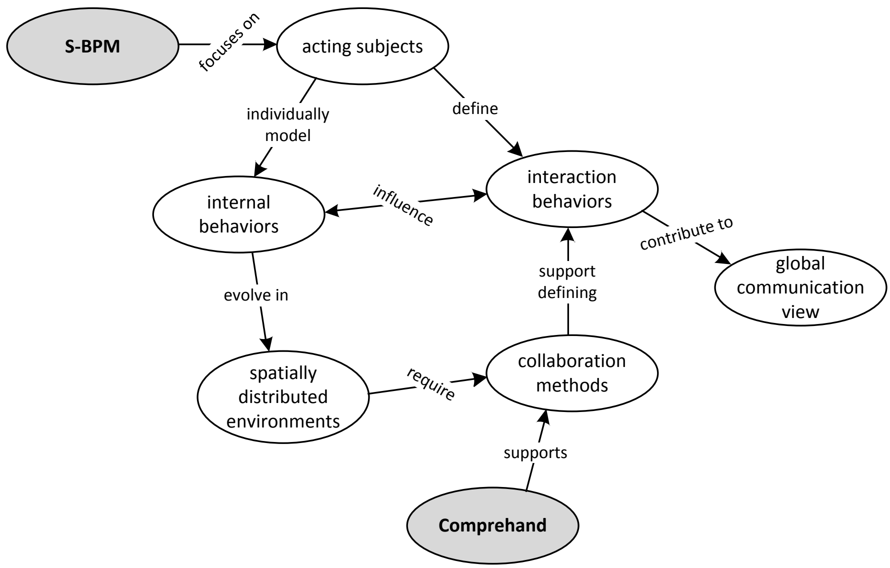



**Fig. 1.** Interrelation of S-BPM and Comprehand

and enable interactive support of the modeling and work processes. Figure 1 outlines the coherence between S-BPM and Comprehand to gain a better understanding of how the separate parts are linked together.

S-BPM focuses on active organizational elements, the so-called subjects (cf. Figure 1), rather than focusing on events or the data being processed [5, p. 85-86]. Subjects constitute the source of every action and interaction, and are responsible for modeling their internal behaviors as well as for defining their interaction behaviors. Since they are not bound to any specific geographic locations, the modeling processes can take place in spatially distributed settings. Due to this fact, tool support is required supporting users while collaboratively eliciting and modeling distributed process knowledge. Comprehand can therefore be used to provide a tool supported modeling environment enabling the usage of collaborative methods.

The goal of this paper is to integrate S-BPM and Comprehand in a way that allows to capture interaction behaviors and integrates individually modeled internal behaviors in a way comprehensible to users inexperienced in (subject-oriented) modeling. Once the interaction behaviors are well-defined, the outcome is a clearly arranged view on the global communication throughout the entire process, ultimately leading to a model that can be validated and executed without further need for augmentation. Working hypothesis in this paper particularly focus on tool acceptance and usability issues of the existing prototype. Supporting users with a basic set of modeling functions especially in terms of S-BPM, practical experiments and tests are intended to identify effects concerning both the user's modeling process as well as the outcome and constitute the basis for the deduction of continuative implementations and research topics.

In the following section, we are elaborating on our basic approach to collaborative elicitation of business process models. Requirements on information awareness in spatially distributed modeling setting are given in the third section. We then briefly describe the necessary technical infrastructure and give an account of the modeling process and according tools support. In the final sections we report on our experiences during the evaluation of the prototype and discuss potential methodological and technical improvements and extensions of the instrument.

## 2 Collaborative Modeling of Subject-Oriented Business Processes

The concept of subject-oriented business process modeling is focused on the communication among subjects. The purpose of this communication-oriented interpretation of business models is to identify and reflect upon existing interaction patterns. A set of defined modeling rules based on natural language constructs supports modelers throughout the modeling process. This establishes a more familiar and convenient modeling environment by imitating an ordinary communication flow [6].

From another perspective, this idea can not only be used for the representation of interaction patterns, but can also effect the way of how business process knowledge is elicited. The collaborative modeling of subject-oriented business processes therefore refers to a specific setting where two or more spatially distributed subjects corporately try to create one model. Even though all involved subjects take part in the same model creation process, they individually design their internal behavior and only publish their interfaces with which they communicate. The internal behaviors are thus not visible to others. By abstracting over available internal behaviors, the global communication view can be extracted that in turn fosters a common process understanding.

### 2.1 Internal Behavior

Internal behavior models illustrate the subjects' contributions to the process irrespective of other subjects' attitudes. It describes the exclusive sequence of actions that is required to successfully accomplish a given task. Function states, send states, and receive states are the constructs that enable the representation of the perceived work situations. Send and receive states here allow modeling of interaction behavior by either defining or using messaging interfaces to or from other subjects.

Figure 2 generically exemplifies a subject's internal behavior using the available elements mentioned above. This simple scenario outlines a message that is sent by one subject and which is addressed to another involved subject within

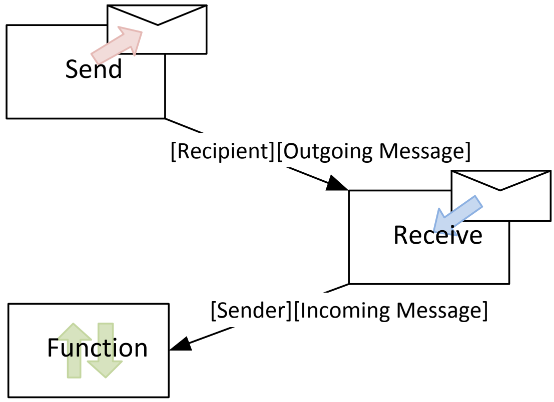



**Fig. 2.** Internal Behavior

the same modeling session. The message itself is assigned to the edge that interconnects this particular sending state with the subsequent state in the model. The same principle can be applied in case of an receiving state. The subject is waiting for an incoming message that is once again assigned to the edge interconnecting the receiving and subsequent state in the model. Once the message is received the subject can move on to the function state and execute the required individual actions.

### 2.2 Interaction Behavior

Interaction behavior models encapsulate internal behaviors of subjects and outline how subjects interrelate and communicate with each other. The level of detail regarding the individual sequence of actions decreases and a special focus is drawn on messages that are exchanged among subjects. The communication flow among subjects is taking center stage.

The introduced approach benefits from the subject's interaction behavior that also exhibits the key feature of the collaborative modeling of subject-oriented business processes. Subjects not only define their internal behavior but also think about existing communication patterns that have to be in place to fulfill a certain task and which in turn link them to other subjects. While defining the interfaces that link them together, real world process knowledge is elicited and simultaneously pictured within a model not requiring any special cooperation actions on the part of the users.

Figure 3 shows the encapsulation of the internal behaviors and totally focuses on the interfaces that are established. As you can see in the picture, *Subject 1* and *Subject 2* are not aware of other internal behaviors and are therefore only communicating through the available message interfaces.

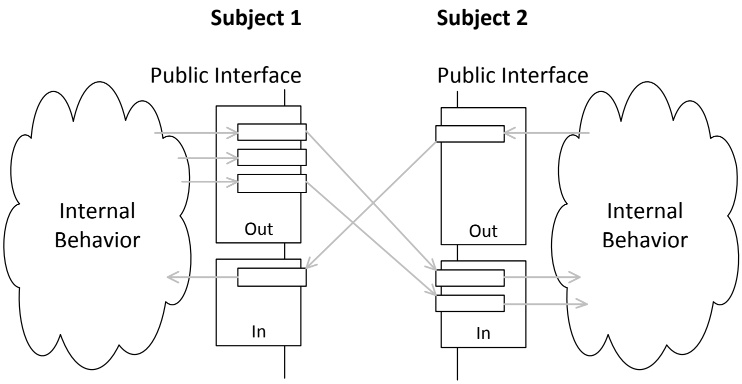



**Fig. 3.** Interaction Behavior

# 3 Information Awareness

Applying the Comprehand approach to S-BPM, each subject is represented by a separate interactive tabletop modeling surface. Those surfaces can be co-located or spatially distributed and even be located on remote sites. In either case, special attention has to be drawn on information awareness in order to recognize or prevent potential problems or inconsistencies within the model and keep users up-to-date throughout the entire modeling session concerning other users' actions and behaviors. Concerning this issue, Nacente et al. [17] introduce the concept of *embodiments* to both give users feedback about their own actions and convey the awareness of other users' involvement with respect to characteristics such as the presence, location, and movement [10]. They further draw a distinction between two specificities named *virtual* and *physical embodiments* where, on the one hand, the users' bodies manipulate objects directly and, on the other hand, virtual representations of involved users serve as placeholders and impart information. Both approaches therefore focus on the representation of people in groupware systems.

It is crucial to provide modelers with information regarding the current state and occurring changes of the model in an unobtrusive way. Missing or misleading model information might result in negative effects on the overall usability of the tool and would have to be communicated to users explicitly.

## 3.1 Appropriation of Spatially Distributed Information

The collaborative modeling of subject-oriented business processes within a spatially distributed setting triggers the generation of a variety of information. In most cases, the generated information is needed at the point of origin as well as by other involved subjects.

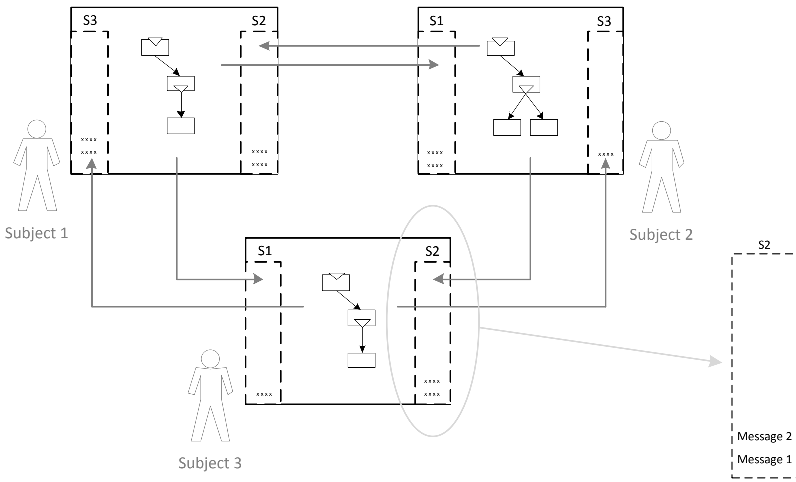



**Fig. 4.** Notification Trays

Due to the distributed environment and since the exchange of messages is a key feature of S-BPM, Comprehand is in need of a user interface that facilitates a mechanism to accurately notify users of incoming messages. The concept of *virtual embodiments* [17] is applied due to the lack of direct contact to other subjects in the modeling session. The implemented *Notification trays* serve this purpose and provide virtual connections among available modeling environments. As illustrated in Figure 4, every subject is interconnected to all other subjects via a notification tray. In other words, every available subject within the modeling session is represented through a dedicated notification tray on the tabletop interface.

Once a message is received from a subject, meaning that another subject has defined a message by using a sending state element, the message appears within the designated notification tray. For instance, if *Subject 2* defines and addresses a message to *Subject 1*, the message is displayed in the notification tray *S2* on the tabletop interface of *Subject 1*. Consequently, the modeler can respond to this event by assigning the message to a receiving state element in his or her own model (internal behavior).

### 3.2 Supporting the Construction of Consistent Models

S-BPM defines a set of rules that have to be adhered to establish a consistent process model that can be validated and executed. Any rule violations conclude in a non-executable model and require iterative adaptions until the model is completely corresponding with the specification of S-BPM.

Comprehand provides an error and information message concept that guides users through the fault correction process once inconsistencies have been identified in the model. Error messages are displayed on demand directly on the tabletop interface and are context-sensitively adapted to the present situation. Conspicuous colors and flashing message box borders are used to additionally attract the user's attention to make him or her aware of apparent errors in the model.

The tool enables following inconsistency prevention mechanisms in its current implementation:

1. Messages can only be assigned to edges if certain rules are adhered. These rules ensure that the point of origin represents a sending or receiving state element and that the edge on which the message should be attached is directed and outgoing.
2. Once a message is defined and attached to an edge interconnecting a sending state element, the system checks whether other subjects already demand the message's availability. In this case, the sending subject is not allowed to detach or redefine the message until all other attachments have been deleted.

## 4 Architecture and Infrastructure

The Comprehand system [18] provides the technical foundations for the addressed extensions that are described in section 2. The open source framework *ReacTIVision* [15] is responsible for the optical recognition of codes in real-time and controls the input stream of elements used on the tabletop surface and precisely evaluates the elements' positions as well as their rotations [22]. The tabletop interface combined with a projector simultaneously serves as an output channel and enables a coherent information output that is used to show the system's reaction towards the model [22].

Compared to the original infrastructure described in [18], some adaptations were implemented to effectively support the collaborative modeling environment. Figure 5 outlines the system's infrastructure including two spatially distributed subjects represented by two separate modeling environments (Comprehand tabletop interfaces). As defined in the concept of S-BPM but not explicitly shown in the figure, these subjects could also constitute computer systems, databases, etc as well.

Due to the spatial distribution of the modeling platforms, network cabling and appropriate network elements are necessary to enable a network-based communication via the Ethernet protocol. Either a local network or the access to the dedicated server over the internet are adequate implementations to meet these requirements.

A dedicated server supporting the Extensible Messaging and Presence Protocol (XMPP) [24] performs communication management among subjects and represents the central point in the network. All communication as well as control messages are handled by this particular server. The messages are therefore not only used to provide the collaborative modeling functionality, but are also available to serve as foundation for further modeling process analysis and model

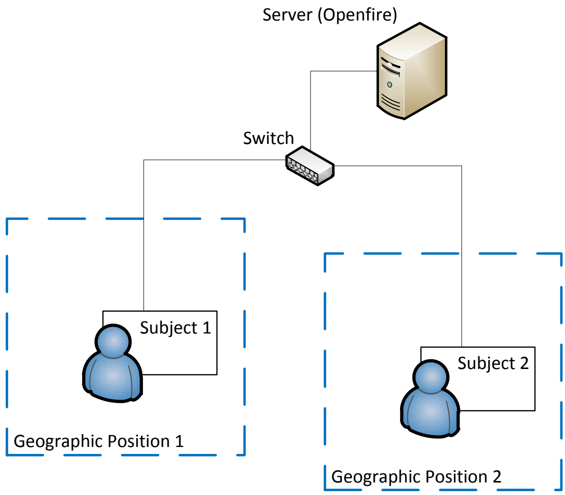



**Fig. 5.** Network Diagram

reconstruction issues. The XMPP protocol itself provides a flexible, scaleable, and extendable framework for the implementation of numerous communication channels, such as text messaging, audio streaming, and video streaming. Based on Matthias Freudenberger's research results, the *Openfire* implementation in combination with the *SmackAPI* is used throughout the project [8].

## 5 Modeling Process

This chapter consolidates the theoretical concept introduced in the previous chapters with *Comprehand* and gives a hands-on approach on working with this tool. The existing features as well as the newly introduced concepts and implemented tools, such as the *Message Tool*, are explained in the following sections. It aims at giving an practical overview of how the tool can auxiliary support the user's modeling experience and possibly can impact the outcome.

### 5.1 Constructing the Model of Individual Behavior

The tangible modeling elements constitute the foundation of business processes that are modeled with *Comprehand*. The users therefore have to place the desired elements on the tabletop interface. The system automatically identifies the chosen elements with their exact position and rotation and registers them in the data model. The position and rotation of each element can simply be updated through moving the element to the desired position.

Elements on the surface can be named at any given point in time. Once the particular element is selected, users can open the input field by pressing the blank key on the keyboard. Hitting the enter key closes the input field, confirms the entered text, and displays it directly next to the element.

Users can interconnect elements by temporarily placing them in close distance. As soon as the minimum distance between two elements falls below a defined value, the system interconnects these two elements by projecting a virtual line starting from one element to the other. As long as the concerned elements are within the range of recognition, the system will identify them as connected regardless of the elements' later positions and rotations. The connection can only be deleted by using a dedicated *Eraser Tool*.

Since the S-BPM approach requires a defined flow of communication, the tool has to support the definition of directed edges. By using the *Arrow Tool*, already established connections between two elements can be directed by directly placing it on the connection.

### 5.2 Establishing the Interaction Behavior

In addition to modeling the internal behavior, *Comprehand* also supports modeling the interaction behavior in terms of determining the communication interface. A *Message Tool* has been introduced to enable users to easily handle incoming as well as outgoing messages. The context sensitive tool shows the appropriate menu once it is used and assigned to a link interconnecting a sending or receiving state element with another element. Irrespective of the context in which it is used, the handling is always the same and happens in a consistent way. In doing so, available items are display around the tool and can be selected by turning the tool until the desired segment is highlighted. Once the tool is removed, the highlighted item is selected for further processing tasks. Regardless the context, the tool always shows an empty segment among the available items allowing users to deselect items.

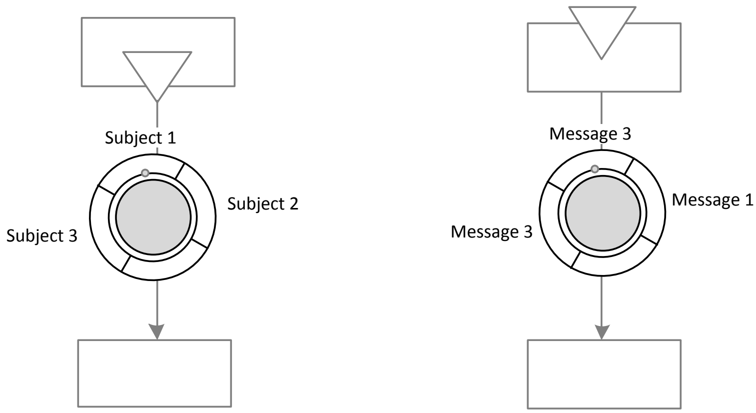
In case of an element representing a sending state, as outlined in Figure 6(a), the message tool displays all available subjects as items. By turning it, the desired subject can be selected and the message can be defined as described in section 5.1 using the input box after removing the message tool from the tabletop surface. Once the message is defined, it is assigned to the edge and the addressed subject is notified immediately.

Using the message tool in the context of a receiving state, as illustrated in Figure 6(b), the message tool displays all available messages as items. The items are an accumulation of all messages displayed within the subjects' notification trays. By turning the tool, the desired message can be selected and is assigned to the edge.

## 6 Exploration Study Design

On the basis of practical experiments and tests, which were conducted at *S-BPM One 2011*[1], several usage patterns have been identified in terms of the usability

[1] Third International Conference on Subject-Oriented Business Process Management, S-BPM One 2011, Ingolstadt, Germany, September 2011.




(a) Sending Context (b) Receiving Context

**Fig. 6.** Message Tool

of *Comprehand*. Potential enhancements are derived based on the exploration results. The modeling scenario as well as the exploration setting used for the study are illustrated in the following section.

### 6.1 Modeling Scenario

Figure 7 illustrates the modeling scenario which was used to conduct the practical experiments and tests. The scenario pictures the vacation application and involves three subjects named *Employee*, *Secretary*, and *CEO*[2].

The employee triggers the entire process by having the desire to go on vacation, and therefore, has to complete the required vacation form. Once the vacation form is completed, the secretary has to be informed by sending the particular form.

As soon as the message is sent, the scenario switches to the secretary who has to handle the vacation request appropriately. After receiving the vacation form, he or she checks for internal conflicts and then forwards the request to the CEO by sending a *vacation request* message. The incoming employee's message obviously differs from the vacation request which is sent to the CEO. Both the name and the data which are associated with the message illustrate the disparity due to the context of communication. Consequently, the CEO decides upon the vacation request and sends back both possible answers represented by

[2] Due to hardware restrictions, the CEO was not present in terms of an independent modeling environment, and therefore, was simulated by a computer. The required messages (confirmation and rejection) were artificially infiltrated.

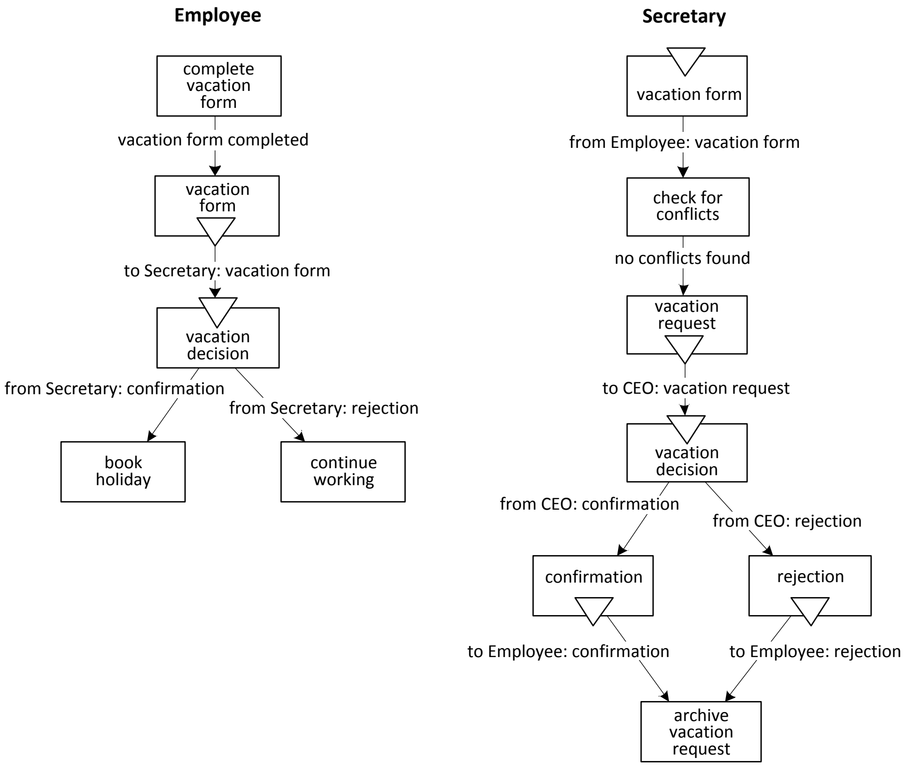



**Fig. 7.** Modeling Scenario

the messages *confirmation* and *rejection*. The secretary's responsibility now is to simply forward the answer to the employee and archive the vacation request for departmental purposes.

Finally, the employee who is waiting for a decision upon his or her vacation request can either book the hotel and go on vacation or continue working depending on the incoming message.

### 6.2 Exploration Setting

Led by the model illustrated in Figure 7, users were asked to independently go through the given example scenario step-by-step using two Comprehand tables located in the same room. The intended course of action was designed to guide the modelers to the usage of all elements and tools available in the context of collaboratively modeling subject-oriented business processes.

During the entire inquiry period, five groups of 2 to 10 people were available for observation. On average, one to three persons per group took an active part in the modeling process, and therefore, were potential candidates for identifying behavior patterns towards the acceptance and usability of the tool.

Discussions throughout the overall modeling process as well as after the sessions in terms of the exchange of experiences constituted the foundation for the determination of prospective enhancements. The most relevant insights that have been identified are described in the following section.

### 6.3 Initial Findings

The implementation of the study described above has provided user feedback and observations on conceptual shortcomings of the current tool support. The following list briefly describes the observed issues and illustrates the underlying indications:

**Inability to choose the correct element type during modeling.** There was indication that modelers have difficulties modeling business processes in a predictive way. For modelers, it appeared that it is hard to choose the right element at the time of use.

**Inexistent tool support for common modeling behaviors.** Observations showed that the modelers unconsciously follow similar ways of modeling business process models. Similarities emerged in the course of the performed scenarios.

**No overview of global process.** There were marginal differences among the various models due to the given example scenario. However, the observed modelers tended to gather process information of other subjects' internal behaviors.

**No tool support for complex process models.** The example scenario was intentionally aligned to the limited modeling area of Comprehand. Even though the predefined scenario, users reached the limit of both available modeling area and code recognition zone by moving the modeling elements or extending the model.

## 7 Resolution of Shortcomings

In the former section, the major conceptual and technological shortcomings of the current implementation of the tool set were identified based on an exploratory user study. This section presents possible solutions to conceptually enhance the usability of Comprehand and discloses continuative research questions.

### 7.1 Generic Elements

The approach presented here is a solution to the issue *Inability to choose the correct element type during modeling*. Comprehand currently supports the modeling of subject-oriented business processes using function, send, and receive state elements as they are defined in the concept of S-BPM. Users that prefer modeling the internal behavior preceding the communication behavior are therefore especially aggrieved since they are forced to anticipate the continuative course of the model. Appropriate elements have to be chosen at the time of their usage. A dynamic role assignment during the modeling process is not possible.

A conceivable approach to enhance the flexibility of creating models would be the introduction of a generic element [21, p. 10]. For the time being, generic elements have no semantic classifications and are aimed at decoupling the prevailing model from the target solution. These elements serve as placeholders and sustain their final role not until the model has evolved sufficiently. In other terms, modelers are free to specify the actual role of elements arbitrarily to any given time. This concept is expected to significantly influence the modeling experience by providing a more flexible and convenient way of modeling. Both modeling extremes, meaning modeling the internal behavior preceding the communication behavior and modeling both behaviors just in time, are then supported equally.

### 7.2 Tool Supported Model Development

The approach presented here is a solution to the issue *Inexistent tool support for common modeling behaviors*. Available information awareness features, described in section 3, guide users through the modeling process but do not exhibit any support by automatically anticipating and applying particular needs derived from the user's displayed modeling behavior.

The conducted user tests and practical experiments have identified following two main attributes which might improve the tool supported development of models with Comprehand:

1. Anticipation of edge directions considering typical modeling behaviors
2. Anticipation of edge titles considering preceding elements

The first item refers to remkarable modeling behaviors that can be used to effectively support users during the modeling process providing context sensitive suggestions regarding the direction of edges. The course of an evolving model underlies such behaviors which are unconsciously followed by users. Observations have shown that models typically possess an intuitive modeling flow from both top to bottom and left to right. Being aware of this particular flow of modeling, the tool can anticipate the desired edge direction and can instantaneously place the appropriate arrowhead. In case of a misplace arrowhead, meaning that the system has wrongly interpreted the desired flow of the model, the user is allowed to toggle the direction using the *Arrow Tool*. Further possibilities of creating inconsistent models in terms of undirected and double directed edges can be obviated by restricting the system to only enable the toggling of directions.

The second item addresses the automatic naming of edges based on preceding elements. Even though this concept can only be applied in case of preceding function state elements, it would simplify the modeling process by decreasing the user's workload. A generic naming algorithm would create the titles by consolidating the name of the preceding function state element with a phrase expressing a finished state. For example, the edge, following a function state named "Complete Form", would be named like "Complete Form Done" or "Form Completed". The implementation of this idea comes along with limitations regarding the use of generic elements mentioned in 7.1. Since the role assignment is performed at a later time by the user, the modeling elements will temporary just serve as a placeholder and cannot be identified immediately to automatically name the following edge. In this case, the naming algorithm cannot be applied. Once an element is determined as a function state and the name is set, the edge can be named automatically as long as it is not set manually.

### 7.3 Common View on Internal Behaviors

The approach presented here is a solution to the issue *No overview of global process*. Once a modeling session is taking place, modelers are responsible for modeling their internal behavior as well as for defining their interfaces which are required for the communication with other subjects. While being focused on the development of their individual model, they do not obtain a common view on neither other subjects' internal behaviors nor the global communication within the entire process. The modelers are therefore very close-minded and focused on their individual point of view on the process. On this account, internal and interaction behaviors are an exclusive result of the own process knowledge and are not influenced through actions caused by other subjects in any way. This might lead to inefficiencies in models caused by subjects trying to achieve local optima rather than trying to contribute to obtain a global optimum.

In this regard, group awareness is considered as an important aspect towards accomplishing a task collaboratively with respect to pursuing the process contributions and activities of others [17]. Nacenta et al. (ibid.) describe following elements that positively impact the information awareness and in turn the overall result:

- Who is working.
- Where they are working (in the task and workspace).
- What they are doing.

Thus, Comprehand has to enable an integrated view on other evolving models (internal behaviors) to foster a common view on the entire process model. The overall goal is to establish an overview of the entire model to accomplish a common understanding and view on activities and processes of all involved subjects. Modelers can then better decide on an appropriate solution of their own behavior according to the realization of an effective and efficient cooperation among all sub-processes.

### 7.4 Tool Support for Complex Process Models

The approach presented here is a solution to the issue *No tool support for complex process models*. Comprehand has to get by with a very limited and restrictive modeling surface in terms of available spatial space. Users quickly encounter restrictions especially when modeling more complex business process models containing numerous elements and edges. There are two potential concepts that are intended to increase the modeling possibilities respective the construction of these models:

1. Temporary removal of unused elements
2. Embedment of sub-models

Item one is targeted to manage the available modeling surface by simply removing the elements that are not taking center stage in the current model state. However, the elements are not removed from the data model and keep persisting until the user removes them explicitly. This concept allows users to easily clean up the modeling surface without losing any model information but also goes along with some negative aspects. Since the elements are associated with unique identification codes and do not have any external labels, it is hard to identify the previously assigned meaning once they have been removed from the surface.

The second concept introduces the embedment of parts of the model into one element. Designated elements therefore contain parts of the existing model but are still present on the modeling surface. An approach for defining an interface with which the collapsed model can be interconnected has to be developed.

Using concepts to enhance the ability to create more complex models can be additionally supported by a monitor showing the current state of the entire model including elements that have been removed or embedded due to the lack of modeling space.

## 8 Discussion

Using the Comprehand approach and tool set to support the collaborative modeling of subject-oriented business processes is a new approach to foster the elicitation of business process knowledge in a stakeholder-driven setting. In its first implementation, the focus has been on providing a technical support platform consolidating methods derived from *Mental Model Alignment* [2] in the context of *Articulation Work* [27], the concept of *S-BPM*, and a spatially distributed collaborative approach to elicit process knowledge. Based on these results, further research can now be conducted to study possible impacts on the elicitation of subject-oriented business process knowledge as well as how this tool can effect learning processes of the S-BPM method itself.

People are capable of externalizing knowledge in terms of diagrammatic, formalized models once they receive appropriate tool support [21, p. 17]. This paper has described the usage of Comprehand in the context to facilitating a working environment in which users do not have to focus or even be aware of modeling language constraints and rules. In fact, it is feasible to almost completely focus on eliciting available process knowledge while the tool set keeps track of the model state and provides modeling scaffolds whenever necessary. Using Comprehand to create an active learning environment for conveying and teaching the S-BPM method is another potential research and application area. The aim here is to identify and support individual learning processes while people are externalizing business process knowledge, using the subject-oriented approach supported by an interactive platform.

Consequently, further research will focus on how Comprehand can help to develop people's subject-oriented business process understanding and externalize their work-specific knowledge even in spatially distributed environments. Research areas will include the identification of positive and negative aspects concerning the outcome in terms of the quality of the business process model and the perceived mapping accuracy between the real world and the model.

**Acknowledgments.** The research leading to these results has received funding from the European Commission within the Marie Curie Industry and Academia Partnerships & Pathways (IAPP) programme under grant agreement n$^o$ 286083 (IANES).

## References

1. Caetano, A., Silva, A.R., Tribolet, J.: Using roles and business objects to model and understand business processes. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1308–1313. ACM Press, New York (2005)
2. Dann, H.-D.: Variation von Lege-Strukturen zur Wissensrepräsentation. In: Scheele, B. (ed.) Struktur-Lege-Verfahren als Dialog-Konsens-Methodik. Ein Zwischenfazit zur Forschungsentwicklung bei der rekonstruktiven Erhebung subjektiver Theorien, pp. 2–41. Aschendorff (1992)
3. Dori, D.: ViSWeb – The Visual Semantic Web: Unifying Human and Machine Knowledge Representations with Object-Process Methodology. VLDB-Journal 13, 120–147 (2004)
4. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement. Hanser Verlag (2011)
5. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010), http://www.springerlink.com/content/m4105554174112q7/
6. Fleischmann, A., Stary, C.: Whom to talk to? a stakeholder perspective on business process development. Universal Access in the Information Society, 1–26, http://dx.doi.org/10.1007/s10209-011-0236-x, doi:10.1007/s10209-011-0236-x
7. Frederiks, P., van der Weide, T.: Information modeling: The process and the required competencies of its participants. Data & Knowledge Engineering 58(1), 4–20 (2006)
8. Freudenberger, M.: Implementierung eines Werkzeuges zur kooperativen, verteilten Erstellung semantisch offener Modelle. Master's thesis, University of Linz (August 2011)
9. Goguen, J.: On Notation. Tech. rep. (1993)
10. Gutwin, C., Greenberg, S.: A descriptive framework of workspace awareness for real-time groupware. In: Computer Supported Cooperative Work (CSCW), vol. 11, pp. 411–446 (2002), http://dx.doi.org/10.1023/A:1021271517844, doi:10.1023/A:1021271517844
11. Herrmann, T., Hoffmann, M., Kunau, G., Loser, K.U.: Modelling Cooperative Work: Chances and Risks of Structuring. In: Cooperative Systems Design, A Challenge of the Mobility Age. Proceedings of COOP 2002, pp. 53–70. IOS Press (2002)
12. Herrmann, T., Loser, K.U.: Vagueness in models of socio-technical systems. Behaviour and Information Technology 18(5), 313–323 (1999)

13. Ifenthaler, D.: Diagnose lernabhängiger Veränderung mentaler Modelle - Entwicklung der SMD-Technologie als methodologisches Verfahren zur relationalen, strukturellen und semantischen Analyse individueller Modellkonstruktionen. Ph.D. thesis, University of Freiburg (2006)
14. Johnson-Laird, P.N.: Mental models in cognitive science. Cognitive Science 4(1), 71–115 (1981)
15. Kaltenbrunner, M., Bencina, R.: reacTIVision: a computer-vision framework for table-based tangible interaction. In: TEI 2007: Proceedings of the 1st International Conference on Tangible and Embedded Interaction, pp. 69–74. ACM Press, New York (2007)
16. Kim, D.: A Framework and Methodology for Linked Individual and Organisational Learning: Applications in TQM and Product Development. Ph.D. thesis, Sloan School of Managment, Massachusetts Institute of Technology (1993)
17. Müller-Tomfelde, C., Nacenta, M.A., Pinelle, D., Gutwin, C., Mandryk, R.: Individual and group support in tabletop interaction techniques. In: Müller-Tomfelde, C. (ed.) Tabletops - Horizontal Interactive Displays. Human-Computer Interaction, pp. 303–333. Springer, London (2010), http://dx.doi.org/10.1007/978-1-84996-113-4_13, doi:10.1007/978-1-84996-113-4_13
18. Oppl, S.: Unterstützung expliziter Articulation Work – Interaktive Externalisierung und Abstimmung mentaler Modelle. Ph.D. thesis, Technical University of Vienna (August 2010)
19. Oppl, S.: Unterstützung expliziter Articulation Work durch Externalisierung von Arbeitswissen. In: Peschl, M.F., Risku, H. (eds.) Kognition und Technologie im kooperativen Lernen, pp. 33–56. Vandenhoeck & Ruprecht (May 2010)
20. Oppl, S.: Subject-Oriented Elicitation of Distributed Business Process Knowledge. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 16–33. Springer, Heidelberg (2011)
21. Oppl, S.: Subject-Oriented Elicitation of Distributed Business Process Knowledge. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 16–33. Springer, Heidelberg (2011)
22. Oppl, S., Stary, C.: Tabletop Concept Mapping. In: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI 2009). ACM Press (February 2009)
23. Oppl, S., Stary, C.: Effects of a Tabletop Interface on the Co-Construction of Concept Maps. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part III. LNCS, vol. 6948, pp. 443–460. Springer, Heidelberg (2011)
24. Saint-Andre, P., et al.: Extensible messaging and presence protocol (XMPP): Core. Tech. Rep. 3920 (2004)
25. Sarini, M., Simone, C.: Recursive Articulation Work in Ariadne: The Alignment of Meanings. In: Proceedings of COOP 2002, pp. 191–206 (2002)
26. Strauss, A.: Work and the Division of Labor. The Sociological Quarterly 26(1), 1–19 (1985)
27. Strauss, A.: Continual Permutations of Action. Aldine de Gruyter, New York (1993)
28. White, S.A., Miers, D.: BPMN Modeling and Reference Guide: Understanding and Using BPMN. Future Strategies Inc. (2008)

# Subjects vs. Objects – A Top-Down Approach

Clemens Krauthausen

IT Engineering
Moritz-von-Schwind-Weg 70, 82343 Pöcking, Germany
c.krauthausen@bp-engineers.de

**Abstract.** This article demonstrates how to improve the link between the requirements of the business and the capabilities of IT. The fundamental starting point is to define business objects solely from the business point of view, thus establishing the predominance of business requirements when interpreting and using established IT functions.

Starting from business objects, the content of the business - what to do – is to be identified and determined. The general principle is that all the business functions as well as the communication needs should be determined and decided by the people responsible for the business: These are defined as the subjects responsible for determining and processing business activities.

As opposed to that, the IT functions have a supporting role. In particular, the many IT functions which result from various technical needs should be identified and be placed under the sole and entire responsibility of the IT. This way a clear focus of the IT on the defined business needs can be maintained.

**Keywords:** S-BPM, business object, business task, business object services, workflow, transaction services.

## 1 Introduction

The quality and efficiency of business processes depends conclusively on the extent to which the functional requirements in the enterprise are recognized properly and transformed correctly. This is particularly true when IT tools are to be used. If any of the functional requirements immanent in the enterprise's environment are disregarded, by mistake or otherwise, this may result in solutions, which could be inefficient, needlessly complex or – in the worst case – entirely useless.

The subject-oriented concept is a natural and most helpful way to describe the commercial purpose and the requirements arising in an enterprise. A full understanding of these factors and their exact and reliable description are the basis on which responsible decisions as regards the introduction and application of the business processes concerned can be founded. At the same time, this basis also constitutes the framework within which IT will have the freedom necessary to implement those IT tools which provide adequate, correct and effective support as required for the respective business processes.

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 163–181, 2012.

## 2 Business Activities

An enterprise is an organization of employees acting to produce and deliver products and services to customers within a market (cf. [9]). This description is sufficiently broad to cover non-profit organizations and public utilities as well.

A general precondition allowing an enterprise to fully serve its purpose is that the actions of all employees are in line with the goals and strategy of the enterprise. Accordingly, an essential task of BPM is to make the operations of a company more successful. This is achieved by ensuring that the organization of the business clearly is in accordance with its goals and strategy and by providing the company with the appropriate tools.

Following Luhmann [17] an enterprise can be seen as a (social) system to which his constructive system theory can be applied. Our aim to represent the ability of enterprises to implement their respective aims and to generate the desired outcome can also be seen in this context. In actual fact, our top-down approach to establish a system based on subjects and business objects is compatible with Luhmann's view. We also follow Luhmann's interpretation that employees are not part of the system; rather, they are represented by subjects.

Communication is an essential aspect of social systems. However, in our approach communication is only dealt with in so far as it is required for controlling the division of labor.

Our top-down approach does not claim universal validity. Nevertheless it can be shown to be useful in those enterprises or business activities, where the focus is not on the activities as such, but on the results to be achieved.

Without an appropriate organization of the activities performed within an enterprise, everyone could do everything. An unambiguous assignment of tasks to the respective functions of employees is essential. Following an idea proposed by Fleischmann, this can be seen as a stepwise restriction of activities to improve efficiency.

Thus subject-oriented BPM (see [11], [12]) is a method which provides a common and comprehensive analysis and description referring to the following concepts:

- subjects (the agent who is acting)
- objects (the aim or outcome of an action or activity)
- predicates (what is done to the object)

An action presupposes the existence of an actor. If various single actions are joined together, action nets (including communication channels) are established. They are usually required in order to accomplish complex tasks. Employees acting in an action net are the subjects of such a net.

## 3 Responsibilities, Tasks and Actions

The basic principle underlying the organization of an enterprise's activities is fairly simple:

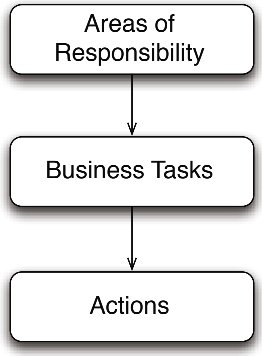



**Fig. 1.** From responsibility to action

- Start acting whenever this is necessary
- Do the right things
- Make sure that every action is completed in good time.

Buchwald [3] points out that a clear definition of responsibilities is a key prerequisite, as these responsibilities are a good starting point for organizing activities.

An area of responsibility assigned to an employee will also define the competence of the respective employee. This will include giving the employee the power to take decisions in his area of responsibility as well as providing him with the resources which are needed to achieve the required results. Assigning certain responsibilities will result in a set of particular business tasks and further in actions, which must be executed in order to reach the pre-defined goals.

## 4 Objects and Tasks

### 4.1 Business Objects

With a result-oriented approach, tasks can be described very well by referring to those items, which are the aim or the outcome of the actions. These items are called business objects, and they are at the center of actions.

Ultimately, all pieces of information required in a business process must be assigned to an appropriate business object.

**Definition 1.** *A **business object** is a well-defined entity of business activities which must be treated in a specific way. For the duration of its life cycle the business object must exist as an identified object.*

A business object always is the outcome of the respective business activities, i.e., it results from actions, which will or should occur in an enterprise. Thus, our definition of business objects is only based on those properties and attributes, which are given by the business. As a business object is required to be well-defined, it should always be unambiguously clear what is part of the business object [1].

Assets will usually be business objects. In particular they will include **tangibles**, which are business objects with a physical existence. Intangible assets, which by definition have no physical existence, can also be business objects.

Going beyond the members of the asset category, there will be other items with or without physical existence, which are business objects, too. With a view to the approach taken by Popper [23], the business objects with physical existence are "World One" items, while business objects without physical existence belong to "World Three". Obviously bank accounts or electronic invoices are also "World Three" items, and as they are affected by business activities, they will be business objects, too. Irrespective of their physical existence all business objects are real with respect to the business.

Whenever business objects are dealt with by IT systems, they are modeled as so-called artifacts [2] and realized as data objects. "World Three" objects can be realized as IT-world artifacts. To make tangible objects manageable for IT systems, one must create an artifact and the corresponding data object(s) as a substitute. For real world objects and their respective IT world substitute artifacts suitable mechanisms must be defined in order to guarantee that they are identical.

Business objects generally require a specific treatment, which follows from their purpose or from legal regulations.

Examples for this are:

- an account allows pre-defined transactions only
- a secure object: certain security regulations will apply
- selected other objects: regulations as regards transport and archiving will be in force
- objects resulting from contracts: statutory or contractual retention periods may apply

Every business object has a defined life cycle. This begins with the creation of the respective individual business object (as an instance) or when it begins to be used as an identified object in the processes of the enterprise. It ends, when the business object is taken out of service by the enterprise [3]. Within its life cycle the business object generally will change, even though it will retain its integrity and identity. The change cycle can go through specific states, depending on the kind of business object. These states depend on its use and the development of its value. Consequently, each business object exists in exactly one valid state at any one time.

[1] This definition includes results, which are created by a supplier. After accepting a result as an identified object, it may be used as a (input) business object.

[2] UML [8] defines an artifact as a tangible by-product produced during development.

[3] During their respective life cycles business objects may appear as various transient or persistent (technical) objects in applications.

During its life cycle, every business object is processed as an individual object. It has an unequivocal identification, which allows it to be addressed directly and to be verified without doubt.

The inevitable end of the life cycle of a business object cannot only be reached by its physical destruction. From the point of view of the enterprise, the existence of a business object will also end, when the business object leaves the enterprise. At that point, systematic knowledge about the business object is no longer maintained in the enterprise: Accordingly, it becomes a non-systematic action object. However, outside the enterprise the (former) business object can still continue to exist. Delivered products, which are no longer maintained, are typical examples for that.

Usually, many dependencies exist between business objects. These can be represented as relationships. To guarantee the integrity of business objects, such relationships must be reproducible, that is, they should exclusively be derived from the content of the business objects, e.g., by reference to other business objects or by clearly defined rules.

The completeness of a set of business objects can be validated by making use of scenarios, which cover the full scope of the system or model concerned. Practical experience suggests that a manageable number of 10 to 20 business objects is sufficient to establish a typical set of scenarios.

### 4.2 Business Tasks

Enterprises have a "very good understanding of their business needs" (see [15]). They know their tasks and how to "do the right things". It is not the function of BPM to create new business needs, but BPM often demands and requires new IT capabilities: Thus the existing business needs are the starting point for all activities.

In [7] Davenport defines a business process as "a set of locally related tasks performed to achieve a defined business outcome". The "Artifact Method for Business Process Design" [5] introduces "functional chunks (tasks) that are consistent with the purpose", which cause the required change of artifacts.

Based on this ideas we define a task as the production of business objects, where reference is made to their limiting conditions and their goals, at the same time omitting all non-essential details. Within a particular task, all those actions are allowed which can lead to the desired results while meeting the given limiting conditions.

**Definition 2.** *A **business task** is a defined task to be executed according to plan and as part of the activities of the enterprise. A business task is defined by:*

- the person responsible
- the internal and external events which cause the actions of the business task
- the business objects which should be achieved by the business task
- the business objects, including their required quality, which should be created by the business task

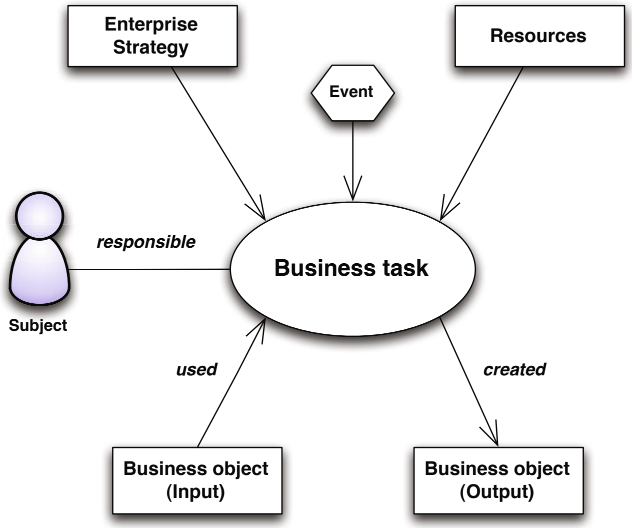



**Fig. 2.** Subjects and objects within a business task

– the business objects, including the defined requirements as regards their quality, which can be used by the business task
– the treatment of exceptions

For particular actions, a business task defines the results, which are to be achieved in certain situations, where a clearly defined target is given. Looking at actions and business objects simultaneously, as is also implied in the scenarios referred to in section 4.1, will – in practical applications – almost automatically result in an appropriate granularity of the business tasks.
The business task does not define the tools to be used, the actions to be performed and the order in which actions should take place.

For each business task there is a specific subject, which is the substitute for an employee who will bear the **responsibility**. The responsibility covers the determination of the business task as well as its execution. As explained later, certain parts of a task, in particular its execution, can be delegated.

The arrival of a message, a signal of a timer, creating an idea, detecting a disaster are examples of specific situations where action may be necessary. Typically, such specific situations constitute an **event**, which defines the starting point for executing the business task. One event may, however, be the starting point of different business tasks. Thus it the responsibility of the subject to provide for the selection of the suitable business task and the initiation of the respective actions.

The business strategy of the enterprise and the available resources determine the goals and the results to be achieved. The results to be achieved and the items to be used are described as business objects. As far as the input is concerned, its

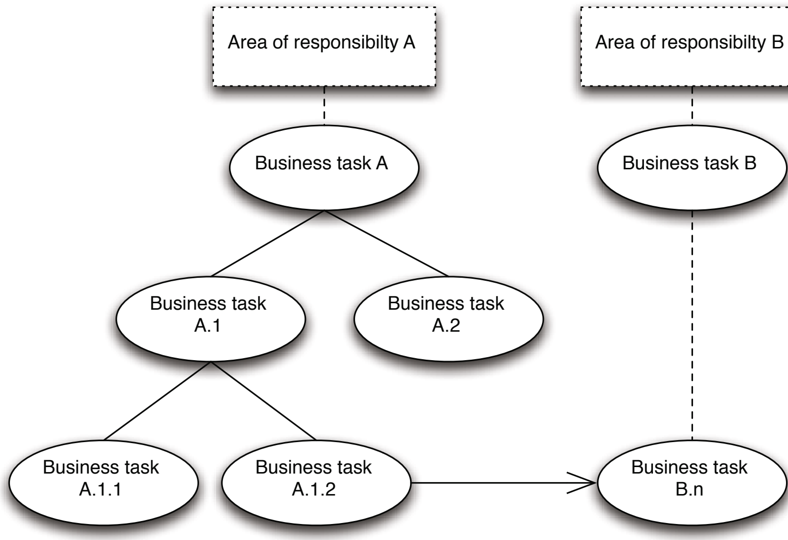



**Fig. 3.** Business task decomposition

quality may be defined by preconditions, whereas postconditions will determine the quality of the output.

An indispensable quality of a business task is that an end is clearly defined. This end is either reached by attaining the defined purposes and results or, in exceptional cases, by breaking off the task in a controlled manner.

### 4.3 Business Task Decomposition

In order to analyze and construct business tasks in detail, subtasks may be worked out, starting from an area of responsibility.

The aim of the decomposition of a business task is finding all the subtasks it consists of. Within the subtasks re-use of business objects will happen, but also new business objects may be introduced. Employing the results of decomposition, business decisions about organizing activities can be documented and communicated in an understandable form. We use the term 'task decomposition' instead of 'functional decomposition' (see [28] p.78 ff) in order to make clear that the focus is on responsibilities and goals rather than on technical aspects.

Re-use of results, i.e. of business objects, presupposes a fundamentally stringent structure of tasks. Hence it induces causal dependencies on the business tasks. These concepts of logical dependencies are well known in software engineering. The 'make' function (or newer concepts like 'ant' and 'maven') is a powerful implementation of dependencies in automated tasks in software production.

Different stages in the production of one business object, which form the input or output of different business tasks, can be distinguished by the various life cycle states.

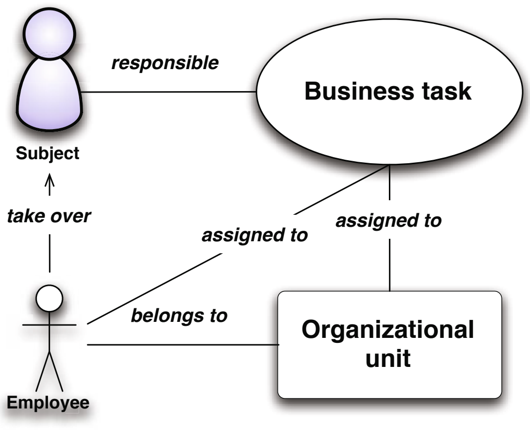



**Fig. 4.** Subject role taken over by an employee

Results which are not re-used have no effect on the value chain and can thus be omitted, except when they are final results of production.

### 4.4 Subject Role

Organizational units provide employees and resources for executing the respective business tasks. As a rule, various different business tasks will be assigned to a particular organizational unit and its employees. The basic concepts of roles as found in [28] p. 103 define a further modularization of organizational units and the respective kinds. Such concepts will not be discussed here.
With this background one can find the suitable individual employees to take over the subject role.

Complexity increases, when a business task is assigned to more than one organizational unit. In such a situation the assignment of employees and their responsibilities is determined on the basis of additional information on the content of business objects, on the basis of relationships between organizational units etc.

## 5 Increasing Productivity

Within the preceeding sections the focus was on the content of the business and the organization of the tasks with respect to their results. This section focusses on improvements in productivity.
Here our focus is on two mechanisms used to improve productivity:

- *division of labor*
- *use of tools and determining activities*

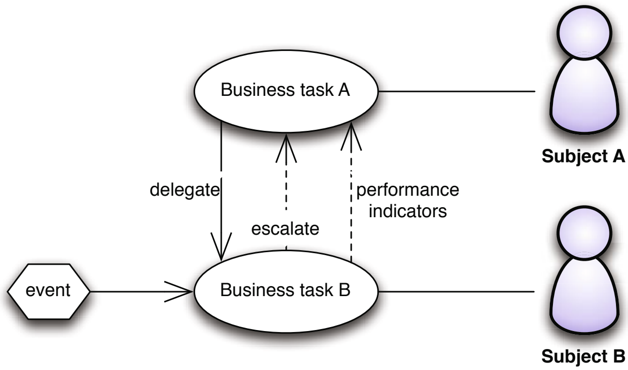



**Fig. 5.** Delegation of a task

### 5.1 Collaboration and Responsibilities

The number of subjects is determined by the number of business tasks. As a rule, the division of labor will not result in an increased number of business tasks[4], but in a greater number of employees, who can take over the respective subject roles. As a consequence, additional communication between the subjects involved is necessary. Such communication is not possible between business objects; in our concept it is only allowed between subjects. No communication is required, whenever one single employee takes over various different subject roles at the same time.

Two types of relationships are introduced as regards the subjects sharing responsibility.

**Delegation.** Delegation is involved, whenever an exactly defined part of a task will be transferred to another employee. Typically, it is the execution of a task which will be delegated. Using the decomposition model, delegation can be described as assigning the subject role in subtask B to an employee who is not involved in the subject role A. Nevertheless, the correct execution of the delegated (part of the) task remains with the delegating subject.

Generally, events will be transferred with the delegated task. In case a delegated task cannot be executed under the given terms, an escalation is generated. To accomplish the task in the way originally intended, various options will be available: More resources will have to be provided, the target-setting could be modified or the task could have to be cancelled. The respective decision is made by the delegating subject.

Whenever a task is delegated, the delegating subject will be responsible for monitoring the correct execution of such a task. Monitoring can be based on performance indicators like cycle time, costs, etc..

[4] The division of labor may result in a higher degree of specialization. Such specialization may in turn require a more sophisticated decomposition of business tasks.

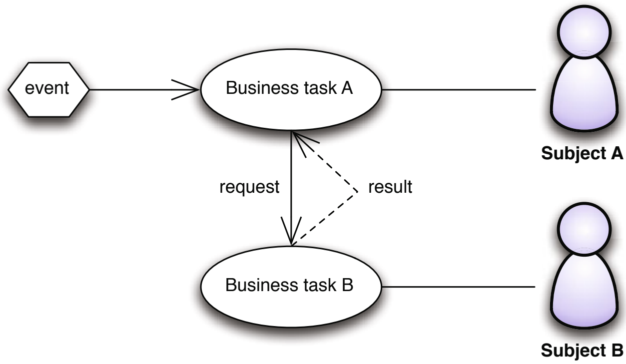



**Fig. 6.** Cooperating business tasks

**Cooperation.** Initiating cooperation is like requesting a service. After completion of the subtask under a cooperation agreement, the results achieved are processed by the former subject, which then can continue its task.

A cooperation is based on an agreement or contract between the partners involved. The principal must take responsibility as to whether the requested service is sufficient for fulfilling the original task. The agent answers for the execution of the particular task, as laid down in the contract.

### 5.2 Communicating by Results

Business tasks can only be executed if the needed business objects have been created or if they have reached the required status in their life cycle. The business task can be executed as soon as the desired result is discovered. The discovery can be made in two different ways: Access to the respective business object can firstly be made via a request by the subject which is required to execute a new business task. Alternatively the subject will be provided with an appropriate message.

In either case the requested business object is controlled by a responsible subject, which usually is its creator. From the point of view of the subjects involved, communication implies either allowing access (e.g. via a db request) or providing the respective message (e.g. via "publish and subscribe"[5]). No further communication between the subjects is necessary. The creator may be able to make do without any further knowledge of the using subjects, and the users will not necessarily need information about the producing subjects.

### 5.3 Access Control

The access capabilities for business objects are determined by their use within the business tasks. As a consequence of the deployment of a greater number of

[5] Publish and subscribe [10] is a powerful concept for reuse of business objects.

employees, the access to sensitive business objects may have to be restricted to the required use. Generally, whenever a model captures a business task completely, access to the corresponding business objects is determined without any exception, such that additional restrictions of access are not needed. However, when a stepwise restriction of activities to improve efficiency is used, the model does not show the required completeness in all respects. In that situation the consequence is that the alternative solution of introducing access restrictions has to be used.

Based on S-BPM, access control will be defined as a relationship between subjects, (tangible or other) business objects and the specific access methods. In the case of tangible business objects, access control affects the complete business object. This concept is easy to understand and to communicate - an important goal of S-BPM. Consequently, the concept of accessing complete business objects should be applied to the other business objects and the respective artifacts, too. Whenever the complete access to a business object is not desired, the respective business object can be split into smaller parts, with appropriate access controls.

## 5.4 Business Task Routing

The decomposition of a business task like a purchase order from a customer leads to many subtasks. Their pre- and postconditions induce a flow of business objects. This flow is accompanied by restrictions to the allowed processing order of the tasks, which result from causal connections. In addition, there are cases where efficiency can be improved by prescribing a particular sequence in the execution of business tasks, even though such sequencing is not forced by causal connections. It represents an orchestration of business tasks, which is implemented by using the concept of business task routes.

**Definition 3.** *A* ***business task route*** *schedules a processing order of business tasks.*

Within a route, alternate or parallel parts may be valid. Whereas alternate parts should depend on some conditions or business rules, parallel parts may be processed in any order.

Furthermore some of the allowed routes may be more time or resource consuming than others. Thus a more detailed definition of a route may reduce cycle time and costs.

Regarding the division of labor, the subtasks constituting a business task may be taken over by different subjects. In addition to the flow of business objects, further communication between the subjects involved may be necessary to focus on the ongoing task. The subjects concerned and the employees taking over the respective subject roles must be identified.

A business task route may form a circle. Thus the starting business task may be the last task in the route too.

### 5.5 Business Task Activities

The foregoing concepts are valid even without considering the activities being part of the business tasks. Looking at a business task in greater detail reveals the actions to be executed by the subject while processing the task. While the determination of "what is to be done" depends on the business task, the question "how to proceed" is answered by referring to the particular business task activities. There may be various different ways of executing a given task. With regard to our goals three categories of measures will be distinguished:

– *Working with business objects*
– *Managing communication with subjects*
– *Orchestrating activities*

With each task, dealing with business objects will be at the center. Business objects are the action items of all business tasks. They will be created, accessed, modified and deleted. Decisions will be made based upon the content of business objects.

Communicating with other subjects is an additional measure which is essential when performing a business task. This will include communication via business objects as well as communication using messages.

The sequencing of the subject's actions, the orchestration, can also result in increased productivity. Orchestration activities can thus be appropriate. Without orchestrating activities, the subject may determine the activities and their particular order.

The details of such activities are highly dependent on the use of tools. With comfortable tools, the number of actions necessary may be reduced significantly. Furthermore, completely orchestrated business tasks can be executed automatically, if it is ensured that all necessary decisions are taken based on clear rules. Orchestrated tasks should be considered as delegated tasks. Following the regulations explained in chap. 'Collaboration and Responsibilities' the delegating subject takes the responsibility of monitoring the correct execution of the respective task.

The performance of the activities constituting a business task essentially is independent of whether or not an orchestration as described above is employed. However, orchestration methods can be used as precision tools in order to improve efficiency, thus avoiding over-engineered processes.

It is clear that determining business task activities is independent of the used business task routes. Thus a workflow can combine these concepts in any desired manner.

## 6 Link to IT

IT is expected to offer tools for raising the abilities of the employees in an enterprise as well as their creativity to the level required by the business. The requirements of the business determine the power of the tools. Requirements must be accurate and precise, but without predetermining a particular solution. In the course of the implementation of the IT tools, many technical problems must be solved. The need for additional functions as well as their effects are sometimes difficult to understand from the point of view of the business.

The previously described concepts of subjects and business objects suggest the division of the IT functions into two categories. The first of them takes the business objects, the second one takes the subjects as its domain.

Service Oriented Architecture (SOA) [21] is a favorite concept. Its modular application software can almost easily be assigned to the respective categories. Nevertheless, legacy applications are prevailing. The complex and complicated functions established in legacy applications should thus be interpreted differently now when taking the business needs into account: Making use of the concepts of subjects and business objects described earlier, the predominance of the business requirements can be considered.

### 6.1 Business Object Services

The purpose of these services is to provide tools, which support activities dealing with business objects. This includes functions like creating, recovering, updating and transforming objects. In IT, business objects are typically realized by data objects. These are determined by their UML artifacts, which lead to their content model, by their relationships to other data objects, and by the operations on business objects provided by the IT. IT functions will normally be implemented in different IT applications. Each of these applications has data models of its own in order to implement specific functions. This is the reason why the business view of one single business object will simultaneously be mapped and copied to various different application specific data objects.

The ESB (Enterprise Service Bus [6]) is an approved IT concept used for data exchange on the basis of objects. Within the ESB, business objects may be defined independently from IT applications.

XML has found wide use as a standard for the platform-neutral description and implementation of business objects. It is the established standard to represent data objects. While UML artifacts are (tangible) by-products, XML can be used to create in-product components. Thus, designing business objects, tasks etc. is possible by creating such items as XML objects, which then get part of the solution.

An additional advantage of XML consists in the fact that many standard tools are available (e.g., validating parser, XPath, XQuery, XML data bases) [18] for prototyping. The fact that no special IT knowledge is necessary to understand XML concepts allows easy co-operation of the persons responsible in the enterprise.

All functions relevant for business can be explained by using such application independent definitions. No knowledge about application specific objects, which might be confusing, should be necessary.

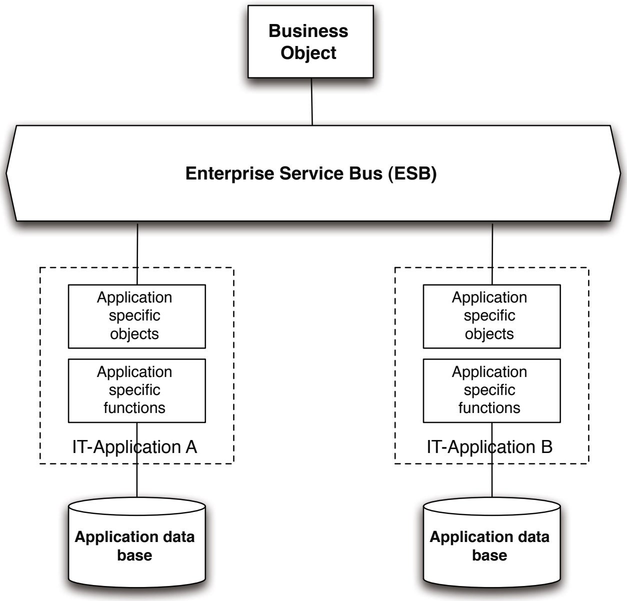



**Fig. 7.** Mapping of business objects to many applications

**Quality of Business Objects.** The quality of business objects is defined by their pre- and postconditions within business tasks. It is of central importance. From the point of view of IT, quality can be defined as syntactical correctness of objects. Business rules are used to define the syntax rules. With respect to business objects there are two categories of rules:

The first category concerns isolated business objects: Is a business object formally correct without knowledge of the existence and the content of its related business objects? The answer can be found by using the concept of schemes. This is the preferred technique to determine mandatory and optional properties of objects including the allowed values[6]. XML Schema (see [27]) is a W3C standard of schemes. CCTS V2.01 (Core Component Technical Specification, see [26]) is an ISO and UN/CEFACT standard for content modeling. Nevertheless the latter so far has not found the desired dissemination.

A standardization of the business objects could release considerable synergy potentials. Communication solutions for specific lines of business like ebXML, RosettaNet or the variants of EDIFACT, have been in use for many years.

The second category concerns related business objects. Related business objects may be addressed directly using their identification, or they may be selected by joining various properties. Thus additional conditions may require the existence of the related objects including their cardinality[7]. Furthermore a relation

[6] One has to keep in mind that the scheme of a business object may depend on the status of a business object within its life cycle.

[7] The cardinality specifies the number of related objects.

may include additional conditions on the related objects. Requirements from this category may be implemented as business rules (see A. Sellner and E. Zinser, Establishing Conceptual and Functional Links between S-BPM and and Business Rules, S-BPM ONE 2010). In order to secure maintainability it is very important to link the business rules to the elements of the business model to which they belong. It may be that requirements on relationships are valid only in the context of a particular business task[8]. In this case the related business rules should be linked to the respective business task.

**Editing Business Objects.** Editing business objects by subjects is a typical activity supported by IT. There may be a lot of additional business requirements for editing:

- *Checks of inputs.* The quality of input should be checked against the quality criteria.
- *Context of information.* It may be necessary to show some information from related objects.
- *Sufficiently comfortable user interface.* The requirements as regards the user interface are highly dependent on the capabilities of the user and on the number of uses. Data inputs and outputs must be understandable from the user's view.

**Transformation of Business Objects.** In many cases the re-use of business objects implies their transformation into new business objects. One to one, one to many or many to one are possible translations.

The precise requirements for such transformations are determined by the business tasks. The resulting specification of the transformations must be understood and agreed by the individuals responsible for the business tasks.

These specifications can take place on the basis of XML. 'Transformation by example' enables non-IT specialists to describe and communicate the required transformation. Furthermore, an implementation based on the XML standard XSLT (see [25]) can be used to validate the transformation with practical examples[9] of business objects. It also can be used as a productive solution, or it can be defined as a reference implementation.

**Additional Technical Functions.** Besides the functions required by the business, there are many further technical functions to ensure that an application works correctly. It is the task of the IT experts, to define those functions in compliance with the requirements of the business.

*Remark 1.* From the point of view of the business, business objects exist during their whole life cycle. From a technical view, the persistence for the associated data objects must be guaranteed. These are typical database functions.

[8] A product referenced in an order must be released for selling, if the order is used while fulfilling the contract. But in case of managing a warranty claim, this condition is not valid.

[9] Examples may be generated from actual data.

*Remark 2.* From the point of view of the business, each business object always has a currently valid value. Thus applications using the same business object must use the same data. From a technical view synchronization of data is necessary. Concurrent changes must be prohibited or handled correctly.

*Remark 3.* The quantity structure of the business tasks is an important fact for technical solutions. Quantities have to be calculated with regard to business objects and they must be mastered by the IT.

### 6.2 Transaction Services

In this context transactions are understood as active processes. The challenge is the co-operation of many employees resulting from the division of labor. Division of labor makes it possible to use the different capabilities of employees and to execute actions in parallel. Coordination of the subjects is necessary. IT can provide the following supporting functions:

- Communication between the subjects involved in a business process
- Control of actions to be executed in business tasks

**Communication between Subjects.** The communication will be supported through messages between the subjects involved in a transaction. The acting subjects must be determined and addressed, and they also must be activated by messages.

Nowadays a communication directory with all available communication addresses is a standard feature in an enterprise. Nevertheless, many applications introduce their specific identification schemes, which may be error-prone.

With regard to business task routes, targeted subjects may be dependent on various conditions, including replacement rules defined in the organizational structure. The mechanism for any address resolution must be determined by the business. The handling of escalations is a further case of conditional routing.

*Remark 4.* In view of S-BPM, automatically executed business tasks are controlled by an automated subject[10]. In such a situation the automated subject is the target of the messages in the transactions.

**Control of Actions.** Within a business task two types of actions occur:

- *Control of access to business objects*
- *Managing communication*

The access to business objects may be supported by business services. Improvement of productivity can be achieved by making suggestions for actions performed by the subject.

On the other hand control can be understood as restrictions to allowed actions. This includes the use of predetermined applications, functions or objects.

[10] Whenever the delegating subject is an automated subject, it may be referred to as an avatar.

Management of communication may be supported by functions for receiving and sending messages or by procedure briefcases.

**Monitoring Transactions.** Complete each action in good time! Transaction monitoring is helpful to gain control over incomplete transactions and over cycle time.

**Technical Solutions for Transaction Services.** These tasks are taken over by workflow tools (see [16]). The BPEL standard (see [21]) sets up a platform-neutral language for the technical description of processes, which are processed by workflow engines.

Many workflow engines are available. With regard to subject orientation, the workflow solution jPass/jFlow/jLive from METASONIC (see [19]) offers a convincing solution.

Furthermore we are prototyping two variants of a workflow implementation based on the concepts of business tasks and business objects. The first of them gives users, who take over subject roles, access to the appropriate input business objects and a set of related functions. The second one implements a so-called steptree on the complete decomposition of a business task. The steptree is controlled by a subject, whereas all subtasks are executed automatically. In the case of an error, the subject may analyze and modify business objects and, if the problem has been solved, restart or continue the automatic execution.

## 7 Conclusion

Business objects as a basis for enterprise-wide data models are in the focus of various authors (see [22] [1] [14]). However, the business object concepts are frequently driven by the concepts of object orientation as defined by IT. Quite a number of them are referred to as "business objects", even though many of these are only of technical interest. Their business relevance is doubtful. This leads to complicated models which are difficult to understand without knowledge of IT methods and procedures.

The S-BPM concept described here brings subjects and business objects into its focus. In the subject orientation concepts, subjects are the actors, which are responsible for determining and processing the activities. Business objects are identified as action items. The top-down approach puts the requirements of the enterprise concerning commercial purposes and enterprise targets into the foreground. On this basis all decisions must be made starting from the enterprise responsibility concerning the details of the business processes. For the allocation and perception of responsibility, the top-down approach on the basis of subject orientation is a useful concept in the design phase. Requirements for IT solutions are based on the intrinsic needs of the business.

Where technical details are in the focus, the bottom-up perspective is recommended. Hence, the change between both perspectives will contribute to a deeper understanding, also improving the quality of the respective solutions.

In total, S-BPM is a powerful method, to determine, to communicate and to organize efficiently the activities of an enterprise, including the use of tools, while always keeping the focus on the business needs.

## References


1. Allweyer, T.: Geschäftsprozess-Management. W3L GmbH (2005)
2. Allweyer, T.: BPMN - Business Process Modeling Notation. Books on Demand (2009)
3. Buchwald, H.: The Power of 'As-Is' Processes. In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 13–23. Springer, Heidelberg (2010)
4. Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.): S-BPM ONE 2009. CCIS, vol. 85. Springer, Heidelberg (2010)
5. Chao, T., Cohn, D., Flatgard, A., Hahn, S., Linehan, M., Nandi, P., Nigam, A., Pinel, F., Vergo, J., Wu, F.Y.: Artifact-Based Transformation of IBM Global Financing. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 261–277. Springer, Heidelberg (2009)
6. Chappell, D.A.: Enterprise Service Bus. O'Reilly Media Inc. (2001)
7. Davenport, T.H.: Process Innovation: Reengineering Work Through Information Technology. Harard Business School (1992)
8. Encyclopedia, W.T.F.: Artifact (software development), http://en.wikipedia.org/wiki/Artifact_(software_development)
9. Encyclopedia, W.T.F.: Business, http://en.wikipedia.org/wiki/Business
10. Encyclopedia, W.T.F.: Publish/subscribe (2011), http://en.wikipedia.org/w/index.php?title=Publish/subscribe&oldid=456928543
11. Fischer, H., Fleischmann, A., Obermeier, S.: Geschäftsprozesse realisieren. Vieweg (2006)
12. Fleischmann, A.: What Is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
13. Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.): S-BPM ONE 2010. CCIS, vol. 138. Springer, Heidelberg (2011)
14. Grässle, P., Baumann, H., Baumann, P.: UML projektorientiert. Galileo Press (2001)
15. Komus, A.: Key Lessons from Wikimanagement and BPM Best Practices: Aspiring for a Truly Holistic Approach in BPM. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 3–16. Springer, Heidelberg (2011)
16. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall (1999)
17. Luhmann, N.: Einführung in die Systemtheorie. Carl-Auer (2009)
18. Melton, J., Buxton, S.: Quering XML - XQuery, XPath, and SQL/XML in Context. Morgan Kaufmann Publishers (2006)
19. Metasonic: Metasonic site, http://www.metasonic.de/
20. Oestereich, B.: Analyse und Design mit UML 2. Oldenbourg (2005)
21. Pant, K.: Business Process Driven SOA Using BPMN and BPEL. Packt Publishing Ltd. (2008)

22. Paper, O.W.: Oracle application integration architecture enterprise business objects (ebo) concepts (2009), http://www.oracle.com/us/products/applications/057277.pdf
23. Popper, K.R.: Objektive Erkenntnis. Hoffmann und Campe (1984)
24. Porter, M.E.: Competitive Advantage. Free Press (1985)
25. Tidwell, D., Laurent, S.S., Romano, R.: XSLT. O'Reilly Media (2008)
26. UN/CEFACT: Core components technical specification, http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf
27. van der Vlist, E.: XML Schema. O'Reilly Media (2002)
28. Weske, M.: Business Process Management. Springer, Heidelberg (2007)

# Using S-BPM for PLC Code Generation and Extension of Subject-Oriented Methodology to All Layers of Modern Control Systems

Harald Müller[1,2]

[1] Johannes Kepler University Linz
Department of Business Information Systems - Communications Engineering
Freistaedterstrasse 315, 4040 Linz, Austria
[2] Metasonic AG
Münchner Straße 29, Hettenshausen, 85276 Pfaffenhofen, Germany
harald.mueller@{jku.at,metasonic.de}

**Abstract.** Interdisciplinary collaboration has become a challenge in industrial enterprises. Substantially for overcoming departmental borders are supporting software systems and a common understanding of technical and also business processes. A Production Planning and Control System (PPC) supports the specialized divisions and provides data management tools with the aim to reduce processing time and increase in productivity. Embedded to an Enterprise Resource Planning System (ERP) a high level of integration can be reached. Communication between the parties involved is changing because of a SOA and standardization among all levels of modern control systems. Therefore BPM 2.0 can be used for business processes as well as control layer processes, if necessary definitions to fulfill communication and execution are added. IEC 61131, the only standard in automation control, and IEC 62541, a draft standard for vertical data integration, supports these technological change. This paper provides an approach for subject-oriented process modeling and inter-layer communication, starting at the control layer up to business process using subject-oriented methodology.

**Keywords:** IEC 61131, IEC 62541, SCADA, BPM 2.0, automation, process control, control process.

## 1 Introduction

Subject-oriented automation process modeling is an enhancement to S-BPM with the aim to influence the subject-oriented modeling methodology on different layers of industrial companies like Manufacturing Execution System (MES), Supervisory Control and Data Acquisition (SCADA) and the Control Layer. This is possible because of technological progress in automation industries which has rapidly increased since the early 1970s. If there were expensive hard-wired logic controllers at the beginning, today program implementation is based on high level computer-language. IEC 61131 defines the basic conditions for the development of programming languages and PLC applications. Part 3 of IEC 61131 deals

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 182–204, 2012.

with programming languages and defines two textual (IL: Instruction List, ST: Structured Text) and two graphical (FBD: Function Block Diagram, LD: Ladder Diagram) PLC programming language standards and the Sequential Function Chart (SFC) which provides elements to organize programs for sequential and parallel control processing. The main advantage is the possibility to use textual and graphical programming languages within one PLC application. In context of automation process modeling, subject-oriented Model Driven Development (MDD), can be used on the Control Layer for code-generation of IEC 61131-3 applications using the PLC programming languages and application flow control. If we focus on the definition of Model Driven Development (MDD) "*... a set of approaches in which code is automatically or semi automatically generated from more abstract models, and which employs standard specification languages for describing those models and the transformations between them.*" [17] and contrast MDD with the graphical languages of IEC 61131-3 (e.g. FBD), there is a congruence. Indeed the abstraction level - the difference between specification and code - is very slightly [11, p.2]. Due to the fact that total costs of programming is often higher than the hardware itself, and a change of process control applications has a huge impact to change management leads to the logical consequence to raise the abstraction level of automation process applications which means that we need to allow developers to work in more abstract models. Model-based Software Engineering (MBSE), besides the discussion about Platform-Independent Models (PIMs) and Platform-Specific Models (PSMs), can be reduced to two fundamental notions [18, p. 384]:

1. "*Raising of the level of abstraction; that is, raising the level of software specifications even further away from underlying implementation technologies (relative to, say, traditional programming languages)*" and
2. "*Raising the degree of computer-based automation used to bridge the widening gap between design specifications and corresponding implementations.*"

The term "model" in context of MBSE "*is often used as a generic term to denote any specification expressed using a higher-level formalism, whether it is an abstraction that omits detail or a fully-fledged implementation specification from which a complete executable program can be auto-generated*" [18, p.384].

The IEC 61499, which is an object-oriented further development of IEC 61131, is partial included to the IEC 61131-3:2009 draft standard. Besides the discussion of including object-oriented aspects, this paper focuses on the subject-oriented approach pursuing the following purposes:

1. Transformation of subject-oriented model to IEC 61131-3 code on a model driven development approach should raise the level of abstraction. The method of modeling should focus on the used standard. The transformation uses the PLCopen XML Formats for IEC 61131-3 [16] definition to enable an exchange between different development environments and to fulfill the MBSE definition of "model".

2. A dynamic allocation of processes by the developer or system operator to PLC or Metasonic Flow, supported by vertical data integration functionality, should make a system more flexible and allow industrial enterprises exchange data by using a jCPEX! [13] plattform.
3. Direct integration of automation data into a Subject-oriented Business Process Model using a Behavioral Interface (BI).

PLCopen is a vendor- and product independent worldwide association trying to increase efficiency during the application development, while increasing the software quality and lowering life-cycle costs. One of the core activities of PLCopen is focused around IEC 61131-3, the only global standard for industrial control programming. The PLCopen XML specification allows to exchange programs, libraries and projects between development environments using the XML standard [16]. It harmonizes the way people design and operate industrial controls by standardizing the programming interface [15] and supports subject-orientation within the control-layer.

Supervisory Control and Data Aquisition (SCADA) as part of Industrial Control Systems (ICS) is subject-oriented enhanced by providing the possibility of vertical data integration using the OPC Unified Architecture (OPC UA) which provides a cohesive, secure and reliable cross platform framework for access to real time and historical data and events [14]. Therefor process data interchange between a programmable logic controller and a process flow control needs to be realized.

The goal of BPM 2.0 is to rapidly react to changing business environment in a complex business world. To reach this goal a BPM 2.0 approach must fulfill properties (see [10, p.86]):

– "*Only the participants in a process truly understand the complexity of the processes they are involved in.*" [10, p.86]
  If all layers of industrial processes (Sect. 2) are enhanced by subject-oriented methodology, divisional specialists are able to communicate using "the same language" and are able to define a common interface (in context of S-BPM: *Behavioral Interface*) in different processes, on different levels. This possibility supports that "*The parties involved in producing products or services have to agree on interaction behaviors for synchronizing their activities.*" [10, p.85]
– "*... the models should be executable without any additional programming or programming know-how ...*" [10, p.85].
  If processes on different levels are merged to one inter-divisional, multi-layer process using the same modeling methodology without having the same execution environments, standardized communication interfaces between these systems even though have to be a prerequisite for BPM 2.0. The ability to execute a control layer process, e.g. programmable logic controller applications, assumes the transformation of a subject-oriented process model to IEC 61131-3 application code. Therefore not the execution environment needs to be created but a Model Driven Software Engineering approach has to be assisted.

– "*... the process environment - the socio-technical system consisting of people, machines and software - should be easily integrated with the BPM model.*" [10, p.85].
  The integration of *machines* whether direct, using vertical data integration, or using a software interface on a higher layer of a modern industrial control systems (e.g ERP System) (Sect. 2) enables a company to illustrate the company wide communication model including people, machines and software.

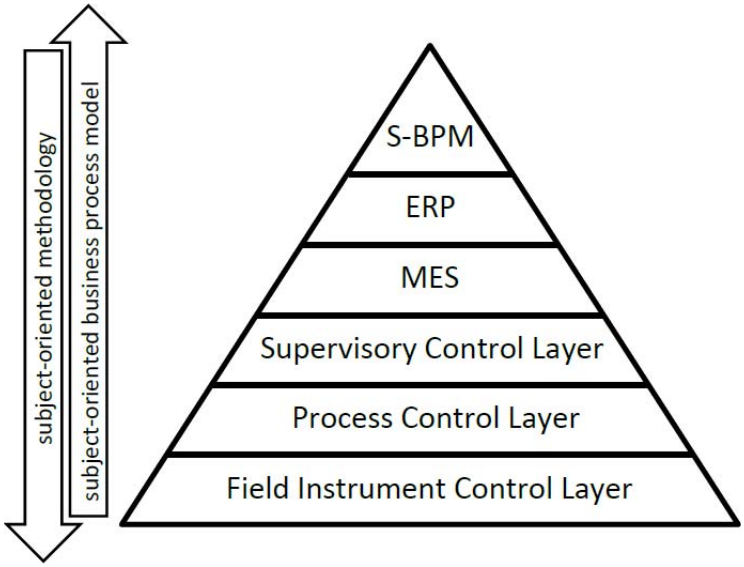
This document describes the possibilities to enhance the system layer architecture of modern control systems using subject-oriented methodology with a focus on the *Control Layer*. An introduction to modern control system architecture and S-BPM in context of automation is described in Sect. 2. Sect. 3 defines the *Subject-oriented Process Model* and introduces to the showcase used to explain the different layers. The use of subject-oriented methodology for MDSE within the *Control Layer* is shown in Sect. 4 and its subsections. The direct integration of control data to business processes and human-machine-interfaces is explained in Sect. 5, followed by the last section "Summary and Issues for Further Research".

## 2 A New Approach in Automation Engineering

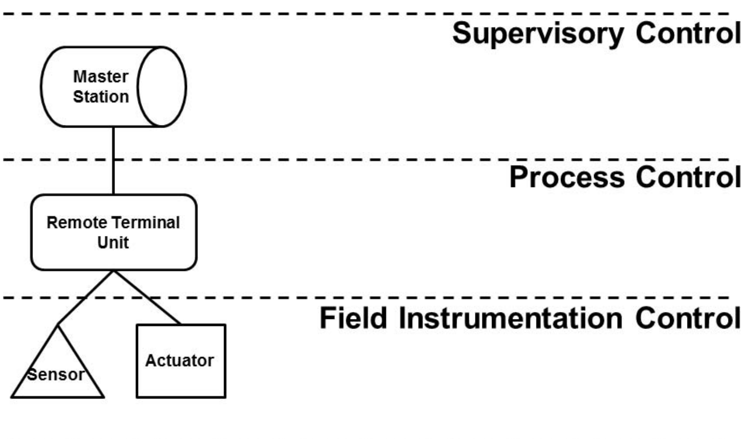
Subject oriented Process modeling in context of automation processes is not just to eliminate the word 'business' in S-BPM. There is a need to describe processes on different levels (Fig. 2). The structure of PLC-systems has changed. At the beginning a centralized PLC transfered the data to an associated server. This changed in the last years toward a more distributed architecture supported by standardization of communication infrastructure, a SOA and IEC 61131. Therefore most classifications divide modern control systems into four layers for control, visualization and production support systems: [1]

– *Field Layer with instrumentation.*
– *Control Layer with automation devices.*
– *Real Time HMI (Human Machine Interface) Layer with visualization devices.*
– *Real Time MES (Manufacturing Execution System) Layer with data processing devices.*
– *A Production Planning System (PPS) or Enterprise Resource Planning System (ERP) can be defined as a separate layer above these.*

Supervisory Control and Data Acquisition (SCADA) is the generic term for the hardware, software and procedures, used to control and monitor industrial processes. "*The latest trend of SCADA system is the three-layer SCADA architecture which depending on open system technology rather than a vendor controlled proprietary environment.*" [9, p.774] The PLCopen XML specification, used in this paper, as well as the OPC Unified Architecture is part of these open system architecture, supporting the communication between various vendors on an independent basis. Figure 2 (b) illustrates the three-layer SCADA system architecture:




(a) Subject-orientation in a modern control system




(b) Three-layer SCADA system architecture [9]

**Fig. 1.** Control system layer architecture

1. *Supervisory control layer (Master Stations)* are non-dedicated PCs processing automation duties (e.g. alarm handling, logging, trending) and have two main functions [9]:
   - *Periodically obtain data from RTUs (Remote Terminal Units)/PLCs*
   - *Control remote devices through the operator station*
2. *Process control layer (RTUs/PLCs)* usually consists of more than one device depending on the situation. The devices used are: [9]:
   - *Programmable Logic Controller (PLC)*
   - *Analog Input and Output Modules*
   - *Digital Input and Output Modules*
3. *Field instrument control layer (Sensors and Actuators)* "*This layer mainly consists of sensors and actuators. The sensors perform measurement and actuators perform control. Sensors get the data (supervision and data acquisition) and actuators perform actions dependent on this data (control). The processing and determination of what action to take, is done by the master control system (i.e. SCADA).*" [9]

Subject-oriented (business) process modeling can match or extend the different levels of the industrial IT system hierarchy. By illustrating the *Field instrument control layer* (*Field Layer*) using subject-oriented methodology, two advantages accrue:

- A standardized IEC 61499-1 [4] *Field Layer* documentation can be exported by code-transformation.
- The *Subject-oriented Hardware Model* (or IEC 61499-1 documentation) is the basic configuration for *Subject-oriented Code Generation* and therefor directly used.

A company's process control layer (*Control Layer*), in context of this document a programmable logic controller (RTUs/PLCs), needs to be described in a way that an inter-divisional process understanding is possible. Not the detailed control process provided by the vendor of a sensor, an actuator or a machine, needs to be illustrated but the process control information, the interaction between machines, sensors and actuators needs to be illustrated and embedded to a company wide process model. The subject-oriented methodology can help to bridge the widening gap between design specifications and corresponding implementations in context of automation applications. The illustration of program organization units (POUs) [7, p.51, 6.5] and configuration elements [7, p.126, 6.7] can be done with a subject interaction diagram wheras the logic of an application implemented in the body of *programs*, *function blocks* and *functions* could be modeled by using the subject behavior diagram.

A business process which mainly communicates with a PLC and works in context of automation, is a control process. If it uses the same methodology and workflow engine as a business process but is based on a different layer (*Control Layer*) the Subject-oriented Application Flow Model enhances this layer.

*SCADA* (*Supervisory Control*) is supported by *jCPEX! Automation Extension (AE) platform* (Sect. 3.1) which provides *Behavioral Interfaces* for communication purposes (e.g. to use in business processes, human-machine interfaces or for inter layer communication).

The *MES* (*Manufacturing Execution System*) in context of this paper is a technological use of the jCPEX! platform [13, p.176, 2]. If jCPEX! is not only used for cross-organizational business processes but enhanced with the aim to provide inter-layer communication, using the *Behavioral Interface*, a standard inter-layer communication interface is established. If this jCPEX! platform provides the possibility to distribute process data based on rules, a process control and control process functionality is added on all industrial system layers. Using jCPEX! platform, a MES is implemented when data exchange between SCADA and an ERP System or business process is realized.

The use of subject-oriented methodology on different layers of an industrial company allows a subject-oriented business process to communicate to all layers directly by using a standard interface, namely *Behavioral Interface* (BI). A SOA and the use of open system technology in combination with subject-oriented modeling methodology in context of BPM 2.0 enables direct communication to the socio-technical system - people, machines and software [10, p.85].

## 3 Subject-Oriented Methodology Enhances Automation on 3 Layers

Figure 2 illustrates the different layers of the subject-oriented process modeling including the automation and business process model. The *Subject-oriented Code Generation Model* (S-CGM) represents the IEC 61131-3 configuration elements [7, p.126, 6.7] and the programming model [7, p.21, 4.1]. Depending on the execution target (PLC or Metasonic Flow) the code-transformation and execution environment differs.

MDSE allows to generate IEC 61131-3 code which can directly be executed on a programmable logic controller using PLCopen XML specification as transformation target. Figure 7 shows the S-CGM and its sub models (S-HWM, S-TM, S-POUM, S-CM) which are described in Sect. 4. The *Subject-oriented Application Flow Model* (S-AFM) as part of *Subject-oriented Program Organization Unit Model* (S-POUM) communicates with the layer above by using a Behavioral Interface to enable communication between Control Layer (e.g. programmable logic controller (PLC)) and S-AFM or S-BPM using jCPEX! Automation Extension (AE) platform as described in Sect. 3.1.

The *Subject-oriented Application Flow Model* (S-AFM) as a separate layer can be used as a communication interface between hardware and S-BPM or by adding process control functionality as a (process) Control Layer similar to a software PLC.

PLC I/O variables (*DataItems* or a group of *DataItems* [5, p.8, 3.4.1]), illustrated as messages, are logically allocated to an Internal Subject and provided to the S-BPM by jCPEX! AE platform. The representation form used for inter-layer communication is a Behavioral Interface (BI) (an example is shown in Fig. 13).

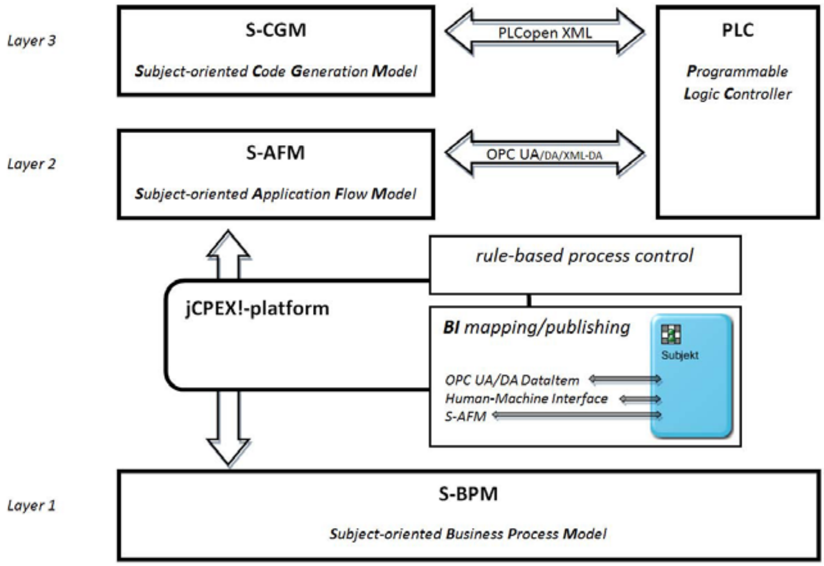



**Fig. 2.** Subject-orientation in context of automation

### 3.1 Inter-layer Communication with jCPEX! Automation Extension (AE)

In context of the jCPEX! approach, the communication between the involved partners is described as an implementation-independent choreography - the so called *Behavioural Interface* (BI) [13, p.176]. Thereby the BI can be seen as "*interface to the private process of the participation partners*"[13, p.176]. In context of automation processes we are less talking about inter-organizational communication than inter-layer communication, but the description of the observable behavior can be illustrated equally. Depending on the process, a corresponding possibility to administrate and distribute the BI has to be implemented (e.g. USDL Repository [13, p.184]).

As shown in Fig. 3, a database takes over the task of an *Behavioral Interface Repository*. A mechanism to allow communication between *Control Layer* and *Business Process* needs to be implemented. If we keep in mind, that a BI can be automatically derived from the internal private process [13, p.184], and the external subject is a known programmable logic controller *DataItem*, a mapping between an external Subject and a *DataItem*, supported by a Mapping Editor, is the basis to adopt the jCPEX! platform toward a inter-layer communication platform.

Figure 3 illustrates the communication between jCPEX! and a programmable logic controller. The modeling process and the export of a BI to the *Behavioral Interface Repository*, as well as the mapping of a BI to a PLC *DataItem* is enabled by the *Subject/DataItem Generator*, which provides a bidirectional data

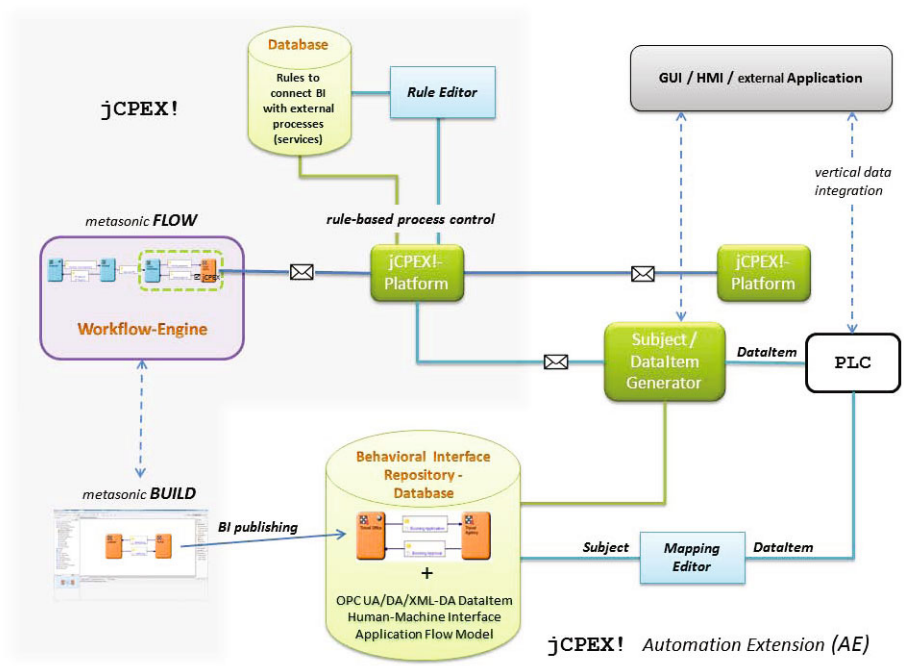



**Fig. 3.** jCPEX! Automation Extension (AE)

transfer between workflow engine and PLC by transforming a message to a corresponding *DataItem*. This communication can be realized by using the *OPC Unified Architecture (UA)* [6], *OPC DataAccess* or *OPC XML DA*.

jCPEX! platforms communicate via web services and use XML as data exchange format. A common interface allows to address different applications such as GUIs, human machine interfaces and other external applications. Figure 4 illustrates the communication between a jCPEX platform and an external application, namely *Subject/DataItem Generator* which provides a bidirectional access to a programmable logic controller and provides an interface itsself. The external application can be addressed using rules which "*facilitates replacement of a partner dynamically dependent on certain conditions - even at runtime*". [13, p.176] Therefore a *RuleEditor* can distribute incoming events to one or, if *MultiPartyBIs* are supported, more *Subjects*.

### 3.2 Showcase

In order to describe the advantage of subject-oriented process modeling, in context of automation process modeling, and the integration into S-BPM a showcase is prepared to explain the different levels of integration. Figure 5 illustrates three different domains: *household*, *factory* and *supplier of energy*.

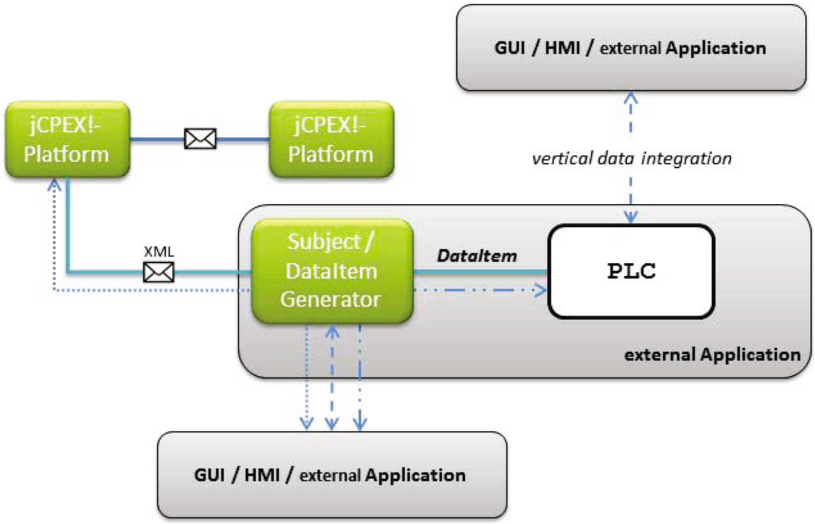



**Fig. 4.** jCPEX! DataItem Generator

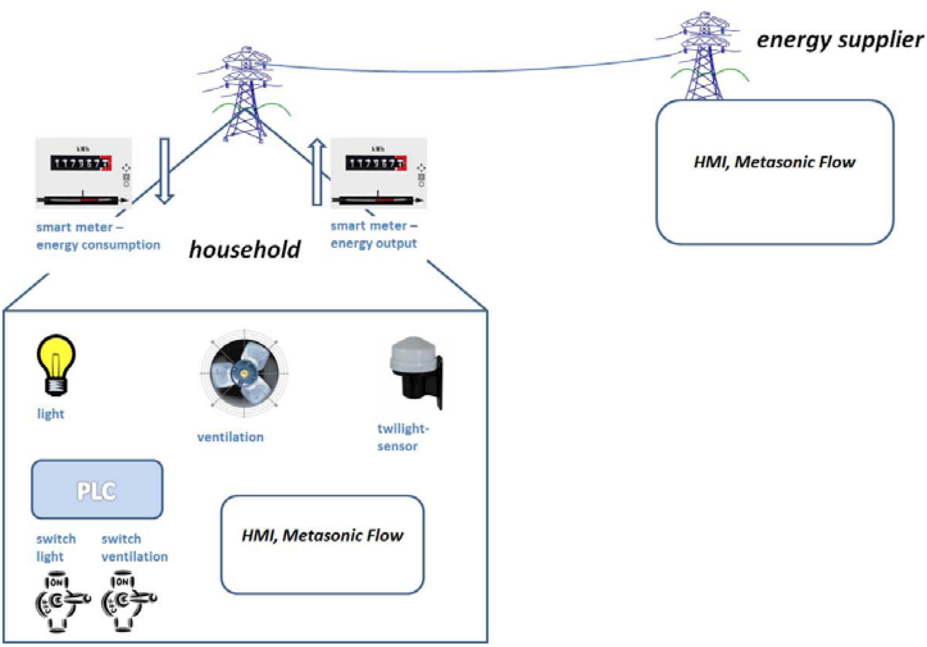



**Fig. 5.** Showcase - house control

The bidirectional communication between *supplier* and *household* as well as *supplier* and *factory* is a cross-organizational business process using the jCPEX! platform [13, p.176, 2]. The exchanged data is measured power consumption or measured power output. The process is described in S-BPM methodology using a Behavioral Interface which "*... contains the interaction behavior between involved actors and therefore represents the choreography for all concerned parties.*" [13, p. 184, 4.2]
Figure 6 illustrates the exchanged data between household and energy supplier. The S-BPM of household is kept simple with only one involved subject representing an tenant.

Within this document the subject-oriented implications are illustrated by using the example of house control. There are three sensors (*light switch*, *ventilation switch*, *twilight sensor*), two actuators (*light*, *ventilation*) and two energy meters (*energy consumption*, *energy output*). In addition to hardware I/O, the system interaction can be done using a Human-Machine Interface which is part of a SCADA system. *Energy consumption* and *energy output* is exchanged by a subject-oriented business process running in Metasonic Flow.

This showcase is used to describe the processes on the three layers illustrated in Fig. 2.

- *Layer 1*, the business process, is a cross-organizational business process with two involved subjects. The internal behavior of household includes the subject *AF automation* (Fig. 13 (b)) which provides the metered data and enables the possibility to switch the *light* and *ventilator*, ON and OFF, out of the business process.
- *Layer 2* enables the communication between *programmable logic controller* and the subject *AF automation*. Layer 2 represents the communication interface between control layer (PLC) and S-BPM using an jCPEX! AE platform.
- *Layer 3* generates IEC 61131-3 code which is described in Sect. 4 and uses Layer 2 for communication with the IT system.

The following Sect. 4 defines the prerequisites to transform a subject-oriented model to a IEC 61131-3 application using MDSE, and illustrates the transformation process using the example of house control as described in Sect. "Showcase". Layer 1 and 2 is described within one section (5) because the *Subject-oriented Application Flow Model*, as part of layer 2 and 3 (Fig. 2 and 7 (b)), is used as communication interface to integrate process data to S-BPM.

## 4 Subject-Oriented Code Generation Model (Layer 3: S-CGM)

Figure 7 (b) shows the S-CGM (***S***ubject-oriented ***C***ode ***G***eneration ***M***odel) which is matched to the procedure of implementing an IEC 61131-3 application.

The IEC 61131-3 Software Model shows the basic high-level language elements and their interrelationship which consists of programmed elements and configuration elements [7, p.21 4.2]. The Subject-oriented Code Generation Model is

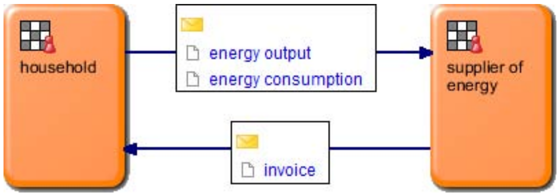



**Fig. 6.** BI: household - supplier of energy

a representation of a the Software Model described in Fig. 7 (a). Therefore it is important to distinguish between *programming* and *configuration elements* (in context of S-CGM: Subjects). In order to be able to communicate with a PLC, basic configurations need to be carried out. Therefore the Subject-oriented Configuration Model (S-CM) and Subject-oriented Task Model (S-TM) as well as the Subject-oriented Hardware Model (S-HWM) is established. A Behavioral Interface (BI), whether it is used for S-CGM or as a communication interface on Layer 2 (S-AFM), is the basic representation form for communication on all layers. The use of jCPEX! AE platform (Sect. 3.1) allows direct interaction between all layers of modern control systems.

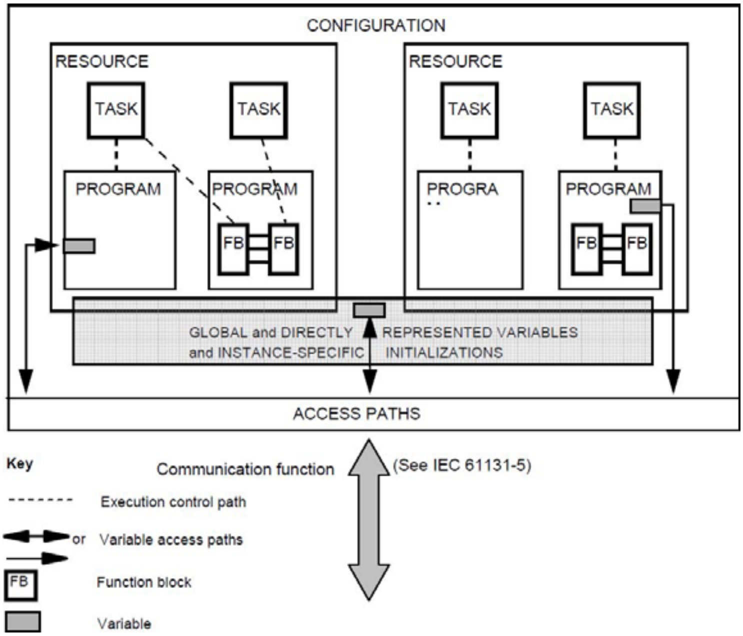
The Subject-oriented Program Organization Unit Model (S-POUM) represents the *programming elements* described in IEC 61131-3 and uses the S-CM and S-HWM.

### 4.1 ***S***ubject-Oriented ***H***ardware ***M***odel (S-HWM)

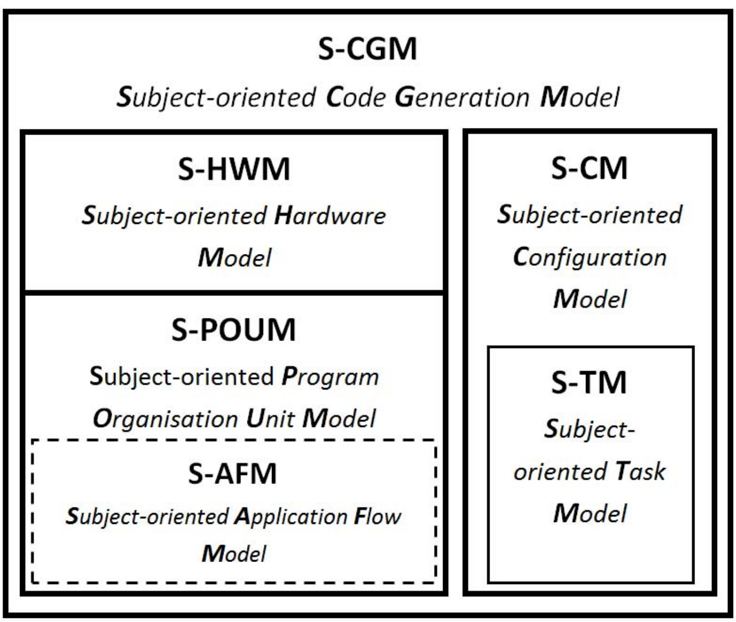
The ***S***ubject-oriented ***H***ardware ***M***odel associates a physical hardware device (sensors, actors) to a PLC I/O represented by a variable. In the domain of S-CGM the corresponding S-HWM represents a *configuration* Subject. It is the part of configuration elements, namely, *configurations*, *resources*, *tasks*, *global-variables*, *access paths*, and *instance-specific initializations* [7, p.21, 4.1], which represent hardware devices logically or physically .

The logical and physical connection of sensors and actuators and the assigned interfaces are available in numerous different documents. Even if the installation is hard-wired, the logical connection has to be documented in a further step which is the basis for advanced process modeling activities. IEC 61499-1 [4] defines a graphical and textual possibility to describe a system configuration which includes assignment of physical to logical port by a XML schema supporting system interoperability. IEC 61131-3 is compliant to IEC 61499 [4, p.85].

The S-HWM consists of sensor interfaces and/or actuator interfaces represented, by a Behavioral Interface, which is provided by jCPEX! AE platform. The exchanged messages are unidirectional or bidirectional depending on the hardware device. A multiple use of a BI, representing only one *DataItem* instance, within the subject interaction diagram has to be allowed because of modeling clarity issues.




(a) IEC 61131-3 - *Software Model* [7]



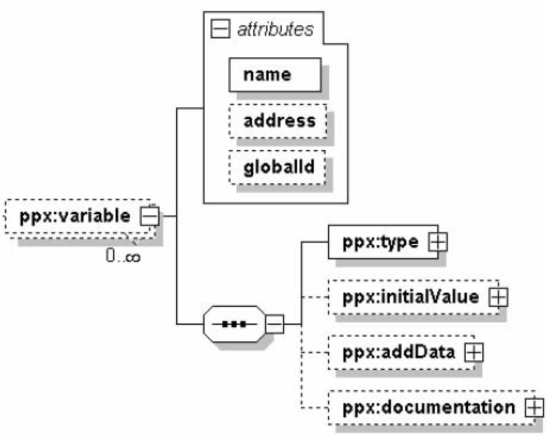

(b) ***S***ubject-oriented ***C***ode ***G***eneration ***M***odel

**Fig. 7.** IEC 61131-3 and corresponding subject-oriented model

| No. | Feature/examples | |
|---|---|---|
| 1a | **Declaration of locations of symbolic variables – AT keyword** | |
| | VAR_GLOBAL<br>LIM_SW_S5 AT %IX27 : BOOL;<br>END_VAR | Assigns input bit 27 to the Boolean variable LIM_SW_5 (NOTE 2) |
| | VAR<br>CONV_START AT %QX25 : BOOL;<br>END_VAR | Assigns output bit 25 to the Boolean variable CONV_START |
| | TEMPERATURE AT %IW28: INT; | Assigns input word 28 to the integer variable TEMPERATURE (NOTE 2) |
| | VAR<br>C2 AT %Q* : BYTE;<br>END_VAR | Assigns not yet located output byte to bitstring variable C2 of length 8 bits |
| 7b | **Declaration of enumerated variables** | |
| | VAR<br>Y : (Red, Yellow, Green);<br>END_VAR | Declaration of an enumerated variable |
| 8c,d | **Declaration of subrange variables** | |
| | VAR<br>Z : SINT(5..95);<br>END_VAR | Declaration of a subrange variable |

(a) IEC 61131-3 - variable assignment [7]

diagram:




attributes

| Name | Type | Use | Default |
|---|---|---|---|
| name | xsd:string | required | |
| adress | xsd:string | optional | |
| globalID | xsd:ID | optional | |

(b) PLCopen XML specification: *variable* [16]

**Fig. 8.** IEC 61131-3 and PLCopen *variable*

Figure 9 (a) illustrates the S-HWM. The *PRG light* represents a program organization unit, namely *program* which directly interacts with the I/O variables (digital input) *di light switch*, (analog input) *ai twilight sensor* and (digital output) *do light*. Because of the sensors and actuators used, the communication is uni-directional but, depending on the hardware device, a bi-directional communication is applicable. Depending on the selected hardware device the messages (business objects), sent and received, includes different values e.g. the raw value, the measurement value or value range. The data type of *single-element variables* [7, p.33, 6.3.2] can be mapped or defined whereas *multi-element variables*, namely *array* and *structure* have to be provided by the S-CM. Special to PLCs is the initialization of variables, according to the rules of IEC 61131-3, which can take one of the following initial values [7, p.45, 6.4.3]:

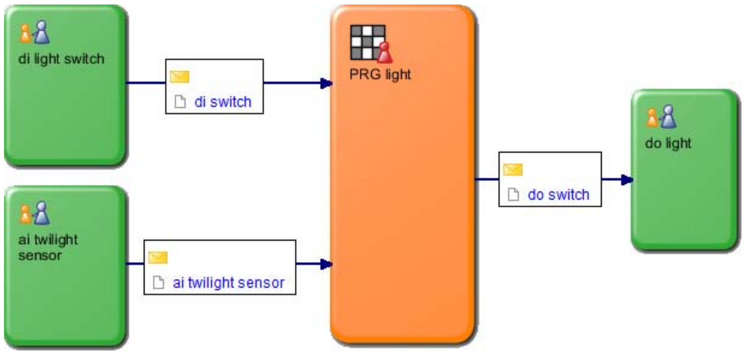
- *the value the variable had when the configuration element was "stopped" (a retained value)*
- *a user-specified initial value*
- *the default initial value for the variable's associated data type*

The declaration of the variables [7, p.45, Table 18] (e.g. *var*, *var_ input*, *var_ output*, *var_ global*, *var_ access* ...) has to be done automatically by model analyze, during the transformation process.

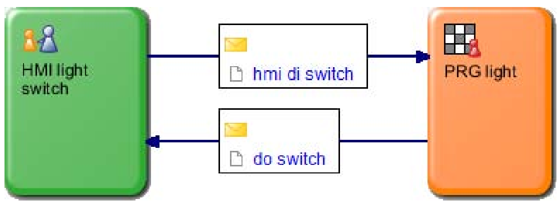
### 4.2 *S*ubject-Oriented *C*onfiguration *M*odel (S-CM)

The ***S***ubject-oriented ***C***onfiguration ***M***odel represents the *configuration elements* defined in part 6.7 of IEC 61131-3 [7]: "*A configuration consists of resources, tasks (which are defined within resources), global variables, access paths and specific initializations.*" In context of subject-oriented process modeling the configuration elements include interfaces to external applications e.g. Human-Machine Interface (HMI), webapplication or other programmable logic controller represented by a BI. Figure 9 (b) illustrates a Subject called *HMI light switch* which is an external HMI. Figure 9 (a) shows the hardware input represented by the single-element variable *di switch* and the hardware output represented by the variable *do light*. These variables are connected to the physical I/O register and represent the physical state of sensors and actuators whereas the Subject *HMI light switch* receives a variable *hmi di switch* which is changed by HMI interaction illustrated in Fig. 9 (b). Therefore *do light* changes the state of *hmi di switch* whenever the light status changes.

The ***S-TM*** (***S***ubject-oriented ***T***ask ***M***odel) is part of S-CM and defines the process execution. For the purposes of IEC 61131-3 [7, p.131, 6.7.3] a task is defined as: "*... an execution control element which is capable of calling, either on a periodic basis or upon the occurrence of the rising edge of a specified Boolean variable, the execution of a set of program organization units, which can include programs and function blocks whose instances are specified in the declaration of programs.*" [7, p. 131, 6.7.3] According to the definition of programming-language standards in IEC 61131-3 textual and graphical notation is available for definition of tasks.




(a) ***S***ubject-oriented ***H***ardware ***M***odel




(b) ***S***ubject-oriented ***C***onfiguration ***M***odel - HMI interface

**Fig. 9.** Subject-oriented communication with external hard- and software

## 4.3 *S*ubject-Oriented *P*rogram *O*rganisation *U*nit *M*odel (S-POUM)

The ***S***ubject-oriented ***P***rogram ***O***rganisation ***U***nit ***M***odel represents the *function*, *function block* and *program* defined in the IEC 61131-3 [7, p.51, 6.5]. POUs are *programmed elements* and can be delivered by the manufacturer, or programmed by the user by the means defined in the IEC 61131-3.

A *program* is defined in IEC 61131-1 as a "*logical assembly of all the programming language elements and constructs necessary for the intended signal processing required for the control of a machine or process by a programmable controller system.*" [3]. *Program* as well as *function*, in context of S-POUM is represented by a *Internal Subject*. A *function* is defined as a program organizational unit (POU) which, when executed, yields no (VOID) or exactly one data element, which is considered to be the function result. [2]

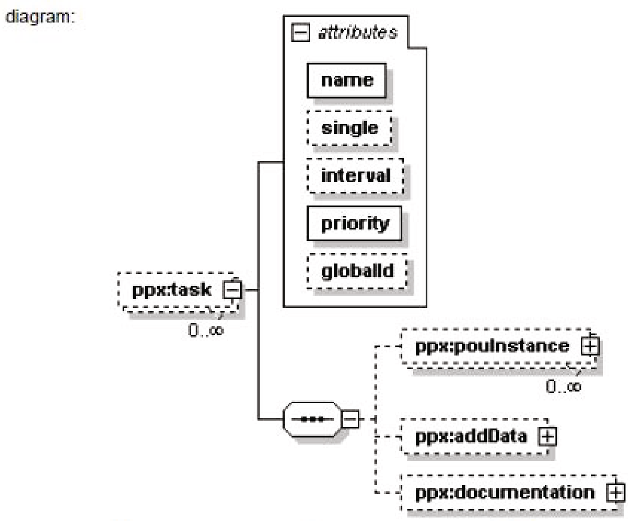
A *function block* is represented by a *Multi Subject* which, "*when executed, yields no or exactly one data element, which is considered to be the function block result (like a function), and one or more values which are considered to be the function block outputs.*" [7, p. 74, 6.5.3.1] Multiple named instances (copies) of a function block type can be created on condition that each instance has an associated identifier (the instance name). Different to *functions*, the variables of a *function block* shall persist from one execution of the *function block* instance

| No. | Description/Examples |
|---|---|
| 1a | Textual declaration of periodic TASK (feature 5a of Table 57) |
| 1b | Textual declaration of non-periodic TASK (feature 5b of Table 57) |
| | Graphical representation of TASKs (general form) |

```
          TASKNAME
        +---------+
        |  TASK   |
BOOL---|SINGLE   |
TIME---|INTERVAL |
UINT---|PRIORITY |
        +---------+
```

```
1   CONFIGURATION CELL_1
2     VAR_GLOBAL  w: UINT;  END_VAR
3     RESOURCE STATION_1 ON PROCESSOR_TYPE_1
4       VAR_GLOBAL  z1: BYTE;  END_VAR
5a      TASK SLOW_1(INTERVAL := t#20ms, PRIORITY := 2) ;
5a      TASK FAST_1(INTERVAL := t#10ms, PRIORITY := 1) ;
6a      PROGRAM P1 WITH SLOW_1 :
8a                   F(x1 := %IX1.1) ;
9b      PROGRAM P2 : G(OUT1 => w,
6b                     FB1 WITH SLOW_1,
6b                     FB2 WITH FAST_1)  ;
```

(a) IEC 61131-3 - *task* graphical and textual [7]




attributes:

| Name | Type | Use | Default |
|---|---|---|---|
| name | xsd:string | required | |
| single | xsd:string | optional | |
| interval | xsd:string | optional | |
| priority | derived by: xsd:integer | required | |
| globalId | xsd:ID | optional | |

(b) PLCopen XML specification: *task* [16]

**Fig. 10.** IEC 61131-3 and PLCopen *task*

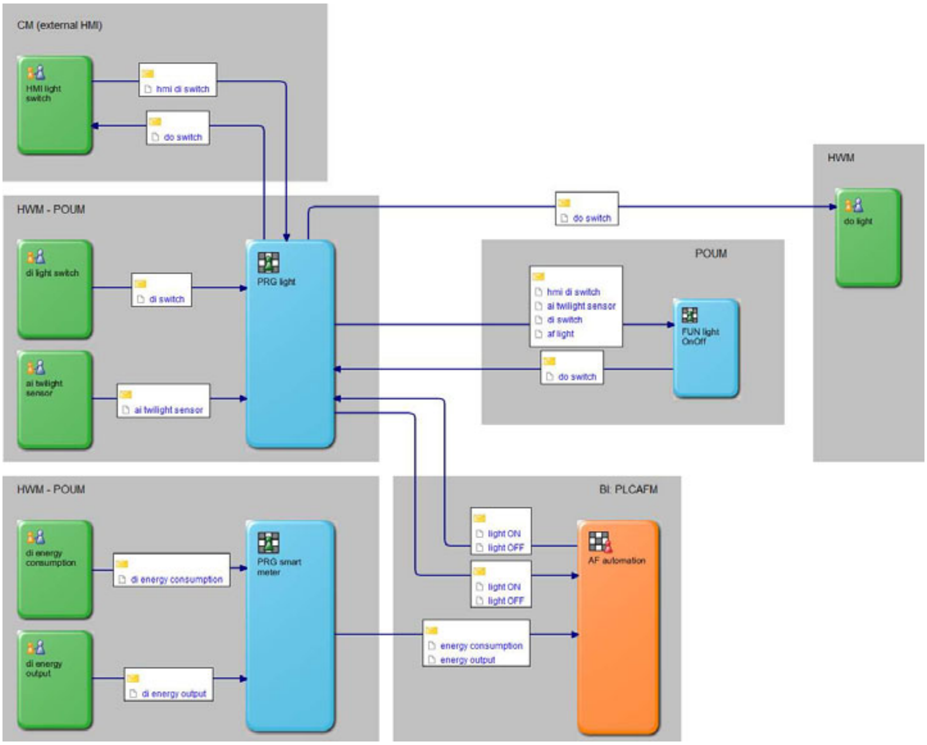



**Fig. 11.** ***S**ubject-oriented **P**rogram **O**rganisation **U**nit **M**odel*

to the next [7]. The messages exchanged between Subjects are the corresponding variables in IEC 61131-3. A *Business Object* is represented by a single-element or multi-element variable [7, p. 43, 6.4.2.4]

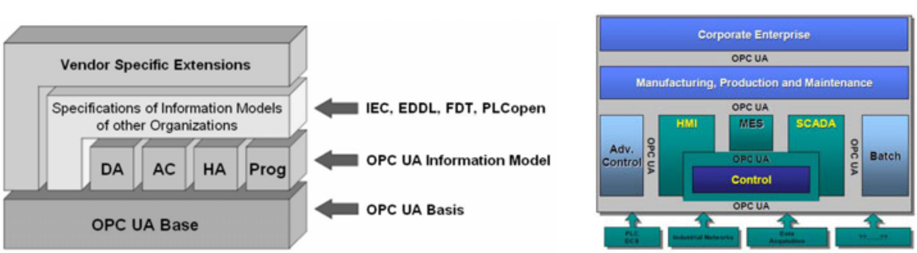
Figure 11 shows the S-POUM including the S-HWM, the S-CM and the BI for the Subject-oriented Application Flow Model. *PRG light* and *PRG smart meter* are two independent *programs* using the same BI for process communication with the S-AFM. The execution of the programs is configured in the Subject-oriented Task Model described in Sect. 4.2. The subject interaction diagram shows the communication structure of the PLC program. *PRG light* calls a function *FUN light OnOff* with a parameter list including the digital input *di switch*, the analog input *ai twilight sensor*, the external HMI variable *hmi di switch* and a variable *af light* which is a message out of the S-BPM represented by *light ON* and *light OFF*. The return value of function *FUN light OnOff* is the digital output *do light* which is a physical output and can not be declared twice. Therefore the variable *do light* just represents the return value type of the function. Figure 11 illustrates a IEC 61131-3 application using a subject oriented representation format. The subject interaction diagram represents the structure and the configuration of an application whereas a subject behavioral diagram represents the programming logic which is not shown in this document.

The use of a vendor specific application element, e.g. library or POU, is possible by importing them using the PLCopen XML standard format. The messages exchanged represent the parameter list, in context of IEC 61131-3 applications: input, output, and input-output variables.

## 5 Subject-Oriented Application Flow Model (Layer 2: AFM) and S-BPM Communication (Layer 1)

Depending on the execution environment (PLC or Metasonic Flow) the transformation of the subjects is different. The S-CGM represents a IEC 61131-3 application using the hardware I/Os directly represented by a variable and the S-POUs represent IEC 61131-3 POUs. Whereas the Subject-oriented Application Flow Model can not use PLC I/Os directly because the S-AFM runs in Metasonic Flow and not on the PLC which is therefore used as hardware I/O interface. There are many possibilities to exchange data between a PLC and an IT system. The three-layer SCADA architecture using open system technology, allows, by using the OPC Unified Architecture, vertical data integration. The OPC Unified Architecture (OPC UA) will be known as IEC 62541 standards.

Figure 12 (a) shows the different layers of information models defined by OPC (OPC UA Basis, OPC UA Information Model) by other organizations (IEC, EDDL, FDT, PLCopen), or by vendors. The base specification covers OPC UA part 1- 7 including all known features from Classic OPC. OPC Unified Architecture is divided into 7 *Core Specific Parts* (part 1-7) and 5 *Access Type Specification Parts* (part 8-12). *Data Access* (DA, part 8) defines automation-data-specific extensions and allows a link to live automation data. Part 9 of the OPC UA is Alarm & Conditions (AC) which specifies an advanced model for process alarm management and condition monitoring. Part 10 Programs (Prog) specifies a mechanism to start, manipulate, and monitor the execution of programs. A mechanism to access historical data and historical events is defined in part 11 Historical Access (HA). [12, p.11]




(a) OPC UA Layered Architecture

(b) OPC UA Target Applications [6]

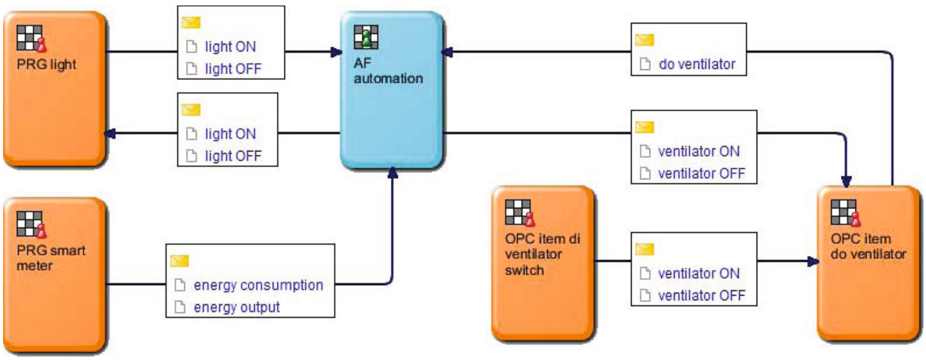
**Fig. 12.** OPC Unified Architecture

In context of S-AFM the Behavioral Interface illustrates the *DataItem* which is a link to arbitrary, live automation data (e.g. device data, calculated data, status information, dynamically-changing system data, diagnostic data) [5, p.8, 3.4]. The exchanged *message* (*business object*) represent two types of Variables, *Properties* and *DataVariables* [5, p.17] which differ in the kind of data they represent. Properties are server-defined characteristics of Objects, DataVariables and other Nodes, whereas *DataVariables* represent the content of an Object. [8, p.17] Figure 13 illustrates the subject-oriented Application Flow Model and its representation in the subject-oriented business model. *PRG light* and *PRG smart meter* represent the S-POUM and the data exchanged between programmable logic controller and the *AF automation* subject. The variable *OPC item di ventilator switch* represents a physical Input on the PLC which variable type is boolean. The *OPC item do ventilator* is an output variable representing a physical device. The subject interaction diagram shown in Fig. 13 (a) illustrates the subject interaction. A subject behavior diagram might show the process of switching on/off the ventilator using a subject oriented modeling methodology. The subject *AF automation* is the behavioral interface linking layer S-AFM and S-BPM shown in Fig. 2.

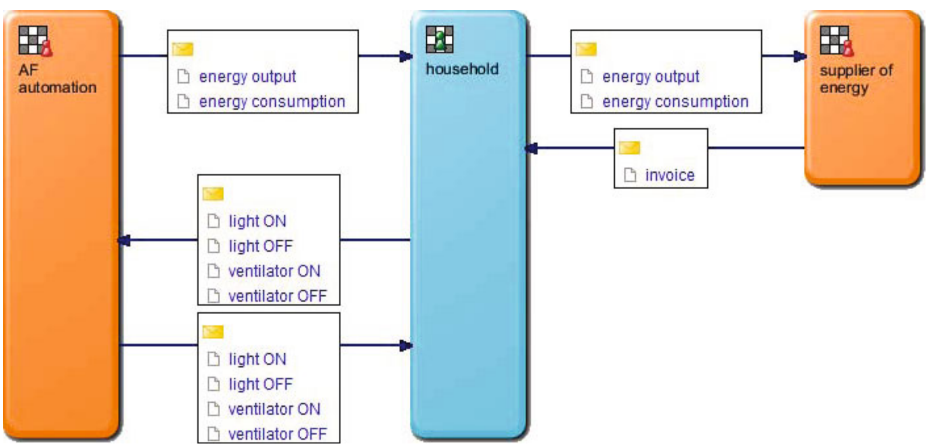
## 6 Summary and Issues for Further Research

A Behavioral Interface (BI) as shown in Fig. 6 "*... describes the observable behavior of the participation processes and their communication via message exchange. It can be automatically extracted from a partner's private process or modeled separately from scratch.*" [13, p.176] The use of a BI in combination with the jCPEX! platform allows cross-organizational business processes e.g. the communication between household and energy supplier who are exchanging the messages *energy output/consumption* and *invoice* as described in Sect. 3.2. A BI can also be used for mapping a *DataItem* (a link to arbitrary, live automation data) to a *business process subject*. A *DataItem* or a group of *DataItems* can be illustrated as a *Subject* used in a business process. The *messages* (*business objects*) exchanged between the two *external subjects* are the *DataItem* instances. To be able to communicate between a control layer device and a business process using vertical data integration, an application needs to be implemented for mapping the *DataItem(s)* to a *Subject* which can be done by *jCPEX! Automation Extension (AE) platform* including a *RuleEditor* which "*facilitates replacement of a partner dynamically dependent on certain conditions - even at runtime*". [13, p.176] In this context routing would not mean cross-organizational routing but rather inter-layer routing and the *replacement of partner* is on the one hand a rule-based adoption of automation processes influenced by a business process and on the other hand a rule-based initialization/instantiation of different business processes according to control layer data.

This *jCPEX! Automation Extension (AE) platform* is the basis for using a Subject-oriented Application Flow Model (S-AFM) which enables the control layer device act as an I/O interface, and use Metasonic Flow as workflow engine.




(a) ***S***ubject-oriented ***A***pplication ***F***low ***M***odel




(b) ***S***ubject-oriented ***B***usiness ***P***rocess ***M***odel

**Fig. 13.** Subject-oriented communication between S-AFM and S-BPM

A Human-Machine Interface (HMI) as part of the Subject-oriented Configuration Model (S-CM) might also use this platform if a SOA supports the communication between HMI and jCPEX! platform and supports the mapping between *HMI interface* and a *Subject*. A *RuleEditor* can distribute incoming events to one or, if *MultiPartyBIs* are supported, more *Subjects*. Another possibility would be to create a Subject-oriented Application Flow Model using the *jCPEX! AE* to dispatch messages to one or more business process Subjects, HMIs or *DataItems* represented by subjects.

The process control layer (RTUs/PLCs) (*Control Layer*) is enhanced by providing the ability to model PLC application code in a subject-oriented way. In a first step not the detailed control process (provided by the vendor of a machine or implemented by a specialist) is focused on, but the IEC 61131 *Configuration -, Communication - and Software Model* (illustrated as subject interaction diagram) are in the center of interest. The logic of an application implemented in the body of *programs*, *function blocks* and *functions* was not treated in this paper, but could be modeled by using the subject behavior diagram.

The integration of POUs (libraries, external code, ...), using a *Subject* or a *Multi Subject*, is adequate to illustrate processes. MDSE is possible because PLCopen provides a XML specification for the IEC 61131-3 standard, the only standard in industrial control programming, and the OPC Foundation provides the communication basis with the OPC Unified Architecture (draft standard IEC 62541) therefore a *BPM system* additionally has to provide *code-transformation ability* and the *ability to interact with these generated applications*.

Summing up the approach described, subject-oriented methodology can be used for business process modeling as well as IEC 61131-3 automation modeling. In combination with jCPEX! Automation Engineering (AE) platform and OPC UA an inter-layer and/or cross organizational use is applicable.

## References

1. Cupek, R., Fojcik, M., Sande, O.: Object Oriented Vertical Communication in Distributed Industrial Systems. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2009. CCIS, vol. 39, pp. 72–78. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-02671-3_8, doi:10.1007/978-3-642-02671-3_8
2. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen, iCS 25.040.40; 35.060; 35.240.50 (2003)
3. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: Speicherprogrammierbare Steuerungen - Teil 1: Allgemeine Informationen (IEC 61131-1:2003); Deutsche Fassung EN 61131-1:2003, iCS 25.040.40; 35.240.50 (2004)
4. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: Function blocks - Part 1: Architecture (IEC 61499-1:2005); German version EN 61499-1:2005, iCS 25.040.40; 35.240.50 (2006)

5. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: OPC Unified Architecture - Part 8: Data Access (IEC 65E/98/CDV:2008); English version FprEN 62541-8:2008, iCS 35.200; 35.240.50 (2008)
6. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: OPC Unified Architecture - Teil 1: Übersicht und Konzepte (IEC 65E/92/CDV:2008); Englische Fassung FprEN 62541-1:2008, iCS 35.200; 35.240.50 (2008)
7. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: Programmable Controllers - Part 3: Programming languages; English version (IEC 65B/725/CD:2009), iCS 35.080 (2009)
8. DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE: OPC Unified Architecture - Teil 3: Adressraummodell (IEC 62541-3:2010); Englische Fassung EN 62541-3:2010, iCS 35.200; 35.240.50 (2011)
9. Endi, M., Elhalwagy, Y., Hashad, A.: Three-layer plc/scada system architecture in process automation and data monitoring. In: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 2, pp. 774–779 (2010)
10. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010), http://www.springerlink.com/content/m4105554174112q7/
11. Frey, G., Thramboulidis, K.: Einbindung der iec 61131 in modellgetriebene entwicklungsprozesse, http://www.aut.uni-saarland.de/uploads/media/GF_KT_AUTOMATION_JUNE_2011.pdf
12. Mahnke, W., Leitner, S.H., Damm, M.: OPC Unified Architecture. Springer, Heidelberg (2009)
13. Meyer, N., Radmayr, M., Heininger, R., Rothschädl, T., Fleischmann, A.: Platform for Managing and Routing Cross-Organizational Business Processes on a Network Router. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 175–189. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-23471-2_13, doi:10.1007/978-3-642-23471-2_13
14. Opc foundation (October 25, 2011), http://www.opcfoundation.org
15. Plcopen (October 25, 2011), http://www.plcopen.org
16. PLCopen: XML Formats for IEC 61131-3, 2.01 edn. (2009)
17. Selic, B.: From model-driven development to model-driven engineering. In: Euromicro Conference on Real-Time Systems, p. 3 (2007)
18. Selic, B.: Personal reflections on automation, programming culture, and model-based software engineering. Automated Software Engineering 15, 379–391 (2008), http://dx.doi.org/10.1007/s10515-008-0035-7, doi:10.1007/s10515-008-0035-7

# Using Social Network Analysis and Derivatives to Develop the S-BPM Approach and Community of Practice

James E. Weber[1], Werner Schmidt[2], and Paula S. Weber[1]

[1] St. Cloud State University, St. Cloud, MN, USA
{jweber,psweber}@stcloudstate.edu
[2] University of Applied Sciences Ingolstadt, Esplanade 10, 85049 Ingolstadt, Germany
werner.wchmidt@haw-ingolstadt.de

**Abstract.** Subject-Oriented Business Process Management (S-BPM) and Social Network Analysis (SNA) are emerging as valuable process improvement techniques within the broader field of Business Process Management (BPM). S-BPM focuses on how subjects perform actions on objects and on how they exchange messages in order to coordinate their process work. SNA and its derivatives look at the interrelationships between actors or nodes within a social network. The use of SNA to complement process improvement techniques is a new approach for BPM, but the focus thus far has been primarily within and between organizations. This paper proposes that SNA can be usefully engaged in the examination of interrelationships in the developing field of S-BPM and that SNA techniques can also help further develop the S-BPM approach. Potential benefits and directions for future research on the use of SNA within the community of practice and field of S-BPM are discussed.

**Keywords:** social networking analysis (SNA), subject-oriented business process management (S-BPM), development of the field of S-BPM, community of practice in S-BPM, S-BPM implementation, S-BPM research, value network analysis (VNA), organizational network analysis (ONA), value network model.

## 1 Introduction

As organizations seek greater effectiveness and efficiency in their operations, BPM, with its orientation toward controlling operational processes, seeks to play an increasingly important role. Subject-oriented Business Process Management (S-BPM) offers the promise of reorienting processes toward the subject (effectiveness) and enhancing the integration of new applications into the existing IT infrastructure (efficiency) [1].

We consider Social Network Analysis to be a means to develop the field of S-BPM in a twofold manner (see figure 1):

(1) In its original form SNA provides a tool for examining patterns of interactions within organizations and within a more broadly defined field of study or community of practice like the relatively young field of S-BPM [2].

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 205–217, 2012.

(2) SNA derivatives such as Organizational Network Analysis (ONA) and Value Network Analysis (VNA) can be utilized to help further develop the (S-) BPM approach [3]. See Figure 1 which depicts this relationship.

Before we discuss these ideas in sections 4.1 and 4.2 we present fundamentals of S-BPM and SNA in sections 2 and 3. The article ends with suggestions for future work in section 5.

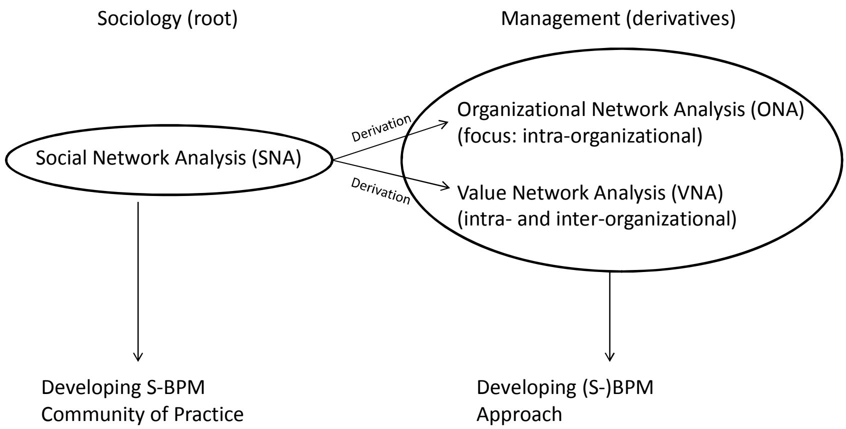



**Fig. 1.** The Use of SNA and Derivatives for Developing the S-BPM Community and S-BPM Approach

## 2 Characteristics of S-BPM

The field of Subject-oriented Business Process Management can be described as focusing on distributed processes in which Subjects (actors) perform Actions on Objects. It differs from traditional Business Process Management (BPM) in its essential focus on both the subjects and their communications as they drive the implementation of every-day activities, as well as the relatively simple integration of user-developed applications directly into existing IT systems [1].

Creating S-BPM models includes the following steps:

(1) Identify the processes in the organization (process map showing processes and their relationships).

(2) Define the subjects involved in a process and their interactions including the messages they exchange (subject interaction diagram showing the communication structure).

(3) Define the subject behavior (subject behavior diagram showing the steps to fulfill a task according to business rules).

(4) Define information processed and exchanged by the subjects via messages (business objects).

Figure 2 depicts the subject interaction diagram (upper part) and the behavior diagram (lower part) for one of the subjects involved in the process for requesting and approving business trips in an organization.

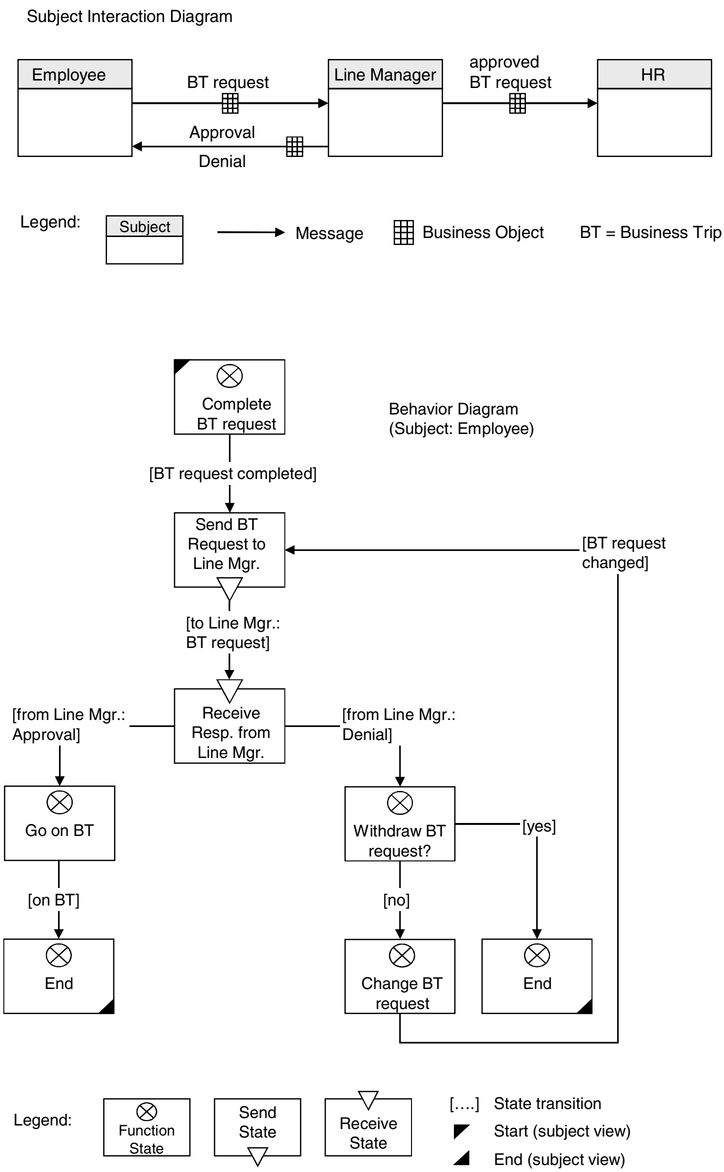



**Fig. 2.** Subject Interaction Diagram and Subject Behavior Diagram

The S-BPM modeling language includes a subject, a predicate and an object, thus using all the building blocks of a complete sentence in natural language. Its graphical notation is based on a process algebra with a clear formal semantic which allows automated code generation [4]. This makes process descriptions executable and empowers process stakeholders to instantly validate the model and model changes without having IT specialists involved. S-BPM allows seamless round-trip engineering, helps keep models and their implementation consistent, and thus avoids the necessity of maintaining both a business process model and its technical representation in the process engine [5, 6].

In this regard, S-BPM goes beyond being just another modeling language. It is a shift in paradigm towards stakeholder-oriented BPM. This means organizational development can be driven by all members of the organization. The easy-to-understand modeling language has only a few elements (see figure 2) and its resulting executable models empower process participants and knowledge workers in particular to (re-) design processes and to immediately experience the changes. Suitable governance rules assure the conformity of development steps with the overall objectives of the process, business unit and enterprise. Four roles describe how S-BPM organizes business development in a participative manner [5]:

**Actors** are those who are working in processes. They represent the subjects in S-BPM models. According to the S-BPM concept actors are the focal point in all BPM activities as there are analyzing, modeling, validating, optimizing, implementing, running and monitoring processes. If actors recognize weaknesses in a process they are working in, they are not limited to triggering organizational development processes to improve the situation. They can drive those steps by themselves with the support of facilitators, experts and governors if needed.

**Facilitators** guide actors through the activities of an organizational development cycle. They help them initiate changes, and also involve the right experts and the right governors as necessary. Facilitators foster communication among all stakeholders of the particular process and act as knowledge brokers. They can be considered as catalysts of organizational development who guide and ease the process along. They may help developing the participants' business and personal qualifications in this process. Examples of facilitators are experienced project leaders, coaches, organizational change agents, service desk managers.

**Governors** are subjects who set the framework for business process management in terms of governance and compliance. They influence the development of the organization as well as its operations. Governor roles can exist on all levels of hierarchy and in many different domains. Examples are: the CEO level defines the corporate or business unit strategy, middle management is responsible for functional strategies, the process owner defines process performance indicators and sets goal values for them, and the organization department makes decisions regarding BPM methods, tools and conventions to apply.

**Experts** are domain specialists who can be involved by the other roles if special expertise seems to be needed to solve a problem. Typical examples for experts are process consultants, business analysts, IT architects, and software developers etc.

With the situation-based collaboration of these roles, the S-BPM approach facilitates organizational learning, using “business process models as transformation enablers” [6].

## 3 Characteristics of SNA

Social Network Analysis (SNA) can be described as the measurement and mapping of relationships and interactions between and among actors (subjects or nodes) in a social network [7]. SNA originated in studies of the social psychology of groups, but its continued development arose from its adoption by the sociology and social anthropology research communities [8]. More recently, disciplines such as economics [9] and organizational studies [10] have begun to use SNA. At their heart, most SNA software programs provide researchers with tools to examine the nature of relationships within a social network, as conceptualized by those disciplines.

Outputs from SNA software programs typically focus on a graphical depiction of studied relationships between nodes or actors within the network (a sociogram), as well as statistics describing relevant aspects of those relationships. For a network of individuals, where people comprise the actors, subjects, or nodes in the network, the sociograms represent the various connections between and among people. In focusing on the structure of the relationships between nodes or actors, SNA differs from traditional research or analytical techniques that treat the actors or nodes as primary targets for analysis.

Researchers can focus on various aspects of the network’s relationships. Some focus on the roles of actors in the network [8], including roles that actors play as providers or users of information, communication resources, or as links between or bridges of gaps between subgroups. Other research focuses on aspects of the network relationships such as type of transactions between actors, authority/power relationships between actors, or the instrumental nature of actor relationships [11].

Depending on the types of questions asked, social network analysis can be helpful in examining communities of practice (networks) with respect to: how easy it is to disseminate information within the network (communications networks), who is an information resource (resource networks), who connects different groups within the network or outside the network (linkage and bridging functions), where communication bottlenecks exist within the network (communication deficiencies), etc. To emphasize a point, in dealing with these issues social network analysis doesn’t typically measure directly assess areas of interest to traditional research (e.g. transfer of knowledge). What SNA does is assess the nature of relationship between actors involved in or observing transfer of knowledge. The researcher then can use this information to improve, in this instance, information transfer processes.

There are two basic types of networks analyzed by SNA, personal or egocentric networks and whole or complete networks. Egocentric networks depict the ties that specific people have, while complete networks examine all the specified ties within a defined population [12]. Organizational researchers tend to concentrate on whole networks in their work, though egocentric networks are sometimes examined to further explore findings from the examination of whole networks.

SNA is very efficient in terms of the number of questions needed to elicit data to analyze a network, but it is very inefficient with regard to the amount of information required as the network grows larger. Dunbar's number provides an upper limit of approximately 150 actors for egocentric networks (13) but whole networks can be much larger. For example, when examining a complete network, one might only ask one or two questions, but ideally that information would be needed for every dyadic relationship between actors in the network. A complete network of N actors has ($N^2$-N) dyadic relationships, as self-relations are generally ignored [14]. If actors are people, the question that elicits data could be asked once with a checkbox or Likert-type scale appended beside every actor in the network. Once a network grows beyond a certain point, this kind of sampling becomes awkward and infeasible.

# 4 Utilizing SNA, ONA and VNA in the S-BPM Context

## 4.1 Utilizing SNA within the S-BPM Community of Practice

S-BPM is a relatively young field, with its first international workshop held in 2009. A search of available literature identified three collections of papers from three initial conferences on S-BPM (2009 [15], 2010 [16] and 2011 [17]), some articles [11, 14, 18, 6] plus a single book on the topic [5]. Gartner suggests that S-BPM is a discipline on the rise, though still lacking 10 years until mainstream adoption [2]. As a young, developing field, it is focused on exploring and explicating the technological, community and methodological foundations and definitions of the field [19]. Buchwald's 2009 paper succinctly outlined some of these basic goals and activities [20]. Still, S-BPM is in the initial stages of documenting and disseminating its actual practices and lacks the evolved infrastructure, established interconnections and communications of a traditional field [2].

Most fields grow haphazardly with unstructured and inefficient patterns of interaction and communication. A systematic examination of the patterns of interaction within an emerging field might lead to a better understanding of how to optimize these patterns for maximal dispersion of information and field growth.

One can observe that new academic fields also tend to exhibit characteristics known in market theory as information asymmetry, where necessary information is not distributed equally across actors in the field. Though firms in market conditions typically attempt to retain aspects of information asymmetry to preserve competitive advantage, academic fields and communities of practice benefit from the dissemination of this information. Over time, an academic field matures and access to information tends to become more equally distributed. One of the problems facing young fields like S-BPM is how to decrease information asymmetry so that as the field grows, academics have access to publications needed to advance theory and practitioners have access to information necessary to advance practice. Historically both groups need to achieve a critical mass of participants in order for the field to prosper.

While S-BPM is developing as a field, it is also in the early stages of expanding its community of practice. Though Buchwald recognizes Community Process in his typology for Community, he focuses on publications and a process for overseeing evolving S-BPM standards. Arguably the concept of community could usefully be broadened to include those interested and involved in the expansion of users and support for the S-BPM process. This argument is recognized implicitly by Schmidt and Stary [21] in their call for an S-BPM community of practice oriented toward altering the mindset of practitioners and researchers to one that recognizes the centrality of the actor in business process modeling (see www.i2pm.net). While S-BPM is in the process of defining its basic parameters as a field, it is simultaneously growing its community of practice. SNA and its various derivatives provide a variety of analytical tools that complement the goals of BPM and facilitate both of these processes.

The use of SNA to depict various relationships between and among S-BPM community members has many potential benefits, though not all benefits accrue to every situation. An examination of core SNA metrics can help to illustrate how the use of SNA to evaluate interconnections within the emerging field and community of practice of S-BPM might be advanced.

Example SNA Metrics are:

- Bridge – a bridge is a link between two nodes within a network that, if removed, would leave those two nodes in different, isolated portions of the graph. In S-BPM, the identification of a bridge shows a critical path whose absence would tend to isolate S-BPM community of practice members.
- Centrality – this metric provides a rough measure of the degree of influence an actor has. A related measure, eigenvector centrality provides a numerical score indicating the importance of the actor to network functioning. If you remove this person from the S-BPM community of practice, how much is the community weakened? Centrality provides a response to that question.
- Closeness – is a measure of the proximity of an actor to all other individuals in a network. Can the S-BPM community of practice count on members getting news through the "grapevine" or are special efforts necessary to reach everybody?
- Prestige – the degree to which an actor is central or peripheral to the network. Is this S-BPM community member embedded within the community or are they liminal to the community?
- Structural cohesion – the number of members that could be removed from a group before it is disconnected. How stable is the S-BPM community? How can we make it more connected and more stable?
- Structural hole – a lack of connection between portions of a sociogram. In the S-BPM community, we can ask if it is necessary to insert an interconnection, and if so, how.

Additionally the shape of a network has an impact on the usefulness of the network to network members. Compared to small, tight networks with many internal connections and few outside connections, looser networks with more outside ties are better at disseminating information and ideas. This idea is similar to Granovetter's discovery that in social networks, weak ties form strong bridges [22]. People within a tight network tend to share the same resources and knowledge. In order to gain access to new information, access to persons outside the tight-knit group is necessary. Dissemination of information and practices in both directions is aided by looser networks with more outside connections.

To recapitulate, SNA can potentially allow researchers to identify S-BPM community of practice network deficiencies in ways that are actionable by network members. The use of SNA can help to identify resources within the network, potential information deficiencies within the network, bottlenecks for information flow, unused or underutilized resources, communication patterns that could be improved, and useful contact points for those outside the network. SNA also has the potential to identify second-level links between actors that the actors themselves are unaware of and for which the combination of multiple sociograms could be informative. Finally, the results can also provide a historical record of the S-BPM movement at different points in time. These benefits can be highly useful to practitioners interested in S-BPM implementation and researchers interested in measuring the impact of their activities.

In order to implement SNA within the S-BPM community of practice it would be necessary to survey a significant portion of that community. For a young field like S-BPM, the single, well-attended conference would be a likely candidate for data collection. A survey consisting of 6-10 questions, a list of conference participants, and space for filling in additional names for each question would be sufficient to provide analyzable data for researchers, but response rates in the neighborhood of 40-50% would be desirable for meaningful results.

Figure 3 depicts some of the benefits that could potentially be gained by using Social Network Analysis on the S-BPM Community of Practice.

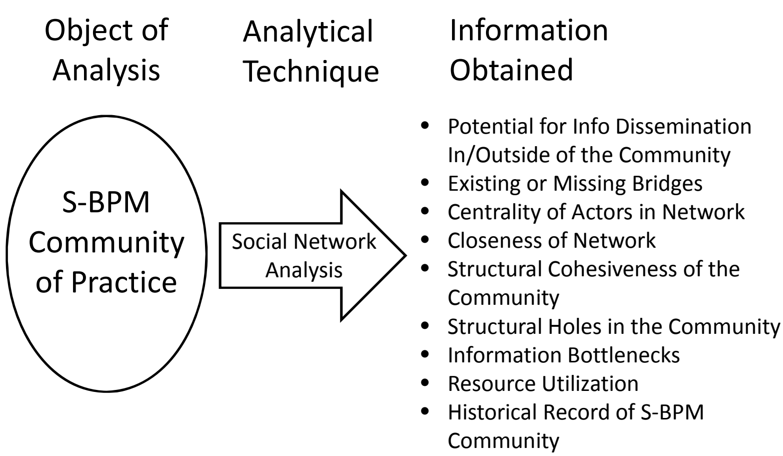



**Fig. 3.** Outcomes of using SNA on the S-BPM Community of Practice

In order to actually realize these benefits, dissemination of the results throughout the S-BPM community of practice is necessary. Several avenues for this dissemination are possible. Again, the single, well-attended conference is a prime candidate for dissemination through conference calls, presentations at the conference, informational handouts, or postings on the conference host's website. In the future, as the field develops, practitioners journals might publish updated results as a service to the community. The goal would be a truly informed S-BPM community of practice.

## 4.2 Utilizing SNA, ONA and VNA to Complement the S-BPM Approach

Organizational Network Analysis (ONA) and Value Network Analysis (VNA) can be considered as special, derivative forms of Social Network Analysis.

ONA applies SNA principles in order to identify information and knowledge flows and informal relationships among individuals in organizations from a business point of view [6, 23].

The goal of Value Network Analysis is to identify multiple interactions between roles in any kind of intra- or inter-organizational networks. Roles are the contributing actors (economic agents) who create value and interact to exchange this value as tangible or intangible deliverables [23, 6]. Roles are populated by participants. Network information can be visualized in a Value Network Model (see fig. 4). Allee sees VNA as a technique that "shows business transaction links in a way that eases asset management, cost/benefit analysis, and other management methods such as business process modeling (BPM)" [23].

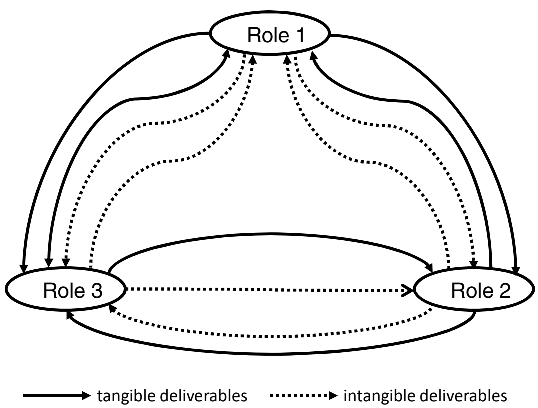



**Fig. 4.** Value Network Model

Gartner suggests that a combination of SNA, ONA and VNA techniques can be used along with traditional business process improvement techniques to help optimize Business Process Management. Gartner's analysts estimate that SNA for BPM is five to ten years from mainstream adoption, but also sees high organizational benefits from adoption in the future [3].

Those benefits can be achieved because the combined techniques allow collection and analysis of additional information on intra- and inter-organizational collaboration not typically obtained by analyzing the formal workflow as is current practice in BPM. Most organizations are replete with examples of informal, often unrecognized or undocumented but necessary processes. Any time that an existing or potential process or communication path is undocumented, there is a possibility it will be overlooked in the formal BPM processes. SNA, ONA and VNA techniques all help to identify processes and value those processes, their actors and their roles. Some of the potential benefits of using SNA with BPM are depicted in figure 5.

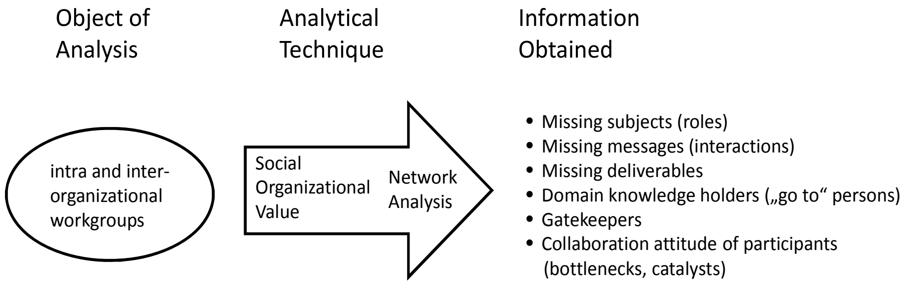



**Fig. 5.** Outcomes of using SNA, ONA and VNA on (S-) BPM

As a communication-based business development technique focusing on subjects and their interactions and fostering self-organization the S-BPM approach is perfectly suited to complementary social network analysis. This leads to benefits that add to what has already been mentioned in figure 5, as combining S-BPM and SNA aspects (e.g. the metrics described in section 4.1) can aid in:

- identifying subjects and interactions for inclusion in S-BPM models (subject and interaction mining). This is important because in reality there usually exist many collaborative processes on an individual basis (especially across organizational borders), which are often totally undocumented and informal. SNA and derivatives can do this by revealing the relationships and placing values on the links. This allows identifying structural roles in the network like central connectors, boundary spanners and information brokers [23]. Those network roles can be both candidates for subjects in business processes as well as for the S-BPM roles (see below).
- populating subjects with the right people (subject representative mining, knowledge and skill mining). As SNA brings hidden processes to light, S-BPM subject – predicate – object relationships may need to be modified with regard to the proper subject for a new combined version of the explicit and hidden processes.
- identifying facilitators, experts and governors with the right knowledge, skills and decision authority necessary for an organizational change driven by actors (S-BPM roles mining, knowledge and skill mining). As SNA uncovers hidden processes, it also identifies these hidden resources that can be used by S-BPM in driving organizational and process change.
- validating existing S-BPM models by comparing/overlaying subject interaction diagrams with the tangible deliverables exchanged by roles in value network models. In instances where VNA identifies informal but valued processes, S-BPM can be used to formalize these processes.

VNA provides several analytical variants [23] which help realize those benefits when applied in the S-BPM lifecycle activities:

**Exchange Analysis** looks for patterns of interactions between roles and for the deliverables (value) being exchanged in the interactions. It supports subject-oriented modeling and validation by identifying subjects in a process and the messages they exchange as well as by recognizing missing or unnecessary subjects and interactions. Exchange analysis also asks for reactions the deliverables trigger and thus helps model and validate the behavior of the receiving subject. The analysis can also help optimize activities in instances where the exchange of intangible deliverables dominates tangible ones. This situation could indicate extensive informal communication taking place to solve structural and behavioral problems occurring along the previously modeled and implemented workflow.

**Value Flow Optimization** uses scenarios to find effective and efficient value flows (workflows) in a completed value network with numerous possible pathways. Parameters for evaluation of different flow options are role execution (subject behavior), transaction speed and channels (implementation and speed of message exchange) and quality or value of deliverables (business objects). Simulating pathways in this way primarily supports the analysis and optimization activities in the S-BPM lifecycle.

**Value Impact Analysis** explores how roles realize gains and benefits from the tangible or intangible value inputs they receive. This analysis needs to be done at the level of individual roles or participants because usually it is not possible to overlook a whole network and it might be satisfactory to just examine problematic roles. It assesses deliverables, their type, where they come from and where they go to (message exchange between subjects and message content, e.g. business objects). It also asks how the input impacts role activities and capabilities (subject behavior), resources and internal assets.

**Value Creation Analysis** looks at how the value network creates value by role and as a whole. Again it is conducted at the role level, but it reverses the analysis by asking about the value of outputs a role (subject) provides to other roles in the network. Like Value Impact Analysis it describes deliverables, but then aims to identify the activities (subject behavior) necessary to create the desired value output.

With the characteristics described above, both Value Impact and Value Creation Analysis primarily help increase efficiency and thus support optimization, but also aid in modeling and validation.

## 5 Future Work

Future research areas regarding the use of SNA and its derivatives in conjunction with S-BPM need to be determined by the S-BPM field. Clearly the use of SNA, VNA and ONA have the potential to complement the use of S-BPM, but the questions of where the value is most evident and the nature of that value are still open. Is it more important to be able to identify the informal processes that are difficult to find using S-BPM or is it more important to be able to examine the nature of the omissions? While S-BPM does a superior job of modeling executable processes, SNA allows the evaluation of some aspects of the quality of the processes. How much value does this approach add to S-BPM? All of these are relevant research questions with regard to the use of SNA, ONA and VNA to complement S-BPM analyses.

With regard to the use of SNA to evaluate the field or the community of practice of S-BPM, it is easily seen that data collection using questions like: "Who do you talk to about S-BPM?", "Who do you go to when you have a problem or question regarding S-BPM?", "Who is an expert in the field of S-BPM?", or "Who possesses expertise in S-BPM implementation?" can provide results useful to both constituencies for a variety of purposes related to the improvement of their internal relationships.

The examination of archival data, including e-mails, blogs or other electronic artifacts provides data bearing on additional questions. The relative youth and small size of the S-BPM movement provides a unique opportunity to track, record and improve the structure of interrelationships within the S-BPM community that will not exist in the same way as the field and community of practice grow larger.

In order to achieve its potential benefits, dissemination of information resulting in an informed S-BPM community of practice is necessary. This proposal assumes that the S-BPM community has an interest in improvements in the areas outlined above…identifying resources within the community, potential information deficiencies within the community, bottlenecks for information flow, unused or underutilized resources, communication patterns that could be improved or contact points for those outside the community an in general decreasing information asymmetry within the S-BPM community of practice. Social Network Analysis is a tool that can facilitate these improvements and promote the growth and development of the field of S-BPM.

## References

1. Metasonic, `http://www.metasonic.de/en/whats-s-bpm` (last access on October 19, 2011)
2. Dixon, J., Jones, T.: Hype Cycle for Business Process Management, 2011. White Paper, Gartner, Inc. (2011)
3. Olding, E., Rozwell, C.: Social Networking Analysis for BPM. In: Dixon, J., Jones, T. (eds.) Hype Cycle for Business Process Management, 2011. White Paper, pp. 23–24. Gartner, Inc. (2011)
4. Schmidt, W., Fleischmann, A., Gilbert, O.: Subjektorientiertes Geschäftsprozess-management, HMD – Praxis der Wirtschaftsinformatik, Heft 266, pp. 52–62 (2009)
5. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement: Mitarbeiter einbinden. Motivation und Prozessakzeptanz steigern, Munich (2011)
6. Fleischmann, A., Stary, C.: Whom to talk to? A stakeholder perspective on business process development. Universal Access in the Information Society, 1–26 (June 2011), doi: 10.1007/s10209-011-0236-x

7. Orgnet.com, http://www.orgnet.com/sna.html
8. Scott, J.: Social network Analysis. Sage Publications, Los Angeles (2009)
9. Manski, C.F.: Economic Analysis of Social Interactions. Journal of Economic Perspectives 14(3), 115–136 (2000)
10. Kilduff, M., Tsai, W.: Social Networks and Organizations. Sage Publications, Los Angeles (2007)
11. Stary, C., Fleischmann, A.: Evidence-based Interactive Management of Change. Knowledge Management & E-Learning: An International Journal 2(1), 17 (2011)
12. Wellman, B., Berkowitz, S.D. (eds.): Social Structures: A Network Approach. Cambridge University Press, Cambridge (1988)
13. Dunbar, R.I.M.: Neocortex Size as a Constraint on Group Size in Primates. Journal of Human Evolution 22(6), 469–493 (1992)
14. Fleischmann, A., Lippe, S., Meyer, N., Stary, C.: Coherent Task Modeling and Execution Based on Subject-Oriented Representations. In: England, D., Palanque, P., Vanderdonckt, J., Wild, P.J. (eds.) TAMODIA 2009. LNCS, vol. 5963, pp. 78–91. Springer, Heidelberg (2010)
15. Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.): S-BPM ONE 2009. CCIS, vol. 85. Springer, Heidelberg (2010)
16. Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.): S-BPM ONE 2010. CCIS, vol. 138. Springer, Heidelberg (2011)
17. Schmidt, W. (ed.): S-BPM ONE 2011. CCIS, vol. 213. Springer, Heidelberg (2011)
18. Fleischmann, A., Stary, C.: Key Features of Subject-Oriented Modeling and Organizational Deployment Tools. In: Stephanidis, C. (ed.) HCII 2011 and UAHCI 2011, Part IV. LNCS, vol. 6768, pp. 205–214. Springer, Heidelberg (2011)
19. Buchwald, H., Fleischmann, A., Seese, D., Stary, C.: Foreword. In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE. CCIS, vol. 85, pp. V-VII. Springer, Heidelberg (2009)
20. Buchwald, H.: Potential Building Blocks of S-BPM. In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 123–135. Springer, Heidelberg (2010)
21. Schmidt, W., Stary, C.: Establishing an Informed S-BPM Community. In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 34–47. Springer, Heidelberg (2010)
22. Granovetter, M.S.: The Strength of Weak Ties. The American Journal of Sociology 78(6), 1360–1380 (1973)
23. Allee, V., Schwabe, O.: Value Networks and the true nature of collaboration, http://www.valuenetworksandcollaboration.com/ (last access on October 19, 2011)

# A Primer to Subject-Oriented Business Process Modeling

Albert Fleischmann[1], Werner Schmidt[2], and Christian Stary[3]

[1] Metasonic AG, Münchner Straße 29, Hettenshausen, 85276 Pfaffenhofen, Germany
Albert.Fleischmann@Metasonic.de
[2] Applied Sciences Ingolstadt, Esplanade 10, 85049 Ingolstadt, Germany
Werner.Schmidt@haw-ingolstadt.de
[3] Johannes Kepler University Linz, Freistaedterstrasse 315, 4040 Linz, Austria
christian.stary@jku.at

**Abstract.** This primer aims towards quick wins in BPM based on a conceptual understanding of S-BPM modeling. We put forward the main constructs in terms of model elements and diagrams. Addressing primarily non-S-BPM-experts we introduce fundamental approaches, namely S-BPM modeling by construction (i.e., when starting from scratch), and S-BPM modeling by restriction (i.e., when refining a generalized peer-to-peer network). Since S-BPM modeling is closely related to standard sentence semantics we start out revisiting essential natural language constructs (subject, predicate, and object) and their role in communication patterns between actors (subjects). We then show how S-BPM constructs can be organized and presented, and introduce interaction and behavior diagrams. Their application ensures coherent representations, and facilitates creating intelligible specifications when business processes are captured from a stakeholder perspective. Finally, we show the essential role S-BPM models play along the open S-BPM life cycle, ranging from process analysis to execution and monitoring.

**Keywords:** business process modeling, subject orientation, model construction.

## 1 Introduction

Process specifications play a crucial role in S-BPM [4]. Focussing on the stakeholder perspective when identifying and specifying processes requires a switch from a purely function-driven to an actor- or a system-driven perspective. In particular, if stakeholders should be able to design their organization in a self-organized, however informed way, they need to build up modeling capacities with respect to participatory organization development [4].

In S-BPM, both, the intuitiveness of the notation, and the coherence of the approach to modeling and processing, reduces the semantic distance between workflow processing and human understanding of business processes. It should require minimal cognitive effort for all stakeholders to recognize and communicate subsequently modeled business processes. Once stakeholders are able to describe and (re)present their organization of work in such a self-contained way, they may share

C. Stary (Ed.): S-BPM ONE 2012, LNBIP 104, pp. 218–240, 2012.

and communicate organizational designs - a prerequisite to participatory and dynamic development of organizations [1] [4] [5].

When describing business processes stakeholders may either start from scratch or (re-)use existing process models. We will detail both ways, as each of the approaches has certain implications for follow-up activities in the open S-BPM lifecycle. In any case, S-BPM allows stakeholders utilizing natural language structures the same way they compile standard sentences. They should be able to map knowledge about business processes (expressed in terms of standard sentence semantics) to S-BPM diagrams with minimal additional cognitive effort.

All diagrammatic S-BPM representations focus on the interaction among actors or systems, denoted as subjects. The approach allows refining interaction patterns of subjects in terms of exchanging messages, both on a general level of description, and on the level of executable specifications. The latter enables hands-on experience of validated S-BPM models when using a corresponding S-BPM suite, such as Metasonic (www.metasonic.de)

In section 2 we motivate the role of natural language and stakeholder-oriented communication structures. We also describe fundamental concepts to create S-BPM models. Section 3 introduces S-BPM modeling by construction. Hereby, modelers start from scratch identifying involved stakeholders (subjects) and their behavior in terms of sending and receiving messages (i.e. describing mail connections), in addition to individually performed activities (functions). Starting from scratch is a common approach to modeling in BPM - cf. [7].

In section 4 we revisit modeling by restriction, starting out with predefined general interaction patterns between subjects (i.e. actors and/or systems). After giving concrete names to each of the involved subjects modelers need only to keep those communication links that are required for effective task accomplishment. All others are removed from a diagrammatic S-BPM scheme [8].

Modeling by restriction is likely to reduce development efforts, as it starts with a generic, while automatically executable behavior pattern. Hence, the reduction of interaction relations does not hinder the seamless execution of S-BPM models [8].

In section 5 we explain the various action bundles in S-BPM, i.e. the open S-BPM lifecycle, and the role subject-oriented models play in each of the bundles. Section 6 concludes the primer summarizing the main elements for S-BPM capacity building.

## 2 Approaches to S-BPM Modeling

This section motivates and details the use of essential natural language constructs for the representation of business processes in S-BPM. Modelers may either start modeling from scratch through step-by-step construction, or from generic interaction patterns by restricting interactions according the organization of work refining and instantiating a general network of actors or systems.

S-BPM originates from the observation that humans, when structuring and describing their observed reality, use subjects, predicates, and objects. Each of them can be mapped to natural language entities. They support human communication effectively, both in written and oral form. In addition, humans use natural language structures as primary means to ensure mutual understanding [6]. In S-BPM we make use of it, as it facilitates understanding business process models, and sharing of these models.

The S-BPM modeling language captures the above mentioned constituent elements of natural language sentences. Models describe structural properties and behavioral alternatives, including the interaction occurring in the technical and/or organizational environment. S-BPM models can be transformed step by step into an executable application in a seamless way. In order to ensure coherence of specifications, the exchange of messages determines the flow of control. As such, S-BPM enriches flow concepts of function-driven BPM approaches by actors sending and receiving mails.

Modeling means to represent parts of the observed reality in terms of languages. In case of S-BPM natural language terms are used, as they allow for universal use and are familiar to stakeholders through daily communication. S-BPM uses the standard semantics for sentences, comprising subject, predicate and object:

- A *subject* is the starting point for describing a situation or a sequence of events.
- Activities are denoted by *predicates.*
- An *object* is the target of an activity.

Existing modeling approaches tend to focus on predicates or objects, finally adding the subject for natural language explanations of the represented information (cf. identifying function trees before specifying eEPCs in ARIS [7]). In contrast to these approaches S-BPM modeling is a 'subject-first' approach. In further steps the functional behavior of subjects, including task-relevant communication and object (data) exchange, is described. For a more detailed discussion of S-BPM in the context of traditional approaches see [3, pp. 315 ff.].

Models address both, individual work tasks, and organization-wide ones. In the course of accomplishing their tasks, stakeholders receive work inputs, process them, and pass on results. Hence, interaction and communication, either direct or indirect, are to be considered as an essential activity of stakeholders or (IT-)systems for subject-oriented modeling.

Holiday application procedure:

An employee fills in a holiday application form. He/She puts in a start and end date of his/her planned vacations. The responsible manager checks the application and informs the employee about his/her decision; the holiday request might be rejected or get approved. In case of approval the holiday data are sent to the human resource department (HR) which updates the days-off in the holiday file.

**Fig. 1.** Natural language description of an application process for holidays

We exemplify S-BPM modeling using a common scenario in organizations. Employees have to apply for going on holidays or taking some days off. Figure 1 shows the natural language description of the respective process.

This simple holiday process can be modeled following two different approaches (see for details section 3 and 4, respectively). They differ in the starting point specifying a process. The traditional approach (modeling through construction) starts from scratch (‘empty sheet’-approach), whereby the process model is constructed step by step, subject-by-subject. Task-relevant actors or systems need to be identified as the process specification evolves, and the lines of interaction need to be included as required for each subject's task accomplishment. In each step, the process description becomes more complete with respect to work results.

The other approach - modeling through restriction - is only available in S-BPM. It starts with a generic process model which is restricted step by step. When beginning to model, it is assumed that all involved actors or systems might interact mutually. Hence, they can be predefined in a networked structure. In the course of modeling the lines of interaction (i.e. communication relationships) need to be adapted to those required for task accomplishment.

Figure 2 shows both approaches while sketching the conceptual difference between the restrictive and the constructive approach to modeling.

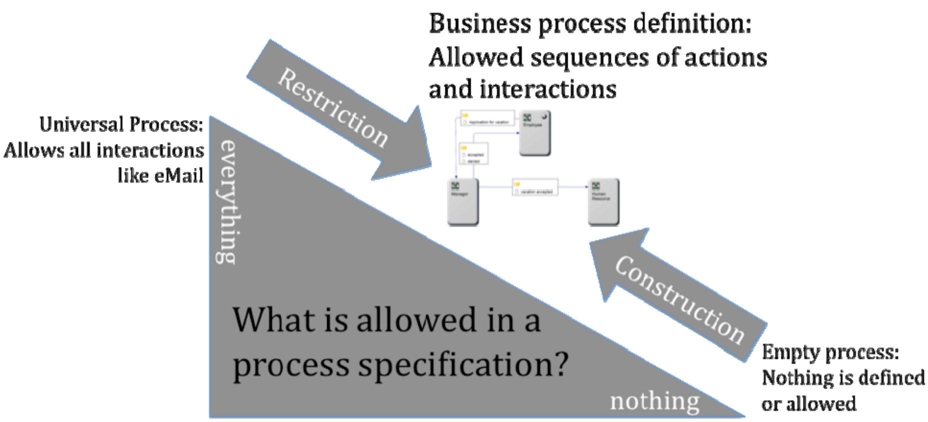



**Fig. 2.** Approaches to model business processes in S-BPM

In the following both approaches will be explained in detail. In section 3 the stepwise creation of a process model is detailed. In section 4 the stepwise reduction of interactions between actors or systems is explained. In both cases actual or envisioned work processes can be represented in a transparent and traceable way. In case validation effort for directly generating workflows from S-BPM models should be minimized, modeling by restriction provides some benefits.

## 3 Modeling through Construction

In this section, the 'empty-sheet' approach to identify and specify business processes in S-BPM is explained. After giving the steps to follow we introduce the S-BPM subject and interaction representations.

### 3.1 Procedure to Follow

Subject-oriented modeling of processes applying the construction approach comprises the following major activities:

- identification and description of the subjects involved in the process, as they occur in the course of accomplishing tasks,
- identification and description of interactions the subjects are part of,
- specifying the messages they send or receive through each interaction, and
- detailing the behavior of each subject.

### 3.2 Subjects and Their Interactions

As subjects are abstract resources representing the parties involved in a process the modeling process starts with identifying the involved subjects and their interactions on a high level of description. It is continued by defining the behavior specifications of each acting party. For completion all exchanges of messages required for achieving work results have to be specified. Each subject is directly addressed, as perceived in reality.

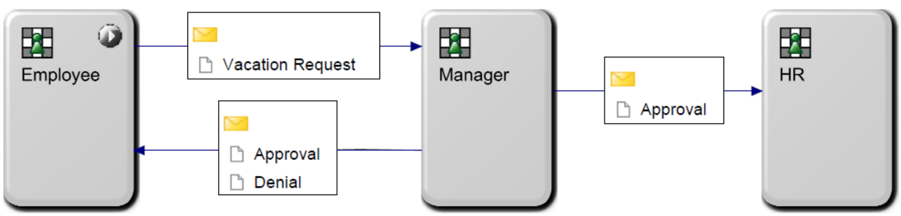



**Fig. 3.** Identified subjects and their communication - the Subject Interaction Diagram

Figure 3 exemplifies the identified subjects and the messages they exchange for the holiday application procedure explained in Figure 1. The modeler has identified the following subjects:

- Employee
- Manager
- Human Resource Department (HR)

The messages they need to exchange according to the scenario description in Figure 1 are:

- Vacation Request from Employee to Manager
- Approval or Denial from Manager to Employee
- Approval from Manager to Human Resource Department (HR)

The resulting diagram is termed Subject Interaction Diagram as it contains all the subjects involved and the interaction relations they need to have for accomplishing a certain task.

### 3.3 Subject Behavior, States and State Transitions

The behaviour of subjects is described by three states (send, receive, internal function) and transitions between these states. These states represent predicates (operations), which means, that they are active elements of the subject description. Services are being used to implement the states. State transitions are necessary to exchange and manipulate business objects.

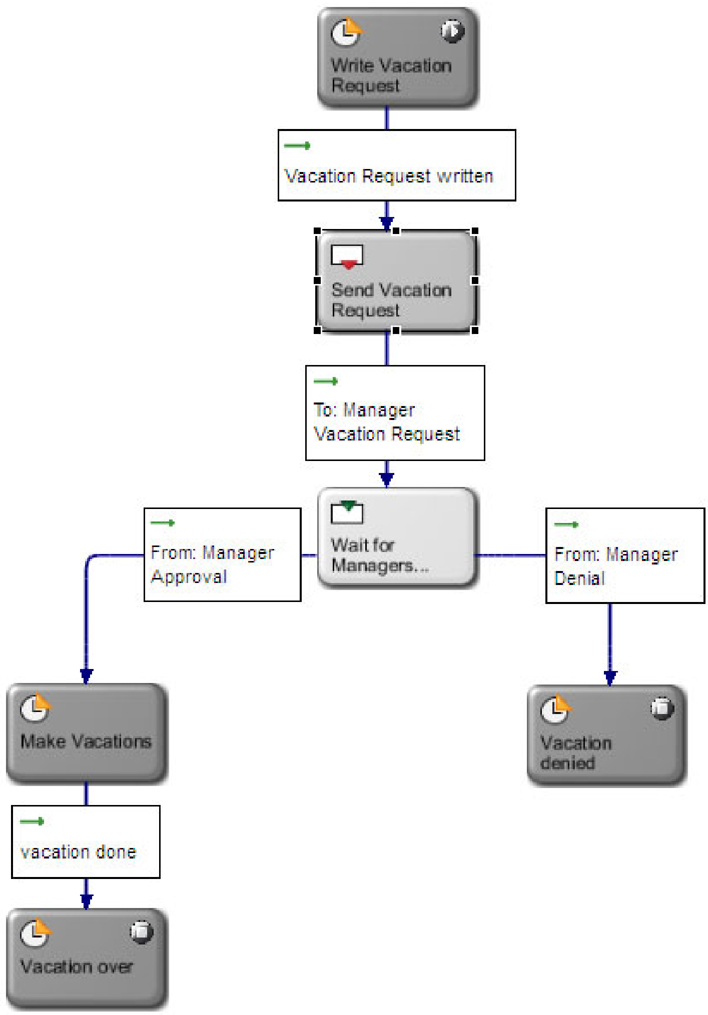



**Fig. 4.** The S-BPM Subject Behavior Diagram for Employee

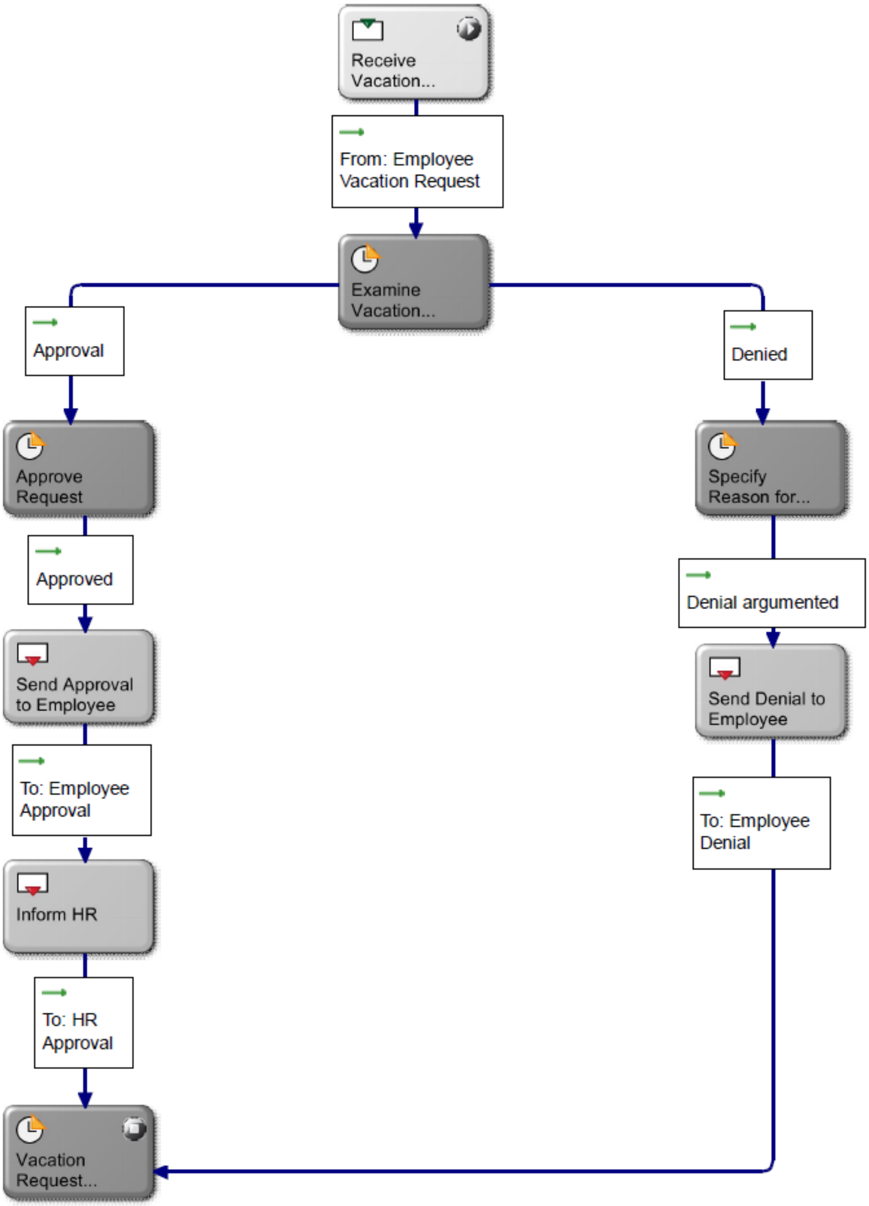



**Fig. 5.** Subject Behavior Diagram for subject Manager in holiday application process

When specifying the behavior of each subject, as shown in Figure 4 for the employee, a sequence of sending and receiving messages, and activities to be set for task accomplishment need to be represented. The initial state on top is marked by a 'Play' symbol in the upper right corner. In this state the employee fills in a holiday

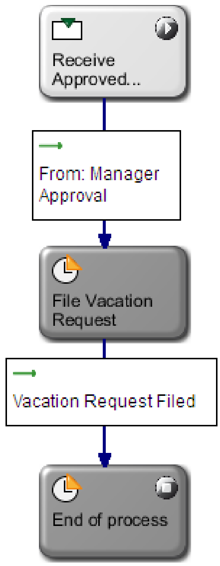



**Fig. 6.** Subject Behavior Diagram for HR department behavior in holiday application process

application form. Upon completion the employee's state switches to the next state via the transition 'Vacation Request written'. This state is a sending state. In this state the holiday application is sent to the manager.

After successful sending the employee reaches the state 'Wait for manager's answer' waiting for approval or denial. This state is a receiving state. In case of denial the process terminates ('Stop' symbol in the upper right corner). In case of approval, the holidays can be taken as applied for. Upon return of the employee the holiday application process terminates, too.

In Figure 4 the employee behavior in the holiday application process is encoded as follows:

- dark grey = function state, e.g., Write Vacation Request
- medium grey = send state, e.g., Send Vacation Request
- light grey = receive state, e.g., Wait for Manager's Answer
- white = state transition, e.g., Vacation Request Written
- start state = on top, i.e. Write Vacation Request
- end state = in the end of sequence of states and transitions, e.g., Vacation Over

The behavior of the manager is complementary to the employee's. The messages sent by employee are received by the manager and vice versa.

Figure 5 shows the behavior of the manager. The manager is on hold for the holiday application of the employee. Upon receipt of the Vacation Request the holiday application is examined (function state). This check can either result in an approval or a denial, leading to either state, informing the employee, and HR (only in case of approval).

In case the holiday application is approved, the HR department is informed about the successful application, and for the subject Manager the process comes to an end.

Finally, the behavior of the HR department has to be detailed (see Figure 6). HR receives the approved holiday application and puts it to the employee's days-off record, without further activities (process completion).

### 3.4 Services

The description of a subject defines the sequence of sending and receiving messages, or the processing of internal functions, respectively. In this way, a subject specification contains a task-relevant sequence of predicates. Predicates can be of the type 'send', 'receive' or 'internal function', the latter dealing with specific objects, such as required when an employee files a holiday application form.

As a consequence at least one operation needs to be assigned to each state. Further detailing of operations is not necessary at the modeling stage of S-BPM (see also lifecycle in section 5), as operations might be processed by existing applications. For instance, filling in a Vacation Request could be supported by a transaction of an ERP (Enterprise Resource Planning) system. A corresponding form based on the structure of an employee data record could be processed for application purposes.

Figure 7 shows how the predicates of a subject are defined by means of objects. They encapsulate all relevant data manipulations based on the Subject Behavior Diagram. Hence, the business object Vacation Request Form for the Holiday Application case contains the following operations:

- Examine application
- Approve Request
- Specify Reason for Denial
- Vacation Finished - Inform HR

As we abstract from implementation details, it is suitable to replace the term operation by the more general term service. A service is assigned to a state and thus, is triggered and processed once the state is reached. The name of the states and the names of the assigned services can be different, as shown in Figure 7. Such differences indicate that in a state several services can be used, in order to define the required functionality executed in a state.

The end conditions correspond to links leaving the state. Each result link of a sending state is assigned to a named service. Before sending this service is triggered to identify the content or parameters of a message. The service determines the values of the message parameters transferred by the message.

Analogously, each output link of a receiving state is assigned to a named service. When accepting a message in this state that service is triggered to identify the parameters of the received message. The service determines the values of the parameters transferred by the message and provides them for further processing.

All services are triggered in a synchronous way, i.e., a subject only reaches its subsequent state once all services called in a certain state have been completed.

### 3.5 Business Objects

For a more complete understanding of the process the content transferred by the send and receive services, and the data processed by internal function services need to be refined. Using the construct 'business object' modelers can define data structures describing the content of messages. They capture process-relevant data elements (with attributes like type, value domain, default values) and nested data structures [3].

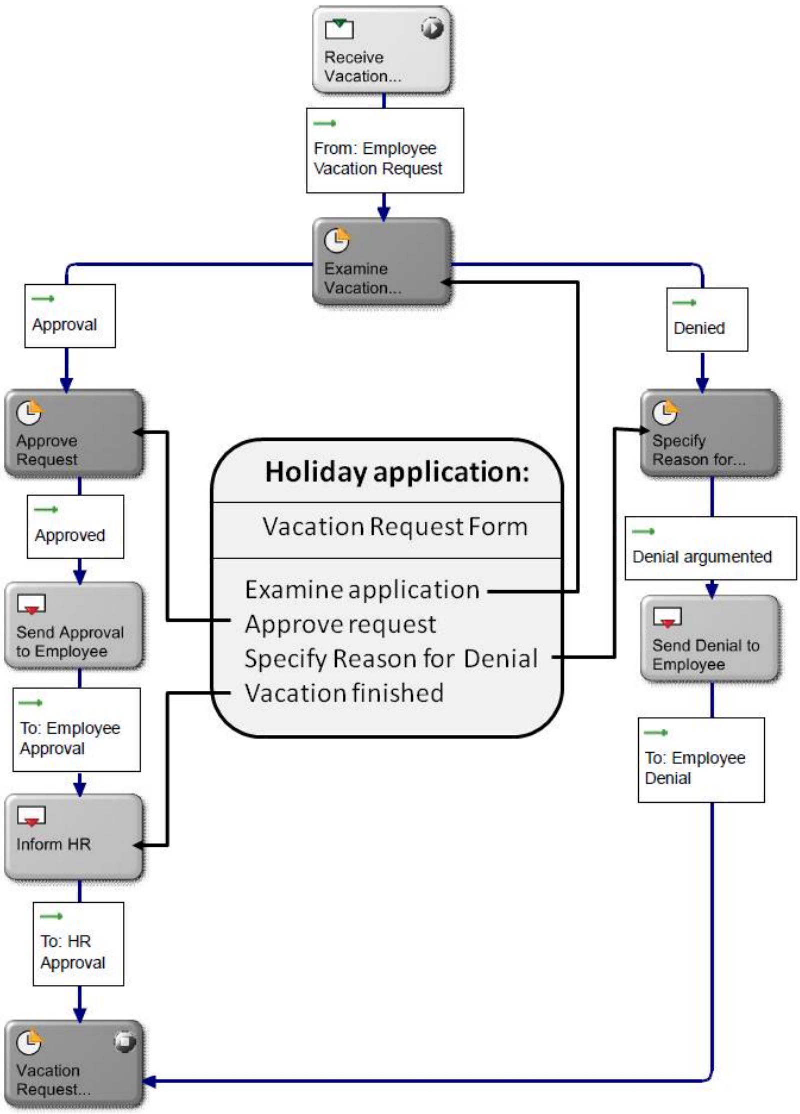



**Fig. 7.** Subject with predicates and corresponding object

Modeling different states of business objects allows physically adding or deleting data elements during runtime. This facilitates process design, e.g., when there is the need restricting the flow of data to business partners across enterprise borders. In addition, defining views on business objects allows limiting the access of subjects to data that exists physically in object instances.

For our example the business object Vacation Request Form is relevant to run the process. In the state Employee it contains data such as internal number, name of the employee, number of days off already taken and remaining, and start and end date of the vacation period requested. Some of these pieces of information are filled in manually by the applicant, others automatically by a service connected to the HR database. In the state Manager a data element capturing absent days due to sickness is available additionally. It could be filled in by a service, and displayed once the manager receives the request for decision making.

# 4 Modeling through Restriction

In this section, the generic networked approach to specify business processes in S-BPM is explained. After giving the steps to follow we introduce the generic network of interlinked subjects and exemplify its stepwise adaptation for the holiday application case. Finally, we sketch some practical benefits when following this approach in BPM.

## 4.1 Procedure to Follow

As mentioned in section 2 the restriction approach in S-BPM starts with an overall generic process model. The procedure requires several restriction steps:

1. Specify a generic template according to the number of parties involved in handling a certain business case (cf. Figure 8)
2. Name the subjects according to the application domain
3. Identify and remove message connections between subjects which are not necessary
4. Name messages and introduce message types according to the application (domain)
5. Adapt specification to actual subject behavior
6. Refine the structure of the business objects transmitted by the various messages

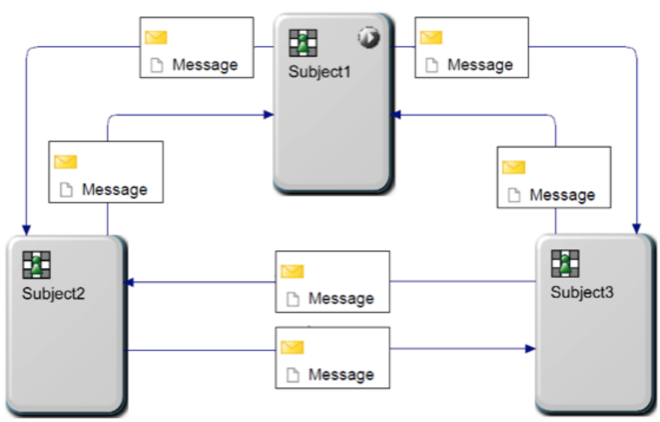



**Fig. 8.** Subject-oriented representation scheme for a 3-party process

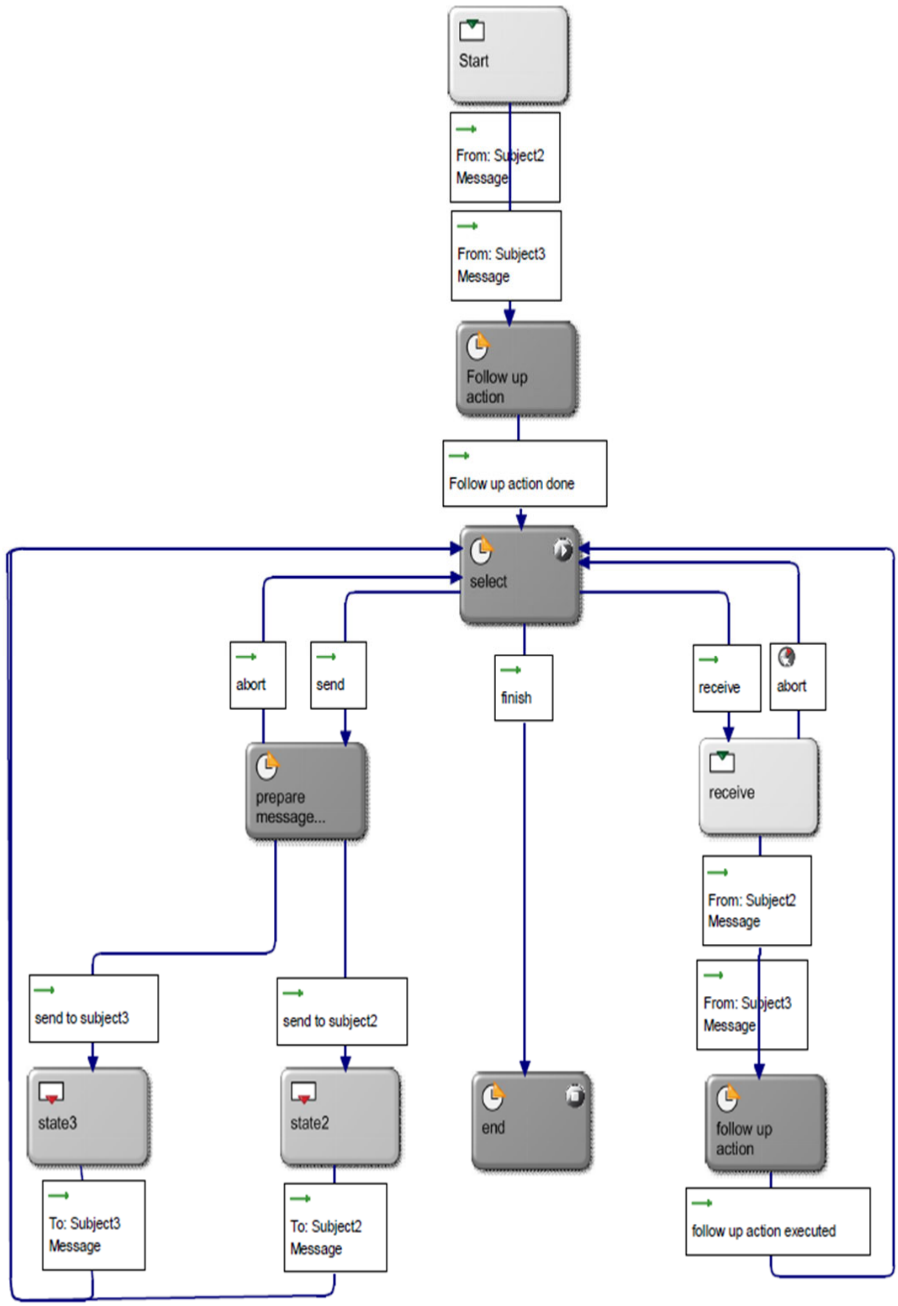



**Fig. 9.** Generic behavior of the start subject Subject1

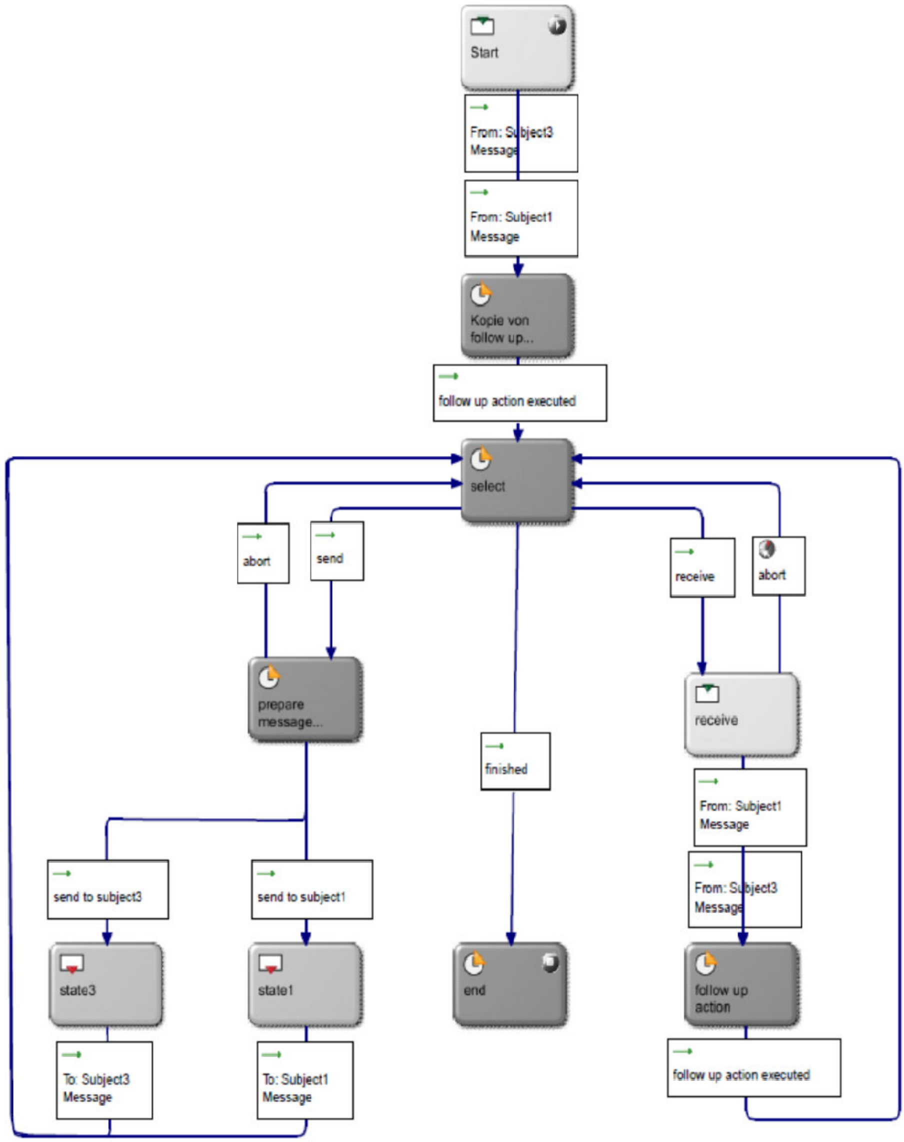



**Fig. 10.** Generic behavior of Subject2

### 4.2 Set Up of Generic Specification Scheme

In the first step a generic template according to the number of parties involved in handling a certain business case needs to be specified. The following figure shows a generic subject-oriented specification scheme with 3 involved parties. It fits to the holiday application process, as the 3 subjects required for are, according to the scenario described in Figure 1: employee, HR department, and manager. In principle, each of the parties can exchange messages with another party. We want to show how this generic process specification can be restricted step by step, in order to achieve a process specification representing the holiday process described in Figure 1.

Each subject starting message exchange is marked with a ‘Play’ symbol (small white triangle) in the upper right corner, as in Figure 8 Subject1.

In principle, each subject can send messages with the name ‘Message’ to any other subject any time. Figure 9 shows the behavior of the subject with the name ‘Subject1’. Since Subject1 is the subject which starts a process its start state is the state ‘select’. The start state is marked with a ‘Play’ symbol. The state ‘start’ and the transitions to the state ‘select’ will be never executed in the start subject.

In the behavior specifications of all other subjects the 'start' state is a 'receive' state because they are all waiting for a message of any other subject (see Figure 10).

In this way all subjects that are not start subjects have to receive at least one message before they can start to send messages. The start subject sends a message to any other subject. The receiving subject can now reach the state ‘select’. In that state any subject can decide upon its next action without restriction. A subject which is in state ‘select’ can send a message to other subjects which are still in the state ‘start’. Now these subjects can also reach the select state and can send messages. Finally, all subjects are in the state ‘select’ and can communicate when addressed.

In the 'select' state the start subject decides whether it wants to send or to receive a message. To start a workflow it does not make sense to receive a message because all the other subjects are waiting for messages (as mentioned before their start state is a 'receive' state). Consequently, the start subject will start with sending messages and the exchange of messages can begin. Choosing the 'send' transition the subject moves to the state ‘prepare message and select address’ and fills out the business object (i.e., the data to be manipulated in the course of task accomplishment). That business object is transmitted by the message ‘message’. After that the subject decides to which other subject the message with the business object as content will be sent.

In the 'select' state a subject can also decide whether it wants to receive a message - this choice can make sense for a start subject further on when moving into the 'select' state the second time.

If there is a message for the subject available it can be accepted and a follow up action can be executed. It is not specified what the follow up action is. This is like receiving an e-mail. The receiver can interpret the content of an e-mail and knows what the corresponding follow up action is. The abort transitions back to the select state enable to step back in case a subject has made the wrong choice.

The representation scheme can easily be set up for any number of participants, following the same principles as shown for 3 parties. The behavior of each subject has to be adapted to the number of subjects in a process. In the send area transitions must be added to send a message to every single new subject, and the same is necessary for the receive area. With that extension scheme the behavior for each type of multi-party process can be generated automatically.

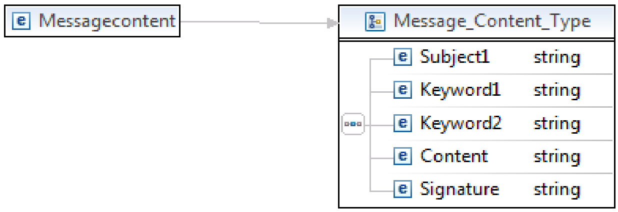



**Fig. 11.** Generic structure of the 'Message' Business Object

With the message 'Message' a business object is sent. The structure of this business object corresponds to the structure of a traditional e-mail with extensions like subject (attention: Here the word 'subject' has a different meaning. It can mean topic, issue, theme etc.), keywords and signature. Figure 11 shows the specification of the business object 'Message' in XSD notation (XML Schema Definition).

### 4.3 Adaption of Generic Scheme to a Specific Application Domain

Following the modeling steps in section 4.1 a process specification is developed corresponding to the involved parties in a business process and their generic interaction structure. In our example the restriction steps (omitting all interactions not relevant for holiday applications) result in a communication structure shown in Figure 12, and a behavior specification of the subject 'Employee' shown in Figure 13.

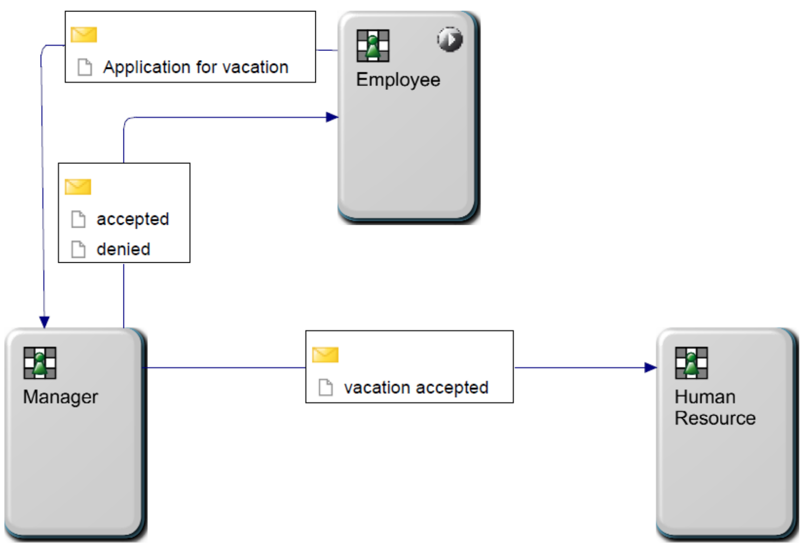



**Fig. 12.** Subjects and exchanged messages after restricting the interaction

With each step in restricting communication, a work task for subject holders becomes more transparent with respect to required inputs for task completion and results. In this way, S-BPM guides organizational developers starting with a network of mutually communicating stakeholders and achieving a focused, since actually required communication scheme for accomplishing tasks.

Comparing Figure 13 with Figure 4 shows that modelling through restriction does not necessarily result in models identical to those created by modeling through construction. Nevertheless both models must deliver the requested results.

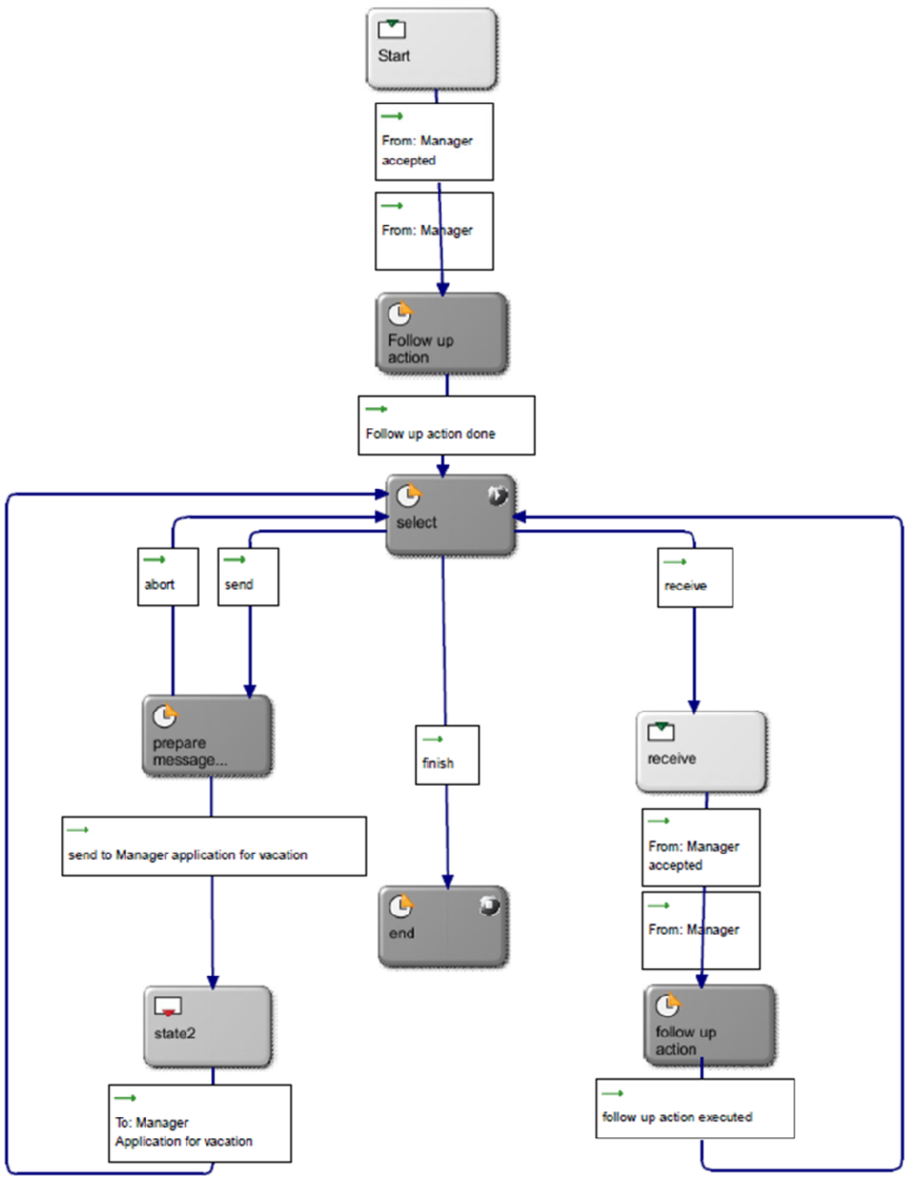



**Fig. 13.** Concrete behavior of subject ‘Employee’

### 4.4 Some Practical Benefits of Modeling through Restriction

The benefits of modeling through restriction can be best explained when comparing it with modeling through construction. As the latter corresponds to a blank paper approach, modeling by restriction is the complete opposite. Modeling by construction starts from scratch, as subjects and their behavior are modeled, once they are considered to be relevant by individual modelers or stakeholders. Subjects and communication relationships are specified in a cumulative way. Communications patterns are defined and explored as the respective modelers or stakeholders perceive work procedures. Each model develops over time and represents the current state of business affairs at a certain point in time. It is not linked to a baseline, such as the generic frame of reference for modeling through restriction, in order to minimize redundancy or provide a certain structure for design cycles. Consequently, revisiting S-BPM models might cause additional modeling workload.

When modeling through restriction modelers utilize a structural frame of reference, as the various possibilities for subjects to interact with each other can be predefined, once the number of involved parties in a business process has been identified. Such a baseline aims to minimize the modeling workload, due to the completeness of the communication pattern set up in the beginning of the modeling process - all possible interactions (i.e., message relationships) between subjects are represented (cf. Figure 8). The generic message relationships serve as placeholders which are removed in case they are not required for completing the process at hand (i.e., as soon as they cannot be named according to their task-specific purpose).

In case of revisiting S-BPM models the generic frame of reference can be revoked to facilitate checking the completeness of an S-BPM model. Candidates for further modeling (either for removal or concrete naming) can be identified easily, as those elements still carry generic labels, such as Message.

Finally, modeling by restriction facilitates the automated execution of S-BPM models. The generalized peer-to-peer network (frame of reference) contains all the subjects that are relevant for a business operation at hand. Since it also contains the possible communication relationships between the subjects, this model represents an S-BPM Interaction Diagram (cf. Figure 8). It contains a complete control flow description for generating workflows. Using a corresponding interpreter (cf. [3]) S-BPM models can be executed on demand - business processes can be experienced interactively, even when some subjects and messages have not been assigned to concrete actors, systems and message paths.

Overall, modeling by restriction is likely to reduce S-BPM efforts, as it starts with a generic, while automatically executable behavior pattern. Neither the reduction of message paths nor level of abstraction hinders the immediate execution of S-BPM models when modeling through restriction. Such models can rather be embedded without further transformations in action bundles of the open S-BPM lifecycle dealing with workflow generation and monitoring (see section 5).

## 5 Models and the Open S-BPM Lifecycle

S-BPM allows organizations to be developed with respect to business functionality and technical process support (cf. [2]). Business-specific aspects of BPM concern relevant management activities such as documentation, design, implementation, control and further development of management processes. They need to be organized in a way that organizations can become fully aligned to the behavior of its stakeholders and its IT support capabilities.

The S-BPM lifecycle provides an open structure for organizing S-BPM activities, as explained in the following subsections. First we detail the different roles required to trigger, guide, implement, and reflect S-BPM projects. Then, we explain the various bundles of activities that can, but need not to be performed in a linear sequence when recognizing the peculiarities of the S-BPM approach. The latter determine the central role of S-BPM models for dynamic organizational development driven by stakeholders, as discussed in the final subsection.

### 5.1 S-BPM Roles

(S-)BPM activities are driven and performed by persons acting in certain roles. Although each of them needs to be considered in the context of the bundle of activities of the open S-BPM lifecycle (see section 5.2), they can be characterized in general as follows:

- **Governors.** They take care about the constraints under which BPM activities are performed. They focus on influential factors that are relevant for change processes, such as market forces or structural particularities of the organization at hand. The governor's tasks range from strategic to operational development. As such, governors address all BPM-relevant stakeholders with respect to organizational development issues. However, they are not responsible for content-wise development and the domain-specific procedures that drive the value chain of a business and are executed by the actors.
- **Actors.** They execute business procedures. Hereby, they manipulate business objects and interact mutually, in order to deliver products and services. They are supported by experts and facilitators with respect to S-BPM activities.
- **Experts.** They are IT-architects, organizational developers, or specific domain specialists. They become part of S-BPM activities once their expertise is required. Typical examples are data engineers, embodying business objects into data management facilities of an organization.
- **Facilitators.** They guide the development process. They handle inputs to (S-)BPM, while ensuring the social acceptability of change proposals on the organizational level. In their interventions they tackle the method and social dimension of organizational change processes. They serve as moderators, and, if required, as mediators.

Governors, actors, experts, and facilitators are involved in each of the bundles of activities (see section 5.2). They are required to analyze business constraints when operating processes, to model and bring models to live, and to learn from and through changes. Their mutual cooperation is particularly essential when the relationships between strategic, tactic, and operative processes of an organization are explored. In general, governors and actors ensure the context-sensitive processing of information, supported by facilitators. They also try to capture complex situations, e.g., looking why performance parameters cannot be met, and consult (domain) experts when required.

### 5.2 S-BPM Activity Bundles

In S-BPM several bundles of activities can be identified (cf. [3]), in line with existing BPM approaches (cf. [2]). They are organized according to phases that need to be performed when organizations shift to process-centered value chains or want to implement business processes. The activities range from analysis to running and monitoring.

**Analysis.** S-BPM models traditionally play a crucial role for analysis, as they are recognized as valid start documentation of BPM projects. They can be created in the course of analysis, if not already existing models are revisited at this stage of development. In any case, the situation of an organization 'as it is' needs to be documented as a result of the analysis process, otherwise the origin or trigger of organizational development gets lost, and changes cannot be checked against a well defined start state of a change or development process. Facilitators, governors, and actors need to collaborate to produce sound analyses. They might be supported by method specialists or domain experts to develop deep understanding of the underlying processes.

**Modeling.** Traditionally, modeling needs to be considered as a separate bundle of activities in BPM, since envisioned processes are specified at this stage of organizational development or change management. Of particular interest in modeling is reducing complex relationships when operating processes, while keeping the specification of business processes coherent. S-BPM allows achieving both:

- *Complexity is reduced* by focusing on subject behavior: Internal behavior is encapsulated in subject-specific models. S-BPM modeling also leads to task- and business-relevant interactions between subjects - the adjusted behavior specifications represent the overall organization of work. In this way, only domain-relevant subjects are considered in the course of modeling. Their structure (representing the organization of work) and internal behavior are strictly separated, while being intertwined in a coherent way: For each subject in the S-BPM Interaction Diagram an S-BPM Behavior Diagram needs to be specified. Finally, for each subject an internal behavior is provided. Its interfaces correspond to the message relationships to those subjects the subject needs to interact with in the course of task accomplishment as given in the S-BPM Interaction Diagram.
- *Coherence is ensured* twofold: First, as it is relevant on the organizational level, interaction patterns need to be complete per se - a sender requires a receiver (and vice versa). Consequently, no incomplete interactions to that respect are allowed. Secondly, being relevant for accomplishing tasks and achieving outcome: Each received message triggers functional behavior of a subject, and leads to further interaction delivering work results (up to the customer), as represented in S-BPM behavior and interaction diagrams.

Apparently, S-BPM models are the core element in the modeling phase. In S-BPM the organizational and subject-specific level, and their interfaces are addressed in a consistent way. An organization is represented in terms of interacting subjects specified in the S-BPM Interaction Diagram. Outcome is generated through the exchange of business objects that are processed by functions. Functions are performed by the involved subjects, and are specified in the S-BPM Behavior Diagram.

In this way, S-BPM captures all essential aspects of BPM, namely the Who, the What, the How, and the When. However, it is the communication-oriented way of specifying organizational and stakeholder behavior ensuring coherence and reducing complexity in change management.

Stakeholders are encouraged by governors and facilitators to participate as active modelers throughout modeling. They might be supported through domain experts or (S-)BPM specialists.

**Validation.** The validation challenges the effectiveness of process models. It is checked whether a process allows producing expected results. As such, a reality check is performed to ensure the expected process performance through models. In this bundle of activities mostly the actors and method specialists are involved, as they need to validate communication and functional procedures with respect to the quality of the results. Facilitators and governors guide them and monitor the process.

**Optimization.** Once a process has been validated, it can be optimized to certain criteria, such as trying to achieve short information paths. In that phase the efficiency of a modeled process is checked. Resource-specific aspects such as time and material consumption are investigated, and might lead to significant changes of models. Again, actors and experts play a leading role in that phase of S-BPM. Here, the approach has the unique benefit that subject-specific behavior, e.g., processing a specific type of material, is primarily encapsulated, but still embodied in its processing context - all interfaces in terms of exchanged information and business objects are represented explicitly.

**Embodying.** Analysis, modeling, and validation occur decoupled from the operational processes going on in an organization. In order to feed the results back to the running business, the process models have to be implemented in the organization. Here, the governors and facilitators take leading roles, as mainly the structure, infrastructure, and strategy of an organization are affected by that task. They might consult actors and method specialists. The implementation is performed on the level of organizing work, and on the level of infrastructure, e.g. IT systems. It may require domain experts to facilitate this process.

**Running and Monitoring.** Once in operation, business processes need to be monitored. In doing so, data are recorded and observations are collected that might trigger further S-BPM analyses and modeling activities. Governors guide this process, eventually consulting actors, experts, or asking facilitators for additional guidance. In S-BPM each actor can track models and trace behavior on the subject and organization layer, as the implemented workflow is a 1:1 mapping of the S-BPM Interaction and Behavior Diagrams to functional software components. Of particular interest at this development stage are meeting expectations, e.g., in terms of removing work hindrances or meeting performance requirements, and recognizing the initialization of further S-BPM activities.

### 5.3 Capturing the Dynamics of Organizational Development

S-BPM is considered a multi-layered change process. Particular bundles of activities or iterations of several bundles enable the emergence of novel organizational behavior, becoming manifest in the various levels of organizational development. Each level corresponds to a certain level of organizational maturity, and can be achieved either in a linear or a non-linear sequence of S-BPM activity bundles, as indicated in Figure 14.

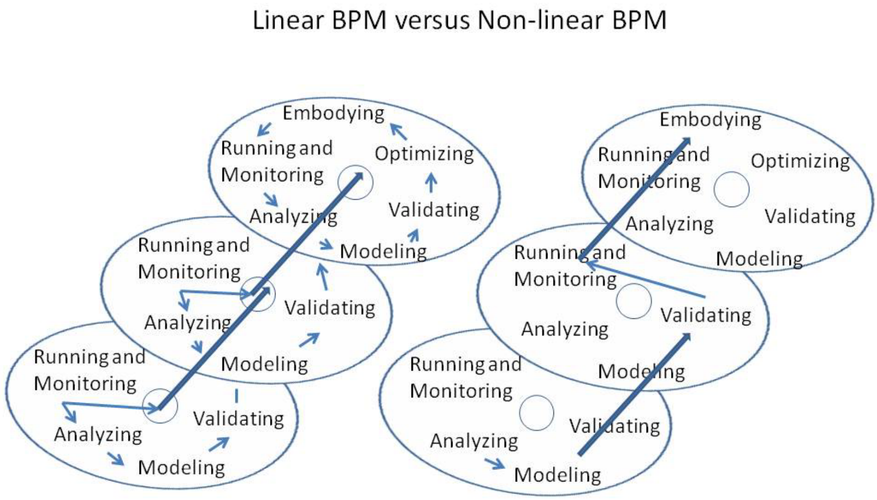



**Fig. 14.** Patterns of organizational development driven by S-BPM

*Linear development* (left part of the figure) corresponds to traditional life cycle approaches to BPM: In order to complete a phase in S-BPM, each activity has to be executed, and needs to be completed at least one time before entering the next life cycle (i.e., the next level in development), even when there are cyclic activities within each life cycle, such as modeling and validating models several times. The transition to the next BPM step is traditionally defined by reaching a dedicated bundle of activities, mainly running and monitoring. It allows observing running a business after modeling and embodying processes into the operation, and before analyzing the effect of implemented process changes. It corresponds to entering already the next BPM cycle, as indicated when following the bold directed link to the upper level in the figure.

In the *non-linear S-BPM approach* (right part of the figure) reaching the next step of organizational development is characterized by being able to switch to a higher stage of development (displayed as upper layer) from each of the activities, as indicated in the figure through the bold directed arcs. The most typical example is changing individual functional behavior while keeping the interaction interface to other subjects. It allows improving the individual organization of work on the fly.

However, its effects become evident on the organizational level through monitoring the concerned subject's behavior in its operative context. Since this emergence of organizational behavior resulting from individual functional behavior modification can be driven by several subjects, the results need to be evaluated (monitoring and analyzing) on another level of organizational development than the one where the changes actually occurred.

The more an execution engine is intertwined with the activities of the life cycle the more direct effects of changes can be experienced and the more likely stakeholder changes lead to the next level of organizational development. It accelerates organizational development.

When handling the S-BPM life cycle in a non-linear way, modeling has to be considered one of the core activities, as models may serve as focal point for improvements or for changes of the communication behavior before becoming effective on the operational level. Due to the coherent decomposition of processes and the resulting behavior management, a step closer to organizational reality can be made.

## 6 Conclusion

Organizations are increasingly forced to restructure their business processes in a flexible way during operation. It requires stakeholders to take responsibility for organizational developments. Traditionally, only few members are skilled in specifying and developing business processes. When using subject-oriented BPM models they can work with natural language constructs (subject, predicate, object) and e-mail-like communication patterns between actors or systems when describing business processes. In this way, individual members of an organization are able to contribute to coherent and intelligible process specifications. In addition, S-BPM models can be processed without transformation, allowing hands-on experience of business processes.

Subject-oriented representation schemes recognize actors or systems as starting point for modeling, regardless of how they evolve. In case they are constructed from scratch they are identified successively, according to the flow of work, still taking into account subjects as part of standard sentence semantics, namely adding operations and business objects in the course of modelling. In case S-BPM models are constructed by restriction, a general communication pattern is successively aligned to the required flow of information and material between the actors or systems necessary to complete a specific process.

Using subjects, stakeholders avoid conveying information reduced either to content or functional business logic. It also ensures coherence, as both, the flow of control, and the addressed data can be kept in their respective context throughout modeling and execution business processes. Consequently, stakeholders and developers should experience less misunderstandings and conflicts in industrial practice. This benefit becomes essential for networked organizations striving for interoperability, as social interaction, cooperation and collaboration aspects have to be reflected by modelling techniques.

## References

1. Acosta, A., Leon, Y.J., Conrad, C.R., Malave, C.O.: Global engineering: Design, decision making and communication. CRC Press, London (2009)
2. Becker, J., Mathas, C., Winkelmann, A.: Geschäftsprozessmanagement. Springer, Berlin (2009)
3. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Geschäftsprozessmanagement. Hanser, München (2011)
4. Heftberger, S., Stary, C.: Partizipatives organisationales Lernen – Ein prozessorientierter Ansatz. DUV, Wiesbaden (2004)
5. Herrmann, T., Hoffmann, M., Kunau, G., Loser, K.U.: A modelling method for the development of groupware applications as socio-technical systems. Behavior & Information Technology 23/2, 119–135 (2004)
6. Pinker, S.: The Stuff of thought: Language as a window into human nature. Allen Lane, London (2007)
7. Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen, 4th edn. Springer, Berlin (2001)
8. Schmidt, W., Fleischmann, A., Gilbert, O.: Subjektorientiertes Geschäftsprozessmanagement. In: HMD – Handbuch der Wirtschaftsinformatik, Heft 266, pp. 52–62 (2009)

# Author Index