

**Markus Nüttgens, Frank J. Rump,
Jan Mendling, Nick Gehrke (Hrsg.)**

EPK 2009

Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten

8. Workshop der Gesellschaft für Informatik e.V. (GI)
und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement
mit Ereignisgesteuerten Prozessketten (WI-EPK)“

26. November - 27. November 2009 in Berlin

Proceedings

Veranstalter

veranstaltet vom GI-Arbeitskreis "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)" der GI-Fachgruppe WI-MobIS (FB-WI) in Kooperation mit der GI-Fachgruppe EMISA (FB-DBIS) und der GI-Fachgruppe Petrinetze (FB-GInf).

Prof. Dr. Markus Nüttgens (Sprecher)
Universität Hamburg
Email: markus.nuettgens@wiso.uni-hamburg.de

Prof. Dr. Frank J. Rump (stellv. Sprecher)
FH Emden/Leer
Email: rump@informatik-emden.de

EPK 2009 / Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Hrsg.:
Markus Nüttgens, Frank J. Rump, Jan Mendling, Nick Gehrke – Berlin 2009

© Gesellschaft für Informatik, Bonn 2009

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Vorwort

Ereignisgesteuerte Prozessketten (EPK) haben sich in der Praxis als Beschreibungsmittel für betriebliche Abläufe etabliert. Mit dem Aufbau der Arbeitsgruppe "Formalisierung und Analyse Ereignisgesteuerter Prozessketten (EPK)" im Jahre 1997 wurde ein erster Schritt unternommen, einen organisatorischen Rahmen für Interessenten und Autoren wesentlicher Forschungsarbeiten zu schaffen und regelmäßige Arbeitstreffen durchzuführen (Organisatoren: Markus Nüttgens, Andreas Oberweis, Frank J. Rump). Im Jahr 2002 wurden die Arbeiten der "informellen" Arbeitsgruppe in den GI-Arbeitskreis "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)" der GI-Fachgruppe WI-MobIS (FB-WI) überführt und inhaltlich erweitert. Der 1. Workshop „EPK 2002“ fand im November 2002 in Trier statt, der 2. Workshop „EPK 2003“ im Oktober 2003 in Bamberg im Vorfeld der 11. Fachtagung „MobIS 2003“, der 3. Workshop „EPK 2004“ in Luxemburg im Rahmen der GI-Fachtagung „EMISA 2004“, der 4. Workshop „EPK 2005“ in Hamburg, der 5. Workshop „EPK 2006“ in Wien, der 6. Workshop „EPK 2007“ in St. Augustin und der 7. Workshop „EPK 2008“ in Saarbrücken statt.

Der Arbeitskreis soll Praktikern und Wissenschaftlern als Forum zur Kontaktaufnahme, zur Diskussion und zum Informationsaustausch dienen. Die Aktivitäten des Arbeitskreises werden unter der Internetadresse <http://www.epk-community.de> dokumentiert (aktuell: 250 Mitglieder).

Der vorliegende Tagungsband enthält zwölf vom Programmkomitee ausgewählte und auf dem Workshop präsentierte Beiträge. Jeder Beitrag wurde innerhalb der Kategorien Fachbeiträge und Diskussionsbeiträge einfach-blind und dreifach begutachtet.

Die Beiträge decken ein breites Spektrum zur Spezifikation und Anwendung Ereignisgesteuerter Prozessketten (EPK) ab:

- EPK Konzepte
- EPKs und Werkzeuge
- EPKs und andere Formalismen
- Qualitätsaspekte von EPKs

Der Tagungsband wurde ausschließlich in digitaler Form publiziert und ist im Internet frei verfügbar (CEUR Workshop Proceedings).

Wir danken der Humboldt-Universität zu Berlin für die Bereitstellung der Räumlichkeiten und den Autorinnen und Autoren und den Mitgliedern des Programmkomitees für die Beiträge zur Realisierung des Workshops.

Hamburg, Emden und Berlin im November 2009

Markus Nüttgens
Frank J. Rump
Jan Mendling
Nick Gehrke

Programmkomitee

Thomas Allweyer, FH Kaiserslautern
Jörg Becker, Uni Münster, ERCIS
Jan vom Brocke, Hochschule Liechtenstein
Jörg Desel, KU Eichstätt
Andreas Gadatsch, FH Bonn-Rhein-Sieg
Christian W. Günther, TU Eindhoven
Frank Hogrebe, Uni Hamburg
Ekkart Kindler, DTU Kopenhagen
Peter Loos, Uni Saarland, IWi im DFKI
Jan Mendling, HU Berlin
Markus Nüttgens, Uni Hamburg (Co-Chair)
Andreas Oberweis, TU Karlsruhe
Michael Rebstock, FH Darmstadt
Peter Rittgen, Uni Boras
Michael Rosemann, Queensland University of Technology
Frank Rump, FH Emden/Leer (Co-Chair)
Carlo Simon, Provadis School of Int. Management and Technology AG, Fft. a. M.
Oliver Thomas, Uni Osnabrück
Mathias Weske, HPI Potsdam

Organisation

Nick Gehrke, Uni Hamburg
Jan Mendling, HU Berlin

Weitere Gutachter

Robin Bergenthum
Sebastian Mauser

Inhaltsverzeichnis

Fachbeiträge

Fabian Friedrich Measuring Semantic Label Quality Using WordNet	7
Constantin Houy, Peter Fettke, Peter Loos Stilisierte Fakten der Ereignisgesteuerten Prozesskette - Anwendung einer Methode zur Theoriebildung in der Wirtschaftsinformatik.....	22
Henrik Leopold, Sergey Smirnov, Jan Mendling On Labeling Quality in Business Process Models	42
Volker Gruhn, Ralf Laue Ein einfaches Verfahren zur Erkennung häufiger Fehler in EPKs.....	58
Nicolas Peters, Matthias Weidlich Using Glossaries to Enhance the Label Quality in Business Process Models.....	75
Gero Decker, Willi Tscheschner, Jörg Puchan Migration von EPK zu BPMN.....	91
 <i>Diskussionsbeiträge</i>	
Ralf Laue, Stefan Kühne, Andreas Gadatsch Evaluating the Effect of Feedback on Syntactic Errors for Novice Modellers	110
Andreas Hoheisel, Thorsten Dollmann, Michael Fellmann Überführung von EPK-Modellen in ausführbare Grid- und Cloud-Prozesse.....	118
Kathrin Figl, Jan Mendling, Mark Strembeck Towards a Usability Assessment of Process Modeling Languages	138
Jens Brüning, Peter Forbrig Modellierung von Entscheidungen und Interpretation von Entscheidungsoperatoren in einem WfMS.....	157
Patrick Delfmann, Sebastian Herwig, Lukasz Lis Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten – Linguistische Analyse und Vorschlag eines Lösungsansatzes	178
Frank Hogrebe, Alexander Jürgens, Sven Pagel, Markus Nüttgens EPK-Varianten auf dem Prüfstand: Explorative Studie zur Gebrauchstauglichkeit von eEPK und oEPK.....	195

Measuring Semantic Label Quality Using WordNet

Fabian Friedrich

School of Business and Economics,
Institute of Information Systems
Spandauer Strasse 1, Berlin, Germany
Fabian.Friedrich@informatik.hu-berlin.de

Abstract: The automatic determination of defects in business process model and the assurance of a high quality standard are crucial to achieve easy to read and understandable models. Recent research has focused its efforts on the analysis of structural properties of business process models. This paper instead wants to focus on the labels and their impact on the understandability and integratability of process models. Metrics which can help in identifying process model labels that could lead to misunderstandings are discussed and a way to automatically detect labels with a high chance of ambiguity is presented. Therefor the lexical database WordNet is used to obtain information about the specificity and possible synonyms of a word. The derived measures were then applied to the SAP Reference Model and the most interesting findings are presented.

1 Introduction

Business Process Modeling has received more and more attention in recent years. Naturally, the interest in the quality of those models grows with their application. Different frameworks have been developed to understand the factors that influence the quality of business process models. A long established framework is Lindland et al. [LSS94] developed in 1994, which served as a basis for many other quality definition approaches [KJ03, KSJ06]. But these Frameworks provide only qualitative statements about process model quality and are rather abstract. So far very little research has been conducted to find appropriate quantitative measurements. Some of them are, for example, the Cross Flow Connectivity or the Density metrics [VCR⁺] which try to analyze the structure of a given process model.

Figure 1 shows an EPC from the SAP Reference Model, which was chosen as a basis for tests for the following analysis. What makes the labels of this model interesting is the fact that the term "wage" and "remuneration" were used within the same model although they are interchangeable. This problem of inconsistent usage of terms arises due to different levels of detail and abstractions used by different modelers [HS06]. Hence, comparing and merging of models or sub-models becomes more complicated because of these conflicts [Pfe08]. To detect and avoid those conflicts this paper will propose solutions by analyzing the labels of process models. The particular approach is to analyze the meaning of these labels, using the well known WordNet semantic database [Mil95] and to define a

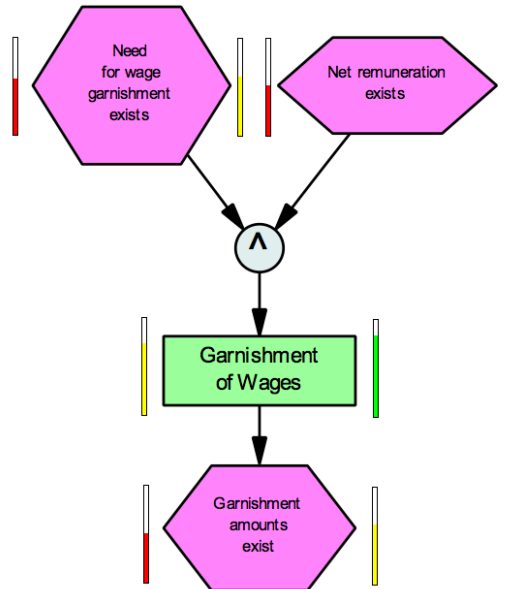


Figure 1: Example of an analyzed EPC. The graphical extension is explained in section 6

quantitative measure which is able to provide clear evidence to whether a label is good or bad.

On the following pages a short introduction of the elements in WordNet is given. Afterwards these elements will be used to derive two quantitative measures for the semantic quality of process model labels. The last chapter will then present the results of the application of those measures to the EPCs of the SAP Reference Model to verify their value. The paper will conclude by critically assessing this application's results and provide an outlook to possible extensions and further research.

2 Background

This section wants to introduce the preliminaries for the metrics that were developed. As the focus of this work are the label's semantics, an overview of the lexical database WordNet will be given. Furthermore, section 4 will make use of semantic relatedness measures to determine the meaning of a word. Therefore, the main principles of semantic relatedness will be shortly introduced, too.

2.1 A Brief Introduction to WordNet

WordNet is a semantic database which was developed in 1985 at Princeton University, mainly for natural language processing [Mil95]. Since then it has steadily grown and today it contains more than 155,000 words. These words are organized into so called SynSets (Synonym Sets). A SynSet contains several words which share the same meaning. Furthermore, the SynSets in WordNet are linked to each other through pointers with different meanings. Thus a program is able to extract different semantic relations for a given word. For example:

- Synonyms - Words which have the same meaning (to work on - to process)
- Homonyms - Words which are written identically, but have a different meaning
- Hypernyms/Troponyms - Nouns which are superordinate to the given noun. The opposite of a hypernym is a Hyponym. The same principle can also be applied to verbs which is called a troponym then. (sue - challenge, tree - plant)
- Meronyms - Structures nouns in a "part-of" relationship. (car - wheel)
- Antonyms - Mainly used for adjectives and adverbs and describes the opposite (wet - dry, hot - cold)

The quality metrics that will be explained in detail in section 3 will make use of the possibility to extract synonyms and hypernyms/troponyms from WordNet.

2.2 Semantic Relatedness

Semantic relatedness is a measure that states in how far two terms are related to each other. Some of the most popular semantic relatedness measures are those of Hirst and St-Onge, Leacock and Chodorow, Resnik, Jiang & Conrath [JC97] and Dekang Lin [Lin98] which were compared in [BH01]. The latter two were used in the conducted experiments. A recent master thesis also investigated on this topic [Scr06]

Both measures leverage the hierarchical structure that is build up within WordNet. But they rely on statistical information on the probability of the occurrence of a word. A possible approach to determine these probability values is to count the number of occurrences within a large corpus of text such as the complete works of Shakespeare, the Brown corpus from the ICAME Collection of English or the British National Corpus¹, which was used for the experiments in this paper. Once the probability values of a concept C_i can be determined, $P(C_i)$, and the first concept C_0 that subsumes both of those concepts has been extracted from the hierarchical structure, the similarity can be computed as follows:

Lin:

$$sim_{Lin}(C_1, C_2) = (2 * \log P(C_0)) / (\log P(C_1) + \log P(C_2)) \quad (1)$$

¹<http://www.natcorp.ox.ac.uk/>

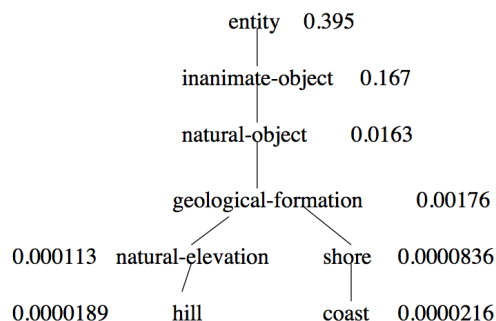


Figure 2: A Fragment of WordNet with its corresponding probability value. [Lin98]

Jiang & Conrath:

$$sim_{JCn}(C_1, C_2) = -\log P(C_0) \quad (2)$$

Taking the example from figure 2 this means that the similarity between the words "coast" and "hill", given that their first common parent is "geological-formation", is 0.59 (Lin) or 9.15 (Jiang & Conrath), respectively. Obviously, the metric defined by Dekang Lin has the advantage that it is always within the bounds of 0.0 and 1.0, but as the word sense disambiguation conducted in section 4 only tries to determine a maximum value it has no influence on the metrics, yet.

3 Quality Metrics

The information that can be gathered through the methods previously explained will now be used to determine the semantic quality for a given label. As this quality largely depends on the environment the label is found in, a model cannot be considered independently, but a collection of many models - a model repository - has to be analyzed. To test the metrics presented in this paper, the 604 EPCs of the SAP Reference Model were used. The focus will be the analysis of nouns and verbs, as especially the specificity metrics which will be described below rely on the hypernym/troponym structure within WordNet.

3.1 Consistency

The first problem that is addressed here is the use of several words with the same meaning, as this contradicts the principle of a shared vocabulary and increases the ambiguity of labels and is responsible for misunderstandings. This occurrence of 2 synonyms referring to the same concept or sense respectively, is a conflict that can lead to problems in the process of consolidating a model repository. Resolving those conflicts usually demands

word	usage count	frequency
order	1025	84%
purchase	199	16%
bill	202	60%
invoice	132	40%

Table 1: Example of synonym words used in the SAP Reference Model

the consultation of several technical and domain experts and can thus become very costly. [DHLS08]

An example of such practice would be the usage of both "invoice" and "bill" within the same model repository.

The first step to detect such rarely used synonyms is to count the occurrence of every word within the model repository. To unify different word forms the stemming algorithm of Porter [Por80] was used. Afterwards synonyms of a given word can be acquired through WordNet. The number of occurrences of the given word divided by the total number of occurrences of all synonyms of the word then determines the relative frequency of the given word within the model repository. Another example are the words "order" and "purchase" which both can be found in the SAP Reference Model (see table 3.1).

Now, the consistency quality can be measured. Either an occurrence of 100% can be demanded for each word compared to its synonyms or a continuous quality value based on a minimum frequency m , which has to be defined externally, can be used. The quality measure for a word x can then be computed using a quality function like this one:

$$q_{cons}(x) = \min(1.0, (\frac{frequency(x)}{m})^2) \quad (3)$$

This quality function has the advantage that it is scaled between 0.0 and 1.0 and as soon as the frequency value x of a words falls below m the quality will rapidly decrease, but a distinction is still possible. Additionally, the quality measure depends on the minimum frequency m . Taking the example above, this means that at a minimum frequency level of 30% the word "purchase" will be assigned a quality value of ~ 0.284 , whereas at $m = 50\%$ it is ~ 0.098 . Hence, it can be defined how strict the quality metric is supposed to evaluate the labels of a given model.

3.2 Specificity

Another problem, that arises when different people are modeling the same domain, is that each can use a different level of abstraction to describe reality. A good indicator for the used level of abstraction is the depth within the WordNet hypernym tree. The deeper the level of a word, the more specific it is. Thus, to align different sub-models within the repository all the words used should be in the same depth range within the WordNet hypernym tree. As this tree is only available for verbs and nouns the analysis

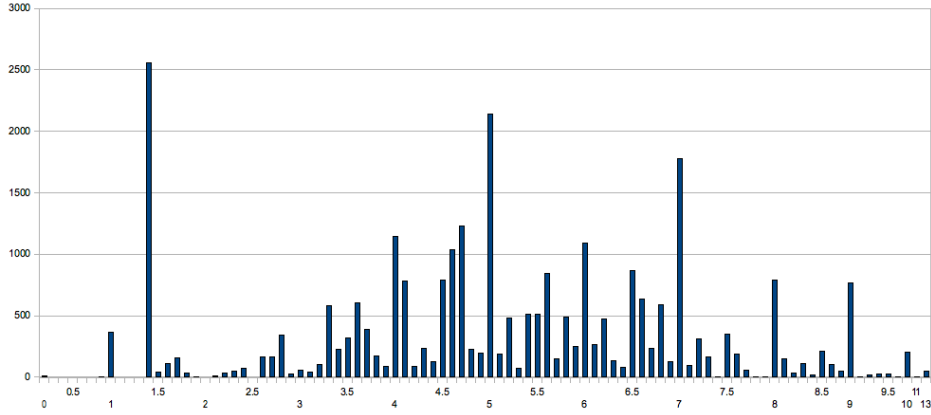


Figure 3: The distribution of average depth values of the words from the SAP Reference Model

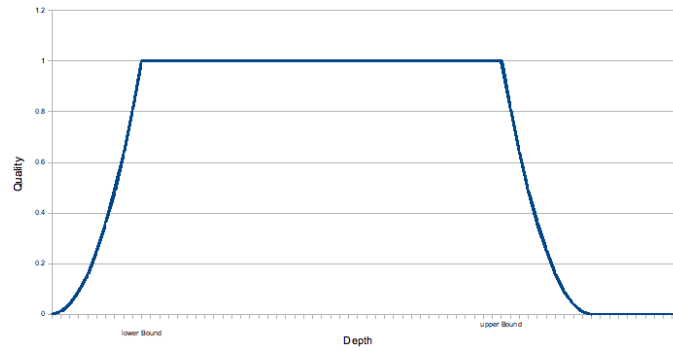


Figure 4: Graphical representation of the q_{spec} quality function

only uses those and leaves the evaluation of adjectives and adverbs for further research. To compute the depth for a given word a first approach was to determine the average depth of all possible senses which are present in WordNet. The resulting distribution is shown in figure 3. It is visible that most of the words are in a depth level of about 5. Similar to the Consistency Metrics presented before, an arbitrary lower and upper bound can now be defined and used in a quality function q_{spec} . As an example this could be:

$$q_{spec}(x) = \begin{cases} 0.0, & \text{if } depth(x) > bound_{lower} + bound_{upper} \\ \min(1.0, (\frac{depth(x)}{bound_{lower}})^2, \frac{depth(x)}{(bound_{upper} + bound_{lower})^2}), & \text{else} \end{cases} \quad (4)$$

$$\min(1.0, (\frac{depth(x)}{bound_{lower}})^2, \frac{depth(x)}{(bound_{upper} + bound_{lower})^2}), \text{ else} \quad (5)$$

as depicted in figure 4. This function provides the same advantages as mentioned for q_{cons} before, but punishes deviations into both directions now. Alternatively, a function which is cut off only on one side could be used if it is assumed that only a superficial labeling style causes problems.

The following section will discuss problems of the methods described so far and how they were further optimized.

4 Consistency and Specificity Quality using Semantic Relatedness

The problem with the approach explained above is that words in WordNet can have many different meanings and the usage of averages can lead to unwanted results. On the one hand this can distort the consistency quality as a synonym might just not be appropriate for the word under investigation. On the other hand it could lead to a depth for the specificity metric that deviates strongly from the one the specific meaning intended for this word has. Thus, it is necessary to determine the meaning of a word within a label prior to calculating its quality.

Determining the semantics of a word is a well known task in natural language processing. Some of the problems and ways to solve them can be found e.g. in [Sus93, BGBZ07, Yar92]. The main idea is to use the context of a word and to evaluate which meaning is the most probable. In our case, the context of a word in a label is the rest of the label, of course, but also the labels of its predecessors and successors. If one of them should be a join or split node (XOR, OR, AND) the predecessors/successors of that node will be taken into account, too. Afterwards, the semantic relatedness as described in section 2.2 is computed and averaged for each of the possible meanings a word has. The meaning with the highest relatedness to its context is selected. This procedure is executed independently for each word. Thus the meaning which was selected for one word does not influence the selection of a meaning of another word, contrary to the approach taken in [Sus93].

Applying this technique makes the results much more accurate as only relevant synonyms are taken into account for the consistency metric. The distribution of depth values also becomes clearer when only concrete meanings are evaluated. See figure 5.

The only problem is that the disambiguation itself cannot be guaranteed to be always correct and a false meaning could get selected. Therefore, the quality aspects of both ways (using averages and using the selected meaning) will be regarded. Furthermore, the influence of both metrics on the quality of a word will be scaled by a factor α which can be externally set. The quality of a single word then becomes:

$$q_{word}(x) = \beta * q_{average}(x) + (1 - \beta) * q_{specific}(x^*) \quad (6)$$

where

$$q_{specific}(x) = q_{average}(x) = \alpha * q_{cons}(x) + (1 - \alpha) * q_{spec}(x) \quad (7)$$

but the specific quality uses the selected meaning (here denoted by x^*).

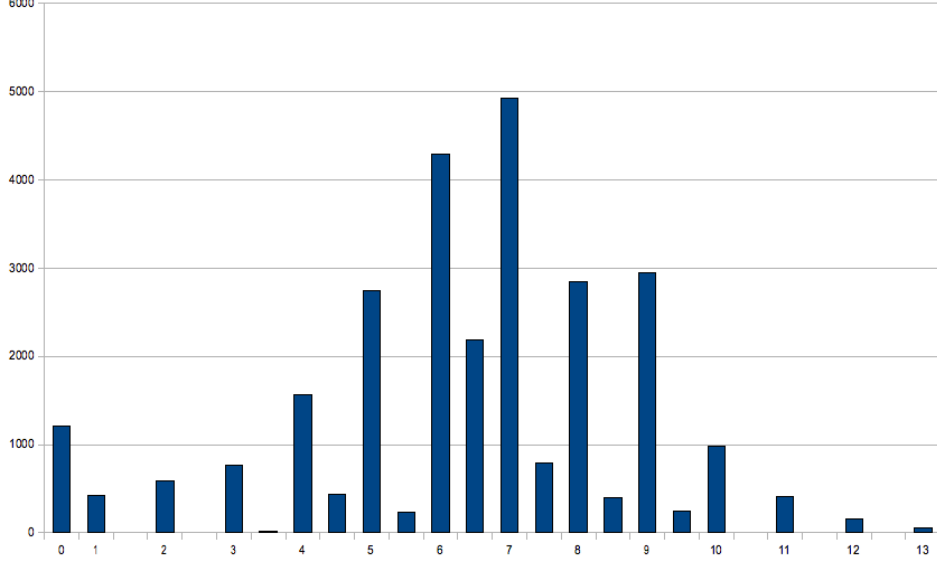


Figure 5: The distribution of depth values after applying the disambiguation technique explained above. Values of .5 exist as a SynSet can have two hypernyms with different levels within the tree (or graph in this case).

5 Aggregation on Label and Model Level

To quickly identify and easily visualize these semantic quality metrics it is necessary to aggregate them for a whole label and/or model. This enables the user to quickly identify models with the least quality and to adjust the labeling in a quality assurance process. Our approach was to calculate the arithmetic mean and variance to determine the quality of a given label/model.

$$q_{label}(Label) = \sum_{i=0}^n \frac{q_{word}(Label(i))}{n} \quad (8)$$

where n is the number of nouns and verbs and $Label(i)$ denotes the i -th noun or verb within the label.

$$\sigma_{label}^2(Label) = \sum_{i=0}^n \frac{(q_{word}(Label(i)) - q_{label}(Label))^2}{n} \quad (9)$$

The same applies for the model level:

$$q_{model}(Model) = \sum_{i=0}^m \frac{q_{label}(Model(i))}{m} \quad (10)$$

where m is the number of labeled elements (in the case of EPCs Functions and Events) and $Model(i)$ refers to the i -th label within the model under investigation.

$$\sigma_{model}^2(Model) = \sum_{i=0}^m \frac{(q_{word}(Model(i)) - q_{label}(Model(i)))^2}{m} \quad (11)$$

On the one hand models with a low quality should be subject to further quality checking, but also models with a high variance are interesting. They will have a single label that is evaluated as bad by our metric, surrounded by many good labels. Examples for both cases will be presented in the next section.

6 Application to the SAP Reference Model

The quality model described before was prototypically implemented and applied to a part of the SAP Reference Model. In particular the 604 EPCs were examined. The parameter α was set to 0.5 so the consistency and specificity metrics were regarded equally important. The degree to which the quality using the specific meaning is used β was set to 0.8. A minimum occurrence of 51% was demanded for the consistency quality q_{cons} and the lower and upper bound for the specificity metric were set to the value of the lowest and highest quantile.

One of the models with the worst total semantic quality was the one shown in figure 1 with a low total semantic quality score of (0.66). The graphical representation was enhanced with two bars which depict the quality using averages (left) and using the specific meaning (right). When the value for the label drops below 0.8 the bar becomes yellow and below 0.6 red. The main issue with this model is the use of the very specific word "garnishment" which lies in the 9th level in the WordNet hypernym tree. On the other hand the very generic verb "exists" is used. Interestingly, even in this small model 2 synonyms "wage" and "remuneration" are used alongside. But while "remuneration" was used 13 times throughout the model repository, wage only appeared twice.

From the information that was acquired, now even recommendations can be given to a user to increase the label quality. Synonym usage can be decreased if "remuneration" is also used in the upper left event and in the function. Furthermore the word "exist" could be replaced with one of its hyponyms. Albeit it is hard to make an automated recommendation as "exist" is very unspecific and lies on the 0-th level within the hypernym tree. Lastly, the word "garnishment" could be replaced by one of its hypernyms e.g. "court order" so the model becomes more abstract and aligns better with the other models in the repository.

Another interesting finding was that a high variance usually arises, when the model contains events which were not labeled at all and were therefore evaluated with a semantic quality of 0.0. But there are also models as the one shown in figure 6 where a single function is responsible for the high variance, although the total quality of the model is not strongly affected. This model is also an example for the problems that arise with our metrics. The quality of the function "RFQ/Quotation" is low because the special abbreviation Request for Quotation (RFQ) is not part of WordNet. A way to solve these kinds of problems would be to create a lexical database similar to WordNet, but specialized for a business

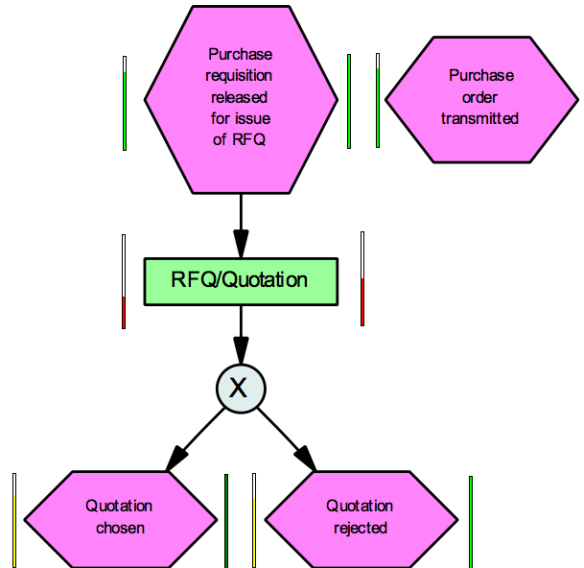


Figure 6: Example of an analyzed EPC with a high variance (0.039) and a semantic quality score in the upper medium range. (0.85)

administration and IT context.

For the following analysis the parameters of the metrics were altered to be a lot stricter with a minimum frequency of 80% and the specificity bounds set to 4.5 and 5.5. The five best and worst labels that resulted from this analysis are depicted in table 2.

By looking into the detailed data for each word within a label it becomes evident why they were evaluated good or bad, although some of the labels seem to be very similar. Table 3 shows in how far the single words of some of the labels shown before do not fulfill the requirements of a consistent labeling style. When interpreting the results it is important to remember that no syntactic features were examined by our quality metric. Thus the label "Distribution" is of high semantic quality regarding its specificity and consistency with the modeling repository, although no clear action can be derived from its syntax. New mechanisms which can be used to tackle those issues are discussed e.g. in [LMS].

The low score of the word "was" could be problematic in the last case as it is used as an auxiliary verb to form the past tense. Although the label could be improved by changing it to "Created Revaluation" the exclusion of some basic vocabulary would prevent such issues.

label	avg. Quality	spec. Quality	total Quality
Material BOM Distribution	0.92	1.00	0.98
Distribution	0.87	1.00	0.97
Electronic Bank Statement	0.98	0.95	0.95
Transaction data	0.95	0.95	0.95
Document Distribution ALE	0.79	0.99	0.95
RFQ/Quotation	0.26	0.06	0.09
Write-up	0.46	0.07	0.15
Post-capitalization	0.38	0.09	0.15
Usage Decision	0.34	0.14	0.18
Revaluation was made	0.20	0.22	0.21

Table 2: The five best/worst labels

word	usage frequency	depth	synonym frequency	possible Hypernym/Troponym
Material	100%	4.5	0% (stuff)	-
BOM	-	-	-	-
Distribution	100%	5.0	no synonyms	-
Transaction	100%	5.5	0% (dealing)	-
data	100%	4.0	0% (information)	raw data; metadata
Usage	51%	6.0	28%(exercise)	activity
Decision	14%	7.0	84%(determination)	choice; selection
Revaluation	100%	8.0	0% (review)	assessment
was	8%	0.0	61%(is) 15%(be)	≥ 100
made	10%	0.0	90%(create)	≥ 50

Table 3: Closer analysis of the labels

7 Related Research

One of the first models used for describing the quality of a process was the one by Lindland et al [LSS94]. As our approach tries to increase the understandability of the model for the audience, it is a part of the pragmatic quality mentioned in this work. Although Lindland also defines the term "semantic quality", meaning the congruence of the model and the domain which has to be modeled, it is not related to what is addressed as semantic label quality in this paper. But, our metrics entirely depend on the WordNet semantic database and try to automatically determine the meaning of a word from its given context. Hence, the term semantic label quality seemed appropriate.

So far quantitative evaluation approaches concentrated on the evaluation of the structure of the process model e.g. [VRM⁺08]. A good overview provides [VCR⁺] and an empirical correlation analysis between understandability and structural features can be found in [MS08]. Some research especially on the impact of labeling and the structure of labels was recently conducted by Jan Mendling [MR08a, MR08b, MRR09]. So far no research that tries to leverage the information of WordNet for the improvement of Business Process Model labels is known to the author.

8 Conclusion

In this paper a quantitative approach for determining the quality of process model labels was presented. The metrics developed utilized the information that is available in the WordNet lexical database, with the aim of minimizing misunderstandings between several stakeholders. To achieve this, ambiguity by the usage of synonyms or words from different levels of specificity is penalized. To test the approach defined here it was applied to the EPCs, of the SAP Reference Model with promising results. Although it has been applied to EPCs the model can be seamlessly transferred onto other modeling notations like BPMN, YAWL or UML Activity Diagrams. A different procedure was developed at the University of Muenster [DHLS08], where the agreement upon a restricted set of words and syntactical constructs has to be found before starting to model. While modeling, these constructs are then enforced by a modeling tool. Although this approach guarantees consistent results, it is quite restrictive for the modeler himself. In contrast to that, our approach ensures that a stringent und unified labeling style emerges through agreement between the different parties involved in the modeling process. Thus, the costly process of defining a set of allowed words prior to modeling can be omitted. Furthermore, recommendations on how to improve the alignment of a model's label with the rest of the repository can be automatically derived from the metrics presented. Therefore, it also addresses the main characteristics which are demanded in [Reb09] (dynamic, evolutionary and community-based) for an approach to handle semantic ambiguity.

8.1 Limitations

Some general problems arise with the use of our methodology. As briefly discussed some domain specific terms that are completely understandable for a domain expert are not contained in WordNet. Another example is the word "R/3" which is specific to the company of SAP and was mapped to the SynSet containing the physical gas constant "r". This also shows that the simple disambiguation heuristic introduced in section 4 is not perfect as well and could be exchanged for a more sophisticated one. But, due to the fact that both WordNet and the BNC corpus, used for the experiments, were designed for general English such problems will always arise. A considerable reduction can probably only be achieved by extending WordNet with domain specific terms and by changing the corpus that is used to determine the probabilities for the semantic relatedness measure. A good starting point for such a corpus would be e.g. a reference manual or domain specific literature.

Another problem is, that semantic relatedness strongly depends on subjective evaluations. Thus, even if in general the computed similarity is accepted, individual persons could have a different perspective and could object to the identified relations. Another alternative is to give the modeler the possibility to adjust the meaning of a word, if he does not agree with the automatic determination. Finally, as the semantic quality and understandability of a model strongly depend on subjective evaluations, too, these metrics can never give a perfect answer, but they are able to point a user or modeler to peculiar labels and provide hints to increase the general understandability and alignability of the model.

8.2 Further Research

Further research could include identifying the possibilities of the information within WordNet which were not used in this paper. Additionally, concepts that are already present, like the determined similarity relation between the labels, could be used to identify labels which do not seem to have a high correlation to their environment. This could lead to the identification of parts that should probably be extracted to a sub-model or integrated with other parts. Another focus will be to conduct an empirical study to verify the usefulness of the metrics.

To conclude, it will also be necessary to combine the measures presented above with techniques that also evaluate the syntactical structure of the label that is investigated, to provide an overall quantitative quality metric, An attempt to that is currently being evaluated at our institute [LMS].

References

- [BGBZ07] Jordan Boyd-Graber, David Blei, and Xiaojin Zhu. A Topic Model for Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1024–1033, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- [BH01] A. Budanitsky and Graeme Hirst. Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures, 2001.
- [DHLS08] Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle. pages 23–38, 2008.
- [HS06] I. Hadar and P. Soffer. Variations in conceptual modeling: classification and ontological analysis. In *J AIS*, pages 568–592, 2006.
- [JC97] J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*, pages 9008+, September 1997.
- [KJ03] J. Krogstie and H.D. Jorgensen. Quality of Interactive Models. In *Conceptual Modeling - ER 2002, Workshops of the 21st International Conference on Conceptual Modeling, Tampere, Finland*, volume LNCS 2784, pages 351–363. Springer Verlag Berlin Heidelberg, 2003.
- [KSJ06] John Krogstie, Guttorm Sindre, and Håvard Jørgensen. Process models representing knowledge for action: a revised quality framework. *Eur. J. Inf. Syst.*, 15(1):91–102, 2006.
- [Lin98] Dekang Lin. An Information-Theoretic Definition of Similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.
- [LMS] Henrik Leopold, Jan Mendling, and Sergey Smirnov. Measuring Label Quality using Part of Speech Tagging. to be published autumn/winter 2009.
- [LSS94] Odd Ivar Lindland, Guttorm Sindre, and Arne Sølvberg. Understanding Quality in Conceptual Modeling. *IEEE Software*, 11(2):42–49, 1994.
- [Mil95] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [MR08a] Jan Mendling and Jan C. Recker. Towards Systematic Usage of Labels and Icons in Business Process Models. In Terry Halpin, Erik Proper, John Krogstie, Xavier Franch, Ela Hunt, and Remi Coletta, editors, *CAiSE 2008 Workshop Proceedings - Twelfth International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2008)*, volume 337, pages 1–13. CEUR-WS.org, June 16-17 2008.
- [MR08b] Jan Mendling and Hajo A. Reijers. The Impact of Activity Labeling Styles on Process Model Quality. In *SIGSAND-EUROPE*, pages 117–128, 2008.
- [MRR09] J. Mendling, H. A. Reijers, and J. Recker. Activity labeling in process modeling: Empirical insights and recommendations. *Information Systems*, April 2009.

- [MS08] Jan Mendling and Mark Strembeck. Influence Factors of Understanding Business Process Models. In Dieter Fensel Witold Abramowicz, editor, *Business Information Systems, 11th International Conference, BIS 2008, Innsbruck, Austria, May 2008*, pages 142–153. Springer-Verlag, 2008.
- [Pfe08] D. Pfeiffer. Semantic Business Process Analysis - Building Block-based Construction of Automatically Analyzable Business Process Models. Münster, 2008.
- [Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [Reb09] Michael Rebstock. Technical opinion: Semantic ambiguity: Babylon, Rosetta or beyond? *Commun. ACM*, 52(5):145–146, 2009.
- [Scr06] Aaron D. Scriver. Semantic Distance in WordNet: A Simplified and Improved Measure of Semantic. Master’s thesis, University of Waterloo, Waterloo, Ontario, Canada, 2006.
- [Sus93] Michael Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *proceedings of the second international conference on Information and knowledge management (CIKM '93)*, pages 67–74, New York, NY, USA, 1993. ACM.
- [VCR⁺] Irene Vanderfeesten, Jorge Cardoso, Hajo A. Reijers, Wil Van Der Aalst, and Jan Mendling. Quality Metrics for Business Process Models.
- [VRM⁺08] Irene Vanderfeesten, Hajo Reijers, Jan Mendling, Wil van der Aalst, and Jorge Cardoso. On a Quest for Good Process Models: The Cross-Connectivity Metric. *Advanced Information Systems Engineering*, pages 480–494, 2008.
- [Yar92] David Yarowsky. Word-sense disambiguation using statistical models of Roget’s categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 454–460, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

Stilisierte Fakten der Ereignisgesteuerten Prozesskette – Anwendung einer Methode zur Theoriebildung in der Wirtschaftsinformatik

Constantin Houy, Peter Fettke, Peter Loos
Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)
Stuhlsatzenhausweg 3, Geb. D3 2
66123 Saarbrücken
Constantin.Houy@iwi.dfki.de
Peter.Fettke@iwi.dfki.de
Peter.Loos@iwi.dfki.de

Abstract: Ereignisgesteuerte Prozessketten (EPK) bilden ein Artefakt der Wirtschaftsinformatik, das sich seit seiner Einführung 1992 stets weiterentwickelt hat und zu dem über den gesamten Nutzungszeitraum Erfahrungen und artefakt-relevantes Wissen dokumentiert wurden. Umfassende und ausführliche Ursache-Wirkungs-Zusammenhänge oder gar Theorien wurden in diesem Bereich noch nicht formuliert. Der vorliegende Beitrag beschäftigt sich mit dem aus der Volkswirtschaftslehre stammenden Konzept stilisierter Fakten, um vorhandenes Wissen und Erfahrungen zur EPK zu erfassen, zu kondensieren, zu verdichten und in diesem Bereich die Bildung von Theorien voranzutreiben. Das Konzept stilisierter Fakten kann innerhalb dieses Prozesses bedeutende Grundlagen schaffen und auch die Bildung von Theorien auf dem Gebiet der Wirtschaftsinformatik unterstützen. Bei stilisierten Fakten handelt es sich um breit gestützte und generalisierte Wissens- oder Erfahrungselemente, die essentielle Eigenschaften eines Phänomens darstellen. Die Anwendung des Konzeptes in grundlegenden Bereichen, wie der Modellierung von Geschäftsprozessen mit der EPK ermittelt interessante Zusammenhänge, unterstützt die Theoriebildung und kann somit bedeutende Grundlagen für die Weiterentwicklung von Technologien im Sinne einer gestaltungsorientierten Forschung liefern, die sich an vorliegenden Erkenntnissen orientiert.

1. Einleitung

1.1 Ausgangssituation

Die Gestaltungsorientierung wird im Allgemeinen als das zentrale Charakteristikum der deutschsprachigen Wirtschaftsinformatik verstanden. Dieser Überzeugung folgend soll wirtschaftsinformatische Forschung insbesondere innovative Artefakte konstruieren oder gestalten, die nützlich und hilfreich bei der Lösung praktischer Probleme sind [He04]. Die Wirtschaftsinformatik wird deshalb häufig als angewandte Wissenschaft bezeichnet. Allerdings gewinnt auch die Entwicklung von Theorien im Sinne einer Grundlagenforschung im Rahmen der Wirtschaftsinformatik an Bedeutung, da sie den Gestaltungs-

prozess von Informationssystemen sinnvoll unterstützen kann. In ihrem Diskussionsbeitrag „Was ist eigentlich Grundlagenforschung in der Wirtschaftsinformatik?“ verdeutlichen Winter, Krcmar, Sinz, Zelewski und Hevner, dass „hinsichtlich der Grundlagenforschung in der Wirtschaftsinformatik nicht das *ob* fraglich ist, wohl aber das *wie*“ [Wi09]. Die Identifikation von Ursache-Wirkungs-Beziehungen und deren technologische Anwendung als umgesetzte Ziel-Mittel-Beziehungen [Ch94] kann für die Wirtschaftsinformatik ebenso von Bedeutung sein wie eine nachträgliche theoretische Fundierung neu entwickelter Artefakte, da beide Varianten Gestaltungsprozesse unterstützen können [Kr09]. In diesem Zusammenhang weist Zelewski in der Diskussionsrunde auf die besondere Eignung des Konzeptes stilisierter Fakten („stylized facts“) zur Identifikation interessanter Untersuchungsobjekte im Rahmen der Grundlagenforschung und Theoriebildung hin [Ze09]. Dieses aus der Volkswirtschaftslehre stammende Konzept und die zugehörige Methode bieten einen Ansatzpunkt, unsystematisch nebeneinander stehende Forschungsergebnisse zu bündeln, eine „Kartografierung des vorhandenen realtheoretischen Wissens“ vorzunehmen und die Theoriebildung innerhalb eines Forschungsgebietes voranzutreiben. Des Weiteren unterstützt es die Identifikation relevanter offener Forschungsfragen in einem Interessensgebiet [WL07]. Das Konzept ist vom Forschungsgebiet unabhängig einsetzbar und kann somit auch im Kontext der Wirtschaftsinformatik angewendet werden, um vorhandenes Wissen wirtschaftsinformatischer Forschung zu erfassen und theoriebildend aufzuarbeiten.

Die Ereignisgesteuerte Prozesskette (EPK) wurde seit ihrer Einführung durch Keller et al. [KNS92] stetig weiterentwickelt und durch bestimmte Funktionalitäten erweitert. Die Methode findet eine breite Akzeptanz im Zusammenhang mit der Modellierung von Geschäftsprozessen und somit bei der Gestaltung prozessorientierter Informationssysteme. Es existiert inzwischen eine Fülle wissenschaftlicher Publikationen zur EPK, die sowohl konzeptioneller als auch empirischer Natur sind. Im Rahmen ihrer Nutzung wurden zahlreiche Erfahrungen dokumentiert, die z. B. im Rahmen von Fallstudien zum Einsatz von EPK [ESS99] oder anderen empirischen Methoden wie Laborexperimenten ermittelt wurden [GD09]. Umfassende Ursache-Wirkungs-Beziehungen wurden in diesem Bereich allerdings noch nicht dokumentiert, was dieses Gebiet zu einem interessanten Untersuchungsbereich zur Anwendung des Konzeptes stilisierter Fakten macht.

1.2 Ziele des Beitrags

Die Ziele des vorliegenden Beitrags liegen darin, die Anwendbarkeit des Konzeptes stilisierter Fakten auf dem Gebiet der Wirtschaftsinformatik und insbesondere auf dem Gebiet der EPK zu untersuchen. In diesem Zusammenhang verarbeitet er eine Reihe von Beiträgen, die Wissen und Erfahrungen zur Modellierungsmethode der Ereignisgesteuerten Prozesskette formulieren, um breit gestützte und generalisierte Wissens Elemente zur EPK zu extrahieren und weiter untersuchen zu können. Die erarbeiteten Wissens Elemente werden darauffolgend auf potentielle Zusammenhänge untersucht, um somit die Bildung theoretischer Konstrukte im Kontext der Geschäftsprozessmodellierung mit der EPK voranzutreiben. Vor diesem Hintergrund lassen sich die Ziele des Beitrags formulieren, die auf folgende Fragestellungen eingehen:

1. Wie lässt sich das Konzept stilisierter Fakten im Kontext der Wirtschaftsinformatik und insbesondere auf dem Gebiet der EPK anwenden?
2. Welche breit gestützten Generalisierungen von EPK-relevantem Wissen lassen sich aus existenter Literatur ableiten?
3. Inwiefern wird dieses EPK-relevante Wissen in der Community akzeptiert und inwiefern stützen sich bestimmte Aussagen durch voneinander unabhängige Untersuchungen?
4. Welche Zusammenhänge oder auch Inkonsistenzen und Widersprüche lassen sich zwischen einzelnen Wissens-elementen (stilisierten Fakten) im Rahmen der Theoriebildung erkennen?

1.3 Aufbau des Beitrags

Nach dieser Einleitung wird in Kapitel 2 der zugrunde gelegte Forschungsansatz expliziert und das Konzept stilisierter Fakten eingeführt. Vorhandene Potentiale für den Bereich der wirtschaftsinformatischen Theoriebildung werden aufgezeigt. Kapitel 3 wendet das Konzept zur Ableitung stilisierter Fakten der Ereignisgesteuerten Prozesskette an und stellt potentielle Zusammenhänge dar. Kapitel 4 diskutiert sowohl ermittelte stilisierte Fakten zur EPK als auch Erkenntnisse, die im Zusammenhang mit der Anwendung der Methode erlangt wurden, und präsentiert Implikationen für die zukünftige Forschung auf diesem Gebiet, bevor Kapitel 5 die dargestellten Inhalte zusammenfasst.

2. Forschungsansatz anhand des Konzeptes stilisierter Fakten

2.1 Allgemeine Beschreibung des Konzeptes

Stilisierte Fakten stellen breit gestützte und generalisierte Wissens- oder Erfahrungselemente dar, die essentielle Eigenschaften eines untersuchten Phänomens beschreiben [HMS07]. Beispielsweise stellt der empirisch breit gestützte Zusammenhang zwischen dem langfristigen Verzicht auf einen Arbeitgeberwechsel bei japanischen Unternehmen und eine im Vergleich zu amerikanischen Unternehmen vielfach höhere Gehaltssteigerung ein typisches stilisiertes Faktum dar [Ao88].

Das Konzept stilisierter Fakten geht nach herrschender Meinung auf Kaldor zurück, der es im Kontext der makroökonomischen Wachstumstheorie anwendete [Ka61]. In diesem Zusammenhang argumentiert Kaldor, dass im Rahmen der Modell- und auch der Theoriebildung die Erläuterung charakteristischer Merkmale („stylized facts“) eines zu untersuchenden Phänomens das bedeutendste Ziel darstellt. Schwerin weist darauf hin, dass das Konzept bereits einige Vorläufer in der empirischen Konjunkturforschung hatte [Sc01]. Das Konzept stilisierter Fakten ist nicht nur in der Volkswirtschaftslehre von methodologischer Bedeutung, sondern wurde auch in der Betriebswirtschaftslehre schon mehrfach angewendet, z. B. bei [CD99], [WL07].

Durch die Grundidee der Stilisierung ermöglicht das Konzept eine einheitliche Sicht („stylized view“) auf charakteristische Merkmale eines Erkenntnisobjektes. Im Rahmen des Prozesses der Ableitung stilisierter Fakten werden ähnliche Aussagen verschiedener Beiträge zu einem Wissensgebiet abstrahiert, um breit gestützte Tendenzen zu identifizieren, die über verschiedene Darstellungen eines Sachverhaltes hinweg erkennbar sind.

Heine et al. entwickeln unter Berücksichtigung eines typischen Vorgehensmodells zur Erstellung von Metaanalysen eine Operationalisierung für eine methodisch korrekte Ableitung stilisierter Fakten, die sie auf betriebswirtschaftliche Themen ausrichten [HMS07]. Diese Erweiterung begründet sich durch einige methodische Mängel, die das ursprüngliche Konzept von Kaldor aufweist. Der im Folgenden dargestellte Generierungsprozess orientiert sich am Beitrag von [WL07], der die Systematisierung von Heine et al. übernimmt, allerdings die Phase der „Extraktion empirischer Ergebnisse zum Phänomen“ im Vergleich zu Heine et al. etwas detaillierter in drei Unterphasen darstellt. Des Weiteren wurde im vorliegenden Beitrag der Prozess um die Phase der Ergebnisinterpretation zur Erarbeitung von Ursache-Wirkungs-Beziehungen erweitert.

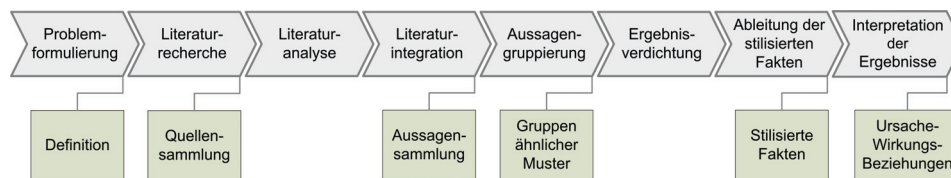


Abbildung 1: Generierungsprozess stilisierter Fakten

Wie bereits erwähnt weist Kaldors ursprüngliches Konzept im Gegensatz zu dem hier verwendeten Vorgehensmodell einige methodologische Unklarheiten in Bezug auf eine konkrete Operationalisierung auf, weshalb es häufig aus theoretischer Hinsicht kritisiert wurde. Ausführliche Kritik übt unter anderem Schwerin [Sc01], dessen umfassende Darstellung auch in die Überlegungen von Heine et al. einfließen. Schwerins Arbeit liefert eine detaillierte Erweiterung des ursprünglichen Konzeptes von Kaldor und stellt einige Grundanforderungen an eine akzeptable und methodisch „saubere“ Ableitung stilisierter Fakten, die vor allem für empirische Arbeiten von Bedeutung sind:

1. Offenlegung des Generierungsprozesses,
2. Methodenübergreifende Faktengenerierung,
3. Vollständigkeit der Faktenmenge,
4. Hinreichender Konsens über jedes stilisierte Faktum und
5. Offenheit der Faktenmenge.

Die Einhaltung der dargestellten Systematik und dieser Forderungen unterstützt somit eine ausgeprägte intersubjektive Nachvollziehbarkeit der ermittelten Ergebnisse.

Das Konzept bedient sich im Rahmen der Faktengewinnung in der Regel der Methoden der qualitativen Inhaltsanalyse. Diese beschäftigt sich mit der systematischen Erhebung und Auswertung von verfügbaren Materialien. Im vorliegenden Beitrag handelt es sich bei den untersuchten Materialien um Literatur. Prinzipiell lassen sich auch andere Materialien wie Bilder oder Filme zu einem bestimmten Themenkomplex im dargestellten Sinne qualitativ untersuchen. Ferner können auch formale und quantitative Methoden zur Ableitung stilisierter Fakten angewendet werden.

Die Ableitung stilisierter Fakten geschieht durch die Zusammenfassung von einzelnen, sich im Detail unterscheidenden Aussagen zu einer gebündelten und abstrahierten Eigenschaft. Zu beachten ist dabei, dass sich diese Eigenschaft durch ein hinlängliches Maß an Repräsentativität und inhaltlicher Übereinstimmung rechtfertigen lässt. In diesem Zusammenhang existiert keine einheitliche Regel oder Grenze, ab wann eine Implikation als hinreichend repräsentativ gelten kann. Dies ist im jeweiligen Kontext zu entscheiden [WL07]. Dieser durch subjektive Entscheidungen geprägte Ableitungsprozess ist durch den Nachweis eines entsprechenden Konsens von Experten, die im Forschungsfeld tätig sind, abzusichern.

2.2 Anwendung des Konzeptes in der vorliegenden Untersuchung

Das vorhandene Wissen zum hier untersuchten Gegenstand der EPK liegt zu einem großen Teil in Form von Texten, meist wissenschaftliche Untersuchungen, vor. Um eine systematische Vorgehensweise zu unterstützen, wird die Stichprobe der zu untersuchenden Fachbeiträge gemäß dem dargestellten Vorgehensmodell zur Ableitung stilisierter Fakten ermittelt und danach deren Inhalte analysiert. Im Gegensatz zur rein subjektiven Interpretation von Texten sind für eine empirische Inhaltsanalyse im Allgemeinen sowie für das Konzept stilisierter Fakten im Speziellen eine stringent durchgehaltene Systematik und die intersubjektive Nachvollziehbarkeit der ermittelten Inhalte von großer Bedeutung [Fr01]. Die Anwendung der explizierten Regeln bezüglich der Stichprobe und Materialauswertung durch verschiedene Subjekte sollte im Idealfall zu ähnlichen Resultaten führen [Di07]. Ein häufig angebrachter Kritikpunkt qualitativer Inhaltsanalysen richtet sich auf die Kontextabhängigkeit der Bedeutung von Zeichen und Symbolen. Diese Bedeutungen können von unterschiedlichen Subjekten auch unterschiedlich aufgefasst und interpretiert werden. Zur Einhaltung einer konsistenten Perspektive auf die untersuchte Literatur, wurden sämtliche Texte zunächst von einem Autor untersucht und die stilisierten Fakten abgeleitet. Anschließend wurde die Konsistenz der abgeleiteten stilisierten Fakten nochmals durch einen weiteren Autor geprüft.

Häufig wird zur Durchführung von qualitativen Inhaltsanalysen empfohlen, die untersuchten Materialien im Hinblick auf eine a priori explizierte Fragestellung oder Hypothese zu untersuchen [Di07]. So soll auch im Rahmen des Konzeptes stilisierter Fakten die Problemstellung im ersten Prozessschritt formuliert werden. Im Kontext der Anwendung des Konzeptes zur Theorieentwicklung, wie im vorliegenden Beitrag, wird das vorhandene Material allerdings explorativ gesichtet, um gehäuft auftretende Muster möglichst theorieunabhängig zu identifizieren („Theorieutralität“) [Sc01]. Eine mehrmalige Durchsicht der Literatur fördert die Identifikation sämtlicher potentieller Faktenkandidaten. Die Generierung stilisierter Fakten stellt sich im Rahmen der Theoriebildung als ein iterativer Prozess dar, der mehrmaliges Untersuchen des vorhandenen Materials erfordert, um belastbare Aussagen extrahieren zu können.

Im Rahmen des Konzeptes stilisierter Fakten wird zur Theoriebildung vor allem die Verwendung empirischer Befunde in den Vordergrund gestellt. Schwerin deutet darüber hinaus an, dass stilisierte Fakten auch durch Befragungen von Experten ermittelt werden können [Sc01]. Es erscheint somit nicht nur die Verwendung von Erfahrungen, die aus empirischen Studien abgeleitet wurden, sondern unter anderem auch Lehrbuchwissen zur Ableitung stilisierter Fakten geeignet, da dieses zweifelsohne als etablierte und konsolidierte Expertenauffassung gelten kann. Vor dem Hintergrund einer solchen Auffassung kann jegliche schriftlich fixierte Aussage zur Ableitung stilisierter Fakten herangezogen werden, sofern diese auf Erfahrungen eines Experten beruht und sich im Rahmen der Konsensanalyse behaupten kann. Insofern werden im Kontext der vorliegenden Untersuchung auch nicht-empirische Arbeiten bei der Herleitung stilisierter Fakten berücksichtigt, wenn sie das genannte Kriterium erfüllen.

Der gegenwärtige Beitrag berücksichtigt bei der Erhebung relevanter Arbeiten die Anforderungen der Forschungsmethode „Review“ [Fe06]. Zur Erhebung der Fakten werden relevante Beiträge zur Ereignisgesteuerten Prozesskette untersucht. Da eine Berücksichtigung aller existierenden Beiträge im Rahmen dieses Artikels als nicht zweckmäßig und aus Platzgründen nicht realisierbar erscheint, wird eine Auswahl aktueller Beiträge untersucht. Die Auswahl beschränkt sich auf die Jahre 1999 bis 2009. Dies führt einerseits zwar dazu, dass eventuell potentielle Fakten nicht berücksichtigt werden können, andererseits wird somit aber ein hoher Gegenwartsbezug der ermittelten Fakten gewährleistet. Die untersuchte Auswahl von Beiträgen stützt sich auf die Bibliographie zur EPK auf der Website des GI-Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK) (<http://www.epk-community.de>). Diese Bibliographie wird seit dem Jahre 2005 auf der genannten Website nicht mehr erweitert. Im Rahmen des vorliegenden Beitrags wird die Menge der in der Bibliographie enthaltenen Konferenz- und Journalbeiträge um einige, auch internationale Journalartikel und Konferenzbeiträge zur EPK aus dem genannten Zeitraum ergänzt. Darin enthalten sind auch sämtliche Proceedings des GI-Workshops zur EPK aus dem genannten Zeitraum. Weitere Artikel wurden systematisch in zwei Datenbanken anhand des Suchbegriffes „EPK“, bzw. „EPC“ recherchiert und auf ihre Relevanz im Bereich der Ableitung von stilisierten Fakten im oben beschriebenen Sinne überprüft. Bei den Datenbanken handelt es sich um die im deutschsprachigen Raum häufig eingesetzte wirtschaftswissenschaftliche Datenbank „WisoNet“ und die international ausgerichtete Datenbanken „Ebsco Host“. Die somit verfügbare Literaturliste umfasste direkt nach der Recherche insgesamt 94 Artikel. Nach der ersten Relevanzprüfung der Beiträge wurden noch 86 Artikel für die folgende Untersuchung berücksichtigt. Bei diesen Artikeln handelt es sich vor allem um konzeptionelle Arbeiten, Experimente, Methoden-erweiterungen, Umfragen und Fallstudien.

Zur Untersuchung und Bewertung des bestehenden Konsenses zu einem stilisierten Faktum (Konsensanalyse) wird im vorliegenden Beitrag das Klassifikationskriterium der Anzahl von bestehenden Untersuchungen, bzw. thematischen Beiträgen von Experten des Forschungsfeldes herangezogen, die die Aussage stützen. Die Anzahl übereinstimmender Aussagen ermöglicht hinreichende Rückschlüsse auf die Qualität des stilisierten Faktums [WL07]. Die jeweiligen ermittelten Fakten werden als gut gestützt (A), durchschnittlich gestützt (B) und schwach gestützt (C) klassifiziert. Zu stilisierten Fakten aus

der Kategorie C besteht noch stärkerer Überprüfungsbedarf im Vergleich zu denjenigen aus der Kategorie A, was allerdings nicht impliziert, dass es sich bei stilisierten Fakten aus der C-Kategorie um unbegründete Zusammenhänge handeln muss. Stilisierte Fakten aus der Kategorie C stellen angenommene Tendenzen dar, die es in weiteren Untersuchungen zu untermauern oder abzulehnen gilt. Zur Festlegung einer Regel wurden im Rahmen der vorliegenden Untersuchung folgende Grenzen angenommen:

1. Kategorie A: mindestens 10 Quellen, die eine Aussage stützen,
2. Kategorie B: mindestens 5 Quellen, die eine Aussage stützen und
3. Kategorie C: mindestens 2 Quellen, die eine Aussage stützen.

Dass Experten auf einem Forschungsgebiet ihre Erkenntnisse aus vorangegangenen Publikationen in der Regel in späteren Beiträgen wieder aufgreifen, ist im Kontext der Konsensanalyse zu berücksichtigen. Die vorliegende Arbeit beachtet analog zur Arbeit [Sc01] mehrere Beiträge desselben Autors oder der exakt selben Autorengruppe zur gleichen Aussage in der Zählung deshalb lediglich einmal.

3. Stilisierte Fakten der EPK

3.1 Ableitung stilisierter Fakten der EPK

Folgende Ausführungen legen den Faktengenerierungsprozess offen, bevor die einzelnen ermittelten stilisierten Fakten präsentiert werden. Dazu wurde die Aussage „Der Einsatz der EPK unterstützt die Kommunikation der Beteiligten im Rahmen der Analyse von Geschäftsprozessen“ exemplarisch hergeleitet.

Quelle	Aussage als Zitat
[Ri00b]	"Event-driven Process Chains, therefore, create a common platform for communication and the analysis of ideas beyond the boundaries of both application and information-system domains."
[DR01]	"Typically the processes are described with the help of a semiformal, graphical language such as the Event-driven Process Chains (EPCs) by Scheer. This approach provides a suitable medium for the communication between the participants: the domain experts and the IT specialists."
[LF01]	"One of the main advantages of the EPC is that it is both powerful and easily understandable for end-users. EPCs are often used for capturing and discussing business processes with people who have never been trained in any kind of modeling technique [...]."
[De02]	"The objective of the approach presented is the use of EPCs as communication language between the domain experts and the application developers throughout the whole procedure."
[FRS02]	"[EPK] können weiters als Kommunikationsplattform zwischen Fachbereich und IT-Abteilung dienen und können sowohl zu Analyse Zwecken entwickelt, als auch bei Designaufgaben eingesetzt werden [...]."
[GL05]	"Ereignisgesteuerte Prozessketten (EPKs) wurden [...] entwickelt, um Geschäftsprozesse zu modellieren. Dies reicht zwar aus, um Prozesse zu dokumentieren und über die Modelle zu diskutieren, [...]."
[LK06]	"The EPC has been developed for modelling business processes with the goal to be easily understood and used by business people. [...] Only the EPC provides an explicit notation element for traditional resources and is therefore well suited for process analysis."
[TD06]	"Die [...] Ereignisgesteuerten Prozesskette wurde zunächst als eine nicht vollständig formalisierte Notation entwickelt und ohne eine feste formale Semantik benutzt. Zur Dokumentation von Prozessen und zur Verwendung der Modelle als Diskussionsgrundlage ist dies ausreichend."
[MDA07]	"It is important to realize that the language is not intended to be a formal specification of a business process. Instead, it serves mainly as a means of communication."
[KDW08]	"Ereignisgesteuerte Prozessketten (EPKs) haben sich als Modellierungstechnik etabliert. Durch ihre Syntax und Semantik ermöglichen sie eine effiziente Kommunikation zwischen den Prozessbeteiligten."
[Re09]	"Intuitive graphical modeling techniques such as the Event-driven Process Chain (EPC) are mostly concerned with capturing and understanding processes for project scoping tasks and for discussing business requirements and process improvement initiatives with subject matter experts."
→ Stilisiertes Faktum: „Der Einsatz der EPK unterstützt die Kommunikation der Beteiligten im Rahmen der Analyse von Geschäftsprozessen.“	

Abbildung 2: Konsensanalyse eines exemplarisch dargestellten stilisierten Faktums der EPK

Elf der untersuchten Quellen beschreiben diesen Zusammenhang, weshalb das stilisierte Faktum in die Kategorie A eingeordnet und somit als ein gut gestütztes Faktum gelten kann. Es wurde vornehmlich anhand von Expertenaussagen abgeleitet. Die Aussage

wurde zunächst bei der Durchsicht der Literatur identifiziert und in die Reihe der potentiellen Kandidaten aufgenommen. Bei der Durchsicht der weiteren Arbeiten wurde die Aussage derart bestätigt, dass ein hinreichender Konsens für eine Eingliederung in die Kategorie A vorlag.

3.2 Stilisierte Fakten der EPK

Die im Folgenden dargestellten stilisierten Fakten (SF) der EPK wurden wie im Beispiel demonstriert ermittelt. Ihre Herleitung wird aus Platzgründen nicht derart ausführlich präsentiert. Die einzelnen Aussagen werden in Abbildung 3 zusammen mit den stützenden Quellen und der jeweiligen zugeordneten Kategorie angeführt.

Stilisierte Fakten der EPK	Kategorie	Stützende Quellen
SF1: „Die EPK ist eine in der Praxis weitverbreitete Modellierungssprache“	A	[Aa99], [MR00], [Ri00b], [Ri00c], [De01], [LF01], [De02], [NR02], [Ro02], [FL03], [MN03a], [MN03b], [MN03c], [MN03d], [SS03], [MBN04], [Th04], [KK05], [LMS05], [SL05], [ST05], [STA05], [SV05], [TKL05], [HOS05], [GL06], [SFO06], [TD06], [GL07], [MDA07], [SS07], [We07], [BF08], [Kr08], [KDW08], [Me08], [Pe08], [DR09], [Fe09], [Ga09], [Re09], [BSS09]
SF2: „Die EPK ist leicht und intuitiv verständlich.“	A	[Aa99], [GR00], [MR00], [Ri00b], [Ri00c], [De01], [LF01], [BAN03], [MN03a], [Ih04], [MZ05], [DJ05], [KUL06], [LK06], [MDA07], [Do07], [We07], [Kr08], [KDW08], [Me08], [Re09]
SF3: „Die EPK lässt sich derart definieren, bzw. transformieren (z. B. in Petrinetze), dass formale Analysen möglich werden, bzw. der abgebildete Prozess als Workflow ausführbar wird.“	A	[Aa99], [MR00], [De01], [DR01], [BAN03], [MN03a], [GL05], [DJ05], [HOS05], [De06], [KUL06], [SI06], [TD06], [AI07], [KK07], [Ko07], [Do07]
SF4: „Die EPK ist unklar im Ausdruck. Deshalb kann sie missverständlich sein und anders interpretiert werden als intendiert.“	A	[GR99], [Aa99], [GR00], [MR00], [Ri00b], [Ri00c], [De01], [DR01], [De02], [FRS02], [ADK02], [FLU3], [TF06], [Re09], [SFG09]
SF5: „Die Missverständlichkeit der EPK ist vor allem durch die vielfältigen Verwendungsmöglichkeiten der Konnektoren bedingt.“	A	[Ri00a], [Ri00b], [Ri00c], [ADK02], [MN03a], [MN03b], [CK04], [CFK05], [GL05], [MZ05], [De06], [MA06], [We07]
SF6: „Erweiterungen der EPK erhöhen deren Sprachausage. Solche Erweiterungen sind notwendig, um die EPK in verschiedenen Kontexten nutzbringend einsetzen zu können.“	A	[GR99], [LF01], [BO02], [Ro02], [THA02], [Be03], [FL03], [ST03], [PI04], [GRA05], [SDL05], [SL05], [SV05], [SI06], [TD06], [BF08]
SF7: „Der Einsatz der EPK unterstützt die Kommunikation der Beteiligten im Rahmen der Analyse von Geschäftsprozessen.“	A	[Ri00b], [DR01], [LF01], [De02], [FRS02], [GL05], [LK06], [TD06], [MDA07], [KDW08], [Re09]
SF8: „Die Transformation von EPK in ein praktikables Austauschformat wie XML fördert den Austausch von EPK-Modellen zwischen unterschiedlichen Systemen.“	B	[GK02], [MN02], [Me03], [MN03b], [MN03c], [MN03d], [MN04], [MBN04], [KK05], [MZ05], [Ba06]
SF9: „Syntax und Semantik der EPK sind bei ihrer Einführung nicht klar definiert worden.“	B	[Aa99], [MR00], [Ri00a], [De01], [DR01], [De02], [Ro02], [GL05], [DJ05], [Gr08]
SF10: „Die EPK dient der Dokumentation von Geschäftsprozessen und bildet somit die Grundlage eines integrierten Geschäftsprozessmanagement.“	B	[LF01], [NR02], [Ro02], [MN03a], [DJ05], [HOS05], [LK06], [AI07], [Kr08]
SF11: „Die semiformale Semantik der EPK führt zu sehr vielfältigen Ausdrucksmöglichkeiten.“	B	[Ri00b], [De02], [FL03], [ST05], [We07]
SF12: „EPK eignen sich auch zur Darstellung von Prozessen im Verwaltungsumfeld.“	C	[Th04], [BAN03], [SDL05], [STA05]
SF13: „Die EPK unterstützt die Organisationsoptimierung.“	C	[FRS02], [Ro02], [MZ05], [STA05]
SF14: „Die EPK ist seit ihrer Einführung kontinuierlich weiter formalisiert worden.“	C	[MN02], [ADK02], [Me03], [SDL05]
SF15: „Es existiert ein große Anzahl unterschiedlicher Verifikationsansätze zur EPK.“	C	[DJ05], [Do07], [Gr08], [PSW08]
SF16: „Die EPK ist für ihre Benutzer leicht anwendbar.“	C	[Ri00a], [Ri00c], [De01], [Ro02]
SF17: „Mit der EPK lassen sich Prozesse ganzheitlich abbilden.“	C	[BAN03], [SS03], [NC04]
SF18: „Mit zunehmender Erweiterung der eEPK sinkt deren Verständlichkeit.“	C	[LT04], [GRA05], [SI06]
SF19: „Die EPK wird vor allem wegen ihres ausgeprägten Toolsupports häufig verwendet.“	C	[De01], [De02], [TKL05]
SF20: „Eine hohe Konnektorendichte erhöht die Fehlerwahrscheinlichkeit in EPK und erschwert darüber hinaus deren Verständnis.“	C	[GLM06], [Me08]
SF21: „Es existiert keine formale Semantik, die die informale Semantik der EPK präzise abbilden kann“	C	[ADK02], [KI03]
SF22: „EPK sind etabliert im Rahmen der Referenzmodellierung“	C	[TKL05], [HOS05]

Abbildung 3: Stilisierte Fakten der EPK

Interessierte Leser können sich für eine ausführliche Sammlung und Darstellung der verwendeten Originalzitate für die Ableitung der stilisierten Fakten an den Erstautor dieses Beitrags wenden. Es sei an dieser Stelle betont, dass die ermittelten stilisierten Fakten nicht sämtliche für die Theoriebildung interessanten und potentiell relevanten Aussagen wiedergeben, die durch die analysierte Literatur ersichtlich wurden. Vielmehr handelt es sich bei den dargestellten Aussagen um diejenigen, über die innerhalb der wissenschaftlichen Diskussion ein gewisser Konsens besteht.

3.3 Potentiell widersprüchliche Befunde

Im Rahmen der Untersuchung wurden auch potentiell widersprüchliche Befunde identifiziert, die im Folgenden kurz skizziert werden sollen. Einige Autoren weisen auf eine gute Verwendbarkeit von EPK im Rahmen der Anforderungsanalyse von Informationssystemen hin, so z. B. [LF01], [FRS02], [SL05] und [SS07]. [De01] formuliert hingegen die Aussage, dass aufgrund des Interpretationsspielraums von EPK-Modellen diese für den Gestaltungsprozess von Informationssystemen weniger gut geeignet seien.

Im Vergleich zwischen der EPK und Aktivitätsdiagrammen äußern zwei Autoren die Ansicht, dass die EPK zur Prozessabbildung besser geeignet sei als Aktivitätsdiagramme [PI04], [Lü06]. Das genaue Gegenteil vertreten [St06] und [GD09]. Letztere Meinung wird vor allem vor dem Hintergrund der Weiterentwicklung von Aktivitätsdiagrammen der UML 2.0 gerechtfertigt. Überraschend ist in diesem Zusammenhang nur, dass die Beiträge [Lü06] und [St06] im gleichen EPK-Proceedingsband publiziert wurden und sich somit sehr wahrscheinlich auf den gleichen Entwicklungsstand der UML beziehen.

3.4 Entwicklung und Darstellung von Zusammenhängen

Da anzunehmen ist, dass zwischen den einzelnen stilisierten Fakten Zusammenhänge bestehen, werden nun bestehende Abhängigkeiten in Abbildung 4 dargestellt und im Sinne der Theoriebildung analysiert. Die Abbildung vermittelt eine komprimierte Darstellung von Ursache-Wirkungs-Hypothesen im Bereich des Forschungsfeldes der EPK, die auf Basis der ermittelten belastbaren Aussagen zur EPK entwickelt wurden. Ihre Darstellung lehnt sich an sogenannte „causal loop diagrams“ an [St00]. In die präsentierten Zusammenhänge wird des Weiteren ein grundlegender Zweck der Nutzung von EPK eingebettet, nämlich die Unterstützung und Umsetzung der Prozessorientierung, aus der eine gesteigerte Wirtschaftlichkeit eines Unternehmens oder einer Verwaltung hervorgehen soll. Eine Auswahl stilisierter Fakten wird im Sinne der Theoriebildung kausal verkettet. Ein Pluszeichen (+) deutet eine unterstützende Wirkung eines bestimmten Faktums auf ein anderes Faktum an. Ein Minuszeichen (-) deutet entsprechend eine hemmende Wirkung an.

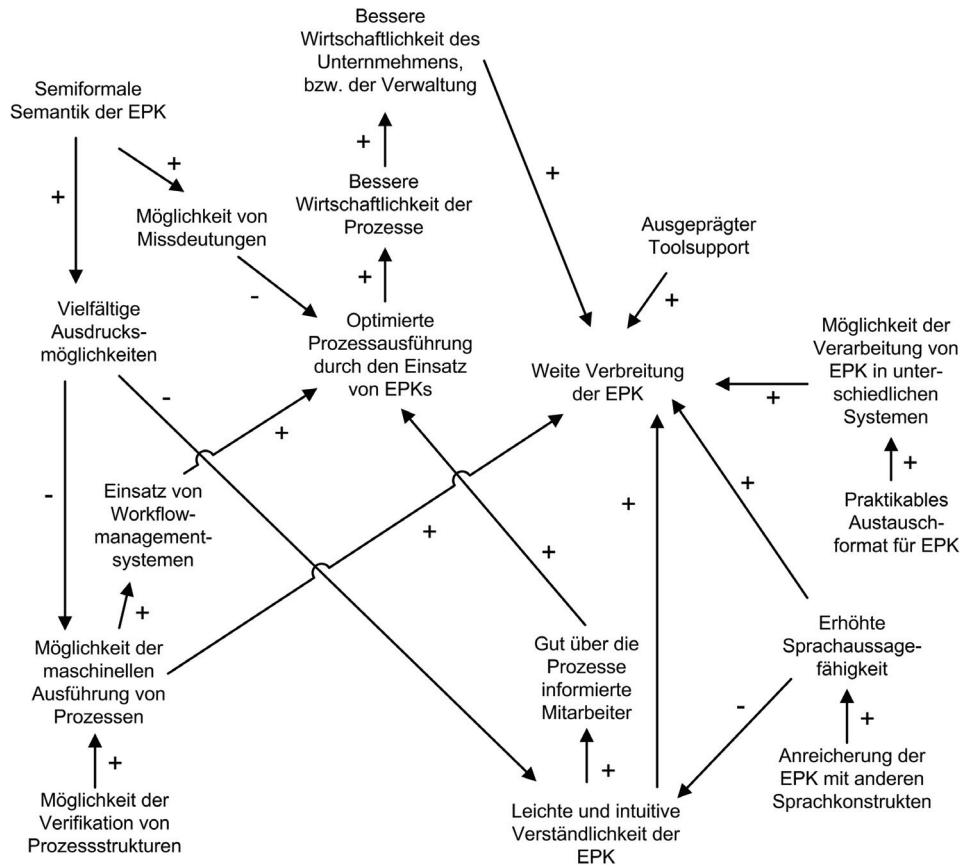


Abbildung 4: Ursache-Wirkungs-Beziehungen zwischen stilisierten Fakten der EPK

Eine umfassende und vollständige Verbalisierung der dargestellten Zusammenhänge soll an dieser Stelle aus Platzgründen nicht stattfinden. Exemplarisch seien die vermuteten Einflüsse, die die semiformale Semantik der EPK auf weitere stilisierte Fakten ausübt, genauer erläutert, um das Verständnis der kompletten Abbildung zu unterstützen. Abbildung 4 erhebt zudem keinen Anspruch auf Vollständigkeit sämtlicher denkbarer Kausalzusammenhänge zwischen den einzelnen stilisierten Fakten. Folgende Wirkungen gehen von der semiformalen Semantik der EPK aus:

1. Die semiformale Semantik der EPK fördert die Möglichkeit von Missdeutungen von Modellen durch Mitarbeiter, da die Notation Sachverhalte nicht eindeutig erfasst und die Modelle somit von unterschiedlichen Subjekten unterschiedlich interpretiert werden können. Dies wiederum kann sich negativ auf eine optimale Prozessausführung auswirken, da unterschiedliche Mitarbeiter die durch das Modell dargestellte Prozessstruktur eventuell unterschiedlich interpretieren und einen Prozess jeweils unterschiedlich umsetzen.
2. Die semiformale Semantik der EPK fördert weiterhin vielfältige Ausdrucksmöglichkeiten, was die maschinelle Ausführbarkeit von Prozessen erschwert. Je

ausdrucksmächtiger eine Notation ist, desto größer ist die Menge und Komplexität möglicher Formulierungen, die durch die Notation ausgedrückt werden können und die im Rahmen der Maschinenumsetzbarkeit zu berücksichtigen sind. Kausal damit verkettet ist auch ein komplexer werdender Einsatz von Workflowmanagementsystemen (WfMS).

3. Durch die semiformale Semantik der EPK und die daraus resultierenden vielfältigen potentiellen Ausdrucksmöglichkeiten kann sich je nach Anwendung der gegebenen Möglichkeiten die Komplexität erstellter Modelle erhöhen. Dies hat wiederum Einfluss auf die Verständlichkeit der abgebildeten Modelle. Schwer verständliche Modelle könnten einen Einfluss auf die Güte des Wissensstandes der Mitarbeiter zur Prozessstruktur haben und sich weiterhin auf die Prozessausführung auswirken.

4. Diskussion und Implikationen

Obwohl die hier angeführte Argumentation die Plausibilität der dargestellten Zusammenhänge stützt, handelt es sich nicht um „Gesetze“, die unter allen Umständen gültig sind, sondern um begründete Vermutungen. Vielmehr ist das angewendete Verfahren als ein früher Schritt im Rahmen der Entwicklung von Theorien zu verstehen. Die Untersuchung der Zusammenhänge ist somit in den Bereich der Entdeckung und noch nicht in den Bereich der Begründung theoretischer Zusammenhänge einzuordnen.

Im Rahmen der dargestellten Ableitung zeigt sich das Konzept stilisierter Fakten als ein interessanter Ansatz zur Unterstützung der Bildung von Theorien. Einzelne stilisierte Fakten, zwischen denen im Rahmen der Theorieentwicklung Ursache-Wirkungs-Beziehungen identifiziert werden sollen, können im Optimalfall als gut gestützte Aussagen gelten und bilden somit ein solides Fundament von Theorien. Die angewandte Methode enthält eine Reihe subjektiver Entscheidungen, die durch eine erhöhte Transparenz der Darlegung intersubjektiv nachvollziehbar zu gestalten ist. Diesbezüglich ist auf eine systematische und transparente Vorgehensweise bei der Entwicklung stilisierter Fakten zu achten. In diesem Zusammenhang ist die Reliabilität der Codierung der stilisierten Fakten von Bedeutung, die im Abschnitt zur Anwendung des Konzeptes diskutiert wurde. Die Autoren haben in diesem Zusammenhang verstärkt qualitativ gearbeitet. Eine Weiterentwicklung der Methode um quantitative Aspekte könnte die Reliabilität der stilisierten Fakten weiter erhöhen.

Einige hier formulierte Zusammenhänge mögen trivial erscheinen, dennoch sollten auch solche Ursache-Wirkungs-Beziehungen im Rahmen der Theoriebildung berücksichtigt werden. Die Theoriebildung sollte sich nicht nur auf die Fundierung komplexer Beziehungen beschränken, sondern sämtliche relevanten Zusammenhänge untersuchen.

Die Methode enthält einige induktive Elemente, was sich ohne eine zusätzliche Betrachtung der Begründungszusammenhänge im Bereich einer fundierten wissenschaftlichen Gewinnung von Wissen als problematisch erweist [Po94]. Um das generierte Wissen und ermittelte Zusammenhänge in den weiteren Schritten im Rahmen der Theorieentwicklung methodisch „sauber“ zu erfassen und abzusichern, ist nach plausiblen Begründungszusammenhängen zu suchen. Erste Ansätze bot hier Absatz 3.4. Eine

weitere Steigerung der Absicherung gewonnenen Wissens liefert die Anwendung der Verfahren der epistemischen Induktion bzw. Abduktion, die einen Vergleich der dargestellten Zusammenhänge mit etablierten Theorien vornehmen und die explanativ und prognostisch wahrscheinlich erfolgreichere Theorie ermitteln können. Eine solche Vorgehensweise berücksichtigt die von Popper geforderte Forschungslogik, die die wissenschaftstheoretische Position des kritischen Rationalismus proklamiert [Sc06]. Im Rahmen der weiteren Anwendung des Konzeptes stilisierter Fakten in der Wirtschaftsinformatik empfiehlt sich der Vergleich mit den verschiedenen Ansätzen im Bereich der Organisationstheorien oder der Systemtheorie, um die Entwicklung eigenständiger Theorien der Wirtschaftsinformatik voranzutreiben [Ze09].

Der Theoriebedarf auf dem untersuchten Gebiet der EPK zeigt sich auch dadurch, dass einige der untersuchten Arbeiten eigens aufgestellte Hypothesen prüfen, diese aber nicht aus Theorien deduzieren, sondern über Plausibilitätsbegründungen rechtfertigen, z. B. [GD09]. Nur wenige Arbeiten im Untersuchungsbereich dieses Beitrags bauen auf bekannten Theorien auf, wie z. B. auf dem Bunge-Wand-Weber-Modell (BWW) wie die beiden Arbeiten [GR99] und [GR00].

Es ließe sich argumentieren, dass das Aufgreifen von Expertenmeinungen ohne empirische Fundierung zur Theoriebildung als kritisch anzusehen ist. Im Abschnitt zur Methodenanwendung wurde die Aufnahme solcher Wissens Elemente in den Generierungsprozess stilisierter Fakten begründet. Dies lässt sich auf Basis der Anzahl unterschiedlicher Vertreter einer Aussage im Rahmen der Konsensanalyse rechtfertigen.

Die in der vorliegenden Arbeit durch das Konzept stilisierter Fakten ermittelten Zusammenhänge geben nicht nur eine Übersicht über relevantes Wissen zur EPK, sondern zeigen auch den weiteren Forschungsbedarf auf diesem Gebiet auf. Weniger gut gestützte stilisierte Fakten können unter Verwendung empirischer Methoden weiter untersucht werden, um das dargelegte Wissen weiter zu fundieren. Dies gilt auch für potentielle Faktenkandidaten, die sich im Rahmen der Konsensanalyse nicht als stilisierte Fakten durchsetzen konnten, da sie lediglich einmal angeführt wurden. Für die Theoriebildung interessant sind beispielsweise folgende sinngemäß wiedergegebene Aussagen:

1. Auf der Ebene der Geschäftsleitung werden EPK als zu formal und streng empfunden [Ri00c],
2. Konzepte der EPK-Verifikation finden in der Praxis kaum Anwendung [De06],
3. Gutes Layout trägt maßgeblich zur Verständlichkeit von grafischen Modellen bei [GL06],
4. EPK und BPMN sind für Anwender gleich gut verständlich [RD07] und
5. die EPK wird in Deutschland häufiger genutzt als in anderen Ländern, weil sie dort entwickelt wurde [Fe09].

Auch diese Zusammenhänge können durch zukünftige Forschung unter Verwendung empirischer Methoden weiterhin untersucht werden. Ermittelte stilisierte Fakten der EPK können teilweise auch im Bereich anderer Modellierungsmethoden gelten und die hier entfalteten Erkenntnisse sowohl für die Weiterentwicklung und Gestaltung der EPK als auch für sonstige Modellierungssprachen von Bedeutung sein.

5. Zusammenfassung und Fazit

Das Konzept der stilisierten Fakten eröffnet die Möglichkeit, die Theoriebildung in unterschiedlichen Forschungsbereichen effektiv und fundiert voranzutreiben. Die dargestellten stilisierten Fakten der EPK konsolidieren relevantes Expertenwissen und stellen dieses in abstrahierter Form dar. Im Rahmen dieses Beitrags wurde zunächst das Konzept eingeführt und bedeutende Aspekte seiner Umsetzung dargelegt. Es wurde im Laufe der Zeit in mehreren Arbeiten modifiziert, um eine wissenschaftstheoretisch fundierte Methodik zu entwickeln und zu etablieren. Auf dieser Basis aufbauend wurde das Konzept in Kapitel 3 zur Ermittlung stilisierter Fakten der EPK angewendet, um breitgestützte Generalisierungen EPK-bezogenen Wissens aus der untersuchten Literatur abzuleiten. In diesem Zusammenhang kann der allgemeine Konsens zu bestimmten Aussagen, der in der EPK-Community besteht, nachvollzogen werden. Darüber hinaus wurden auch verschiedene widersprüchliche Befunde in untersuchten Arbeiten identifiziert und erläutert. Im Anschluss daran wurde eine Reihe von Ursache-Wirkungs-Beziehungen auf dem Gebiet der praktischen und wissenschaftlichen Beschäftigung mit der EPK erarbeitet, als Übersicht dargestellt sowie ein Teil der Ansätze näher diskutiert. Die Diskussion entwickelte verschiedene Implikationen für den Einsatz des Konzeptes stilisierter Fakten in der Wirtschaftsinformatik. Darüber hinaus präsentierte dieses Kapitel einige weitere interessante Forschungsfragen, die sich durch die Anwendung des Konzeptes eröffnen.

Die Gewinnung stilisierter Fakten stellt ein anspruchsvolles Verfahren dar, das einen umfassenden Diskussionsprozess verschiedener Fachexperten unter Berücksichtigung empirischer Befunde nachvollzieht, um belastbare Aussagen zu einem bestimmten Interessensgebiet zu ermitteln und zu fundieren. Auch im Bereich der Wirtschaftsinformatik ist das Konzept zur Unterstützung der Theorieentwicklung und der Etablierung einer Grundlagenforschung der Wirtschaftsinformatik geeignet, was seine Anwendung auf das dokumentierte Expertenwissen der EPK im Rahmen dieser Arbeit zeigt. Zukünftige Arbeiten können das Konzept in anderen Bereichen anwenden. Dabei sind nicht nur bedeutende Erkenntnisse für die zukünftige Konstruktion und Gestaltung von Informationssystemen und relevanten Artefakten in diesem Kontext von Interesse, sondern auch weitere Erfahrungen bezüglich der Anwendung des Konzeptes stilisierter Fakten in der Wirtschaftsinformatik.

Literaturverzeichnis

- [Aa99] van der Aalst, W. M. P.: Formalization and verification of event-driven process chains. In: Information & Software Technology, 41 (1999) 10, S. 639-650.
- [ADK02] van der Aalst, W., Desel, J. & Kindler, E.: On the semantics of EPCs: A vicious circle. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 71-79.

- [Al07] Allweyer, T.: Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 23-38.
- [Ao88] Aoki, M.: Information, Incentives, and Bargaining in the Japanese Economy. Cambridge University Press, Cambridge 1988.
- [Ba06] Barborka, P., Helm, L., Köldorfer, G., Mendling, J., Neumann, G., van Dongen, B. F., Verbeek, E. & van der Aalst, W. M. P.: Integration of EPC-related Tools with ProM. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 105-120.
- [BAN03] Becker, J., Algermissen, L. & Niehaves, B.: Prozessmodellierung in eGovernment-Projekten mit der eEPK. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 31-44.
- [Be03] Becker, J., Delfmann, P., Falk, T. & Knackstedt, R.: Multiperspektivische ereignisgesteuerte Prozessketten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 45-60.
- [BO02] Brabänder, E. & Ochs, H.: Analyse und Gestaltung prozessorientierter Risikomanagementsysteme mit Ereignisgesteuerten Prozessketten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 17-35.
- [BSS09] vom Brocke, J., Sonnenberg, C. & Simons, A.: Wertorientierte Gestaltung von Informationssystemen: Konzeption und Anwendung einer Potenzialmodellierung am Beispiel Serviceorientierter Architekturen. In: Wirtschaftsinformatik, 51 (2009) 3, S. 261-272.
- [BF08] Brüning, J. & Forbrig, P.: Zur Automatischen Ermittlung von Testszenarien aus EPK-Schemata. In: (Loos, P., Nuttgens, M., Turowski, K. & Werth, D., Hrsg.): Proceedings of the Workshops colocated with the MobIS2008 Conference: Including EPK2008, KobAS2008 and ModKollGP2008. Saarbrücken, Germany 2008, S. 31-43.
- [Ch94] Chmielewicz, K.: Forschungskonzeptionen der Wirtschaftswissenschaft. Schäffer-Poeschel, Stuttgart 1994.
- [CD99] Colombo, M. G. & Delmastro, M.: Some stylized facts on organization and its evolution. In: Journal of Economic Behaviour & Organization, 40 (1999) 3, S. 255-274.
- [CFK05] Cuntz, N., Freiheit, J. & Kindler, E.: On the semantics of EPCs: Faster calculation for EPCs with small state spaces. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 7-23.
- [CK04] Cuntz, N. & Kindler, E.: On the semantics of EPCs: Efficient calculation and simulation. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2004 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Proceedings des GI-Workshops und Arbeitskreistreffens. Luxemburg 2004, S. 7-26.
- [De01] Dehnert, J.: Four Systematic Steps Towards Sound Business Process Models. In: Hrsg.): Proceedings of the 2nd International Colloquium on Petri Net Technologies for Modelling Communication Based Systems. Berlin 2001, S. 55-64.
- [De02] Dehnert, J.: Making EPCs fit for Workflow Management. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 51-69.
- [DR01] Dehnert, J. & Rittgen, P.: Relaxed Soundness of Business Processes. In: (Dittrich, K. L., Geppert, A. & Norrie, M. C., Hrsg.): Proceedings of the 13th Conference on Advanced Information Systems Engineering (CAISE'01). Interlaken, Schweiz 2001, S. 157-170.

- [De06] Denne, S.: Verifying Properties of (Timed) Event Driven Process Chains by Transformation to Hybrid Automata. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 157-176.
- [Di07] Diekmann, A.: Empirische Sozialforschung - Grundlagen, Methoden, Anwendungen. Rowohlt, Reinbek bei Hamburg 2007.
- [DJ05] van Dongen, B. F. & Jansen-Vullers, M. H.: EPC Verification in the ARIS for MySAP reference model database. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 24-40.
- [Do07] van Dongen, B. F., Jansen-Vullers, M. H., Verbeek, H. M. W. & van der Aalst, W. M. P.: Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. In: Computers in Industry, 58 (2007) 6, S. 578-601, hier verwendet: elektronischer Preprint: <http://is.tn.tue.nl/staff/wvdaalst/publications/p392.pdf>, letzter Abruf: 21.10.2009.
- [DR09] Dünnebacke, D. & Rhensius, T.: Unternehmensindividuelle Auswahl von Prozessmodellierungstools. In: ERP Management, 2 (2009) S. 47-51.
- [ESS99] Ege, C., Seel, C. & Scheer, A.-W.: Standortübergreifendes Geschäftsprozessmanagement in der öffentlichen Verwaltung. In: (Scheer, A.-W., Hrsg.) *Veröffentlichungen des Instituts für Wirtschaftsinformatik*. Universität des Saarlandes, Saarbrücken 1999.
- [Fe06] Fettke, P.: State-of-the-Art des State-of-the-Art. Eine Untersuchung der Forschungsmethode „Review“ innerhalb der Wirtschaftsinformatik. In: Wirtschaftsinformatik 48 (2006) 4, S. 257-266.
- [Fe09] Fettke, P.: Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. In: zfbf - Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, 61 (2009) 8, S. 550-580.
- [FL03] Fettke, P. & Loos, P.: Ontologische Evaluierung von Ereignisgesteuerten Prozessketten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 61-78.
- [FRS02] Fichtenbauer, C., Rumpfhuber, M. & Stary, C.: Sprachgerechte unternehmensnahe Modellierung von Ereignisgesteuerten Prozessketten - Zur adäquaten Aus- und Weiterbildung von ModelliererInnen. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 109-118.
- [Fr01] Früh, W.: Inhaltsanalyse. Theorie und Praxis. UTB, Stuttgart 2001.
- [Ga09] Gadatsch, A.: Integriertes Geschäftsprozess- und Workflow-Management – Konzeption, Rollen und organisatorische Einbindung. In: HMD- Praxis der Wirtschaftsinformatik, 266 (2009) S. 35-42.
- [GK02] Geissler, M. & Krüger, A.: Eine XML-Notation für Ereignisgesteuerte Prozessketten (EPK). In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 81-86.
- [GRA05] Gottschalk, F., Rosemann, M. & van der Aalst, W. M. P.: My own process: Providing dedicated views on EPCs. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 156-175.
- [GR99] Green, P. & Rosemann, M.: An Ontological Analysis of Integrated Process Modelling. In: (Jarke, M. & Oberweis, A., Hrsg.): Proceedings of the 11th International Conference on Advanced Information Systems Engineering. Heidelberg, Germany 1999, S. 225-240.

- [GR00] Green, P. & Rosemann, M.: Integrated Process Modelling: An Ontological Evaluation. In: Information Systems, 25 (2000) 2, S. 73-87.
- [GD09] Groß, A. & Dörr, J.: Experimenteller Vergleich zweier Notationen zur Prozessmodellierung: Ereignisgesteuerte Prozessketten vs. UML-Aktivitätsdiagramme. In: Software-technik-Trends, 29 (2009) 1, S. 13-14.
- [GL05] Gruhn, V. & Laue, R.: Einfache EPK-Semantik durch praxistaugliche Stilregeln. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 176-189.
- [GL06] Gruhn, V. & Laue, R.: Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 69-84.
- [GL07] Gruhn, V. & Laue, R.: Forderungen an hierarchische EPK-Schemata. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 59-76.
- [Gr08] Gruhn, V., Laue, R., Kern, H. & Kühne, S.: EPK-Validierung zur Modellierungszeit in der bflow* Toolbox. In: (Loos, P., Nuttgens, M., Turowski, K. & Werth, D., Hrsg.): MobIS 2008 - Modellierung betrieblicher Informationssysteme – Modellierung zwischen SOA und Compliance Management. Saarbrücken, Germany 2008, S. 181-194.
- [GLM06] Gruhn, V., Laue, R. & Meyer, F.: Berechnung von Komplexitätsmetriken für ereignisgesteuerte Prozessketten. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 189-202.
- [HOS05] van Hee, K., Oanea, O. & Sidorova, N.: Colored Petri nets to verify extended event-driven process chains. In: Lecture Notes in Computer Science, vol. 3760. Springer, Berlin et al. 2005, S. 183-201.
- [HMS07] Heine, B.-O., Meyer, M. & Strangfeld, O.: Das Konzept der stilisierten Fakten zur Messung und Bewertung wissenschaftlichen Fortschritts. In: Die Betriebswirtschaft, 67 (2007) 5, S. 583-601.
- [He04] Hevner, A. R., March, S. T., Jinsoo, P. & Ram, S.: Design Science in Information Systems Research. In: MIS Quarterly, 28 (2004) 1, S. 75-105.
- [KK05] Kahl, T. & Kupsch, F.: Transformation und Mapping von Prozessmodellen in verteilten Umgebungen mit der EPK. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 54-73.
- [Ka61] Kaldor, N.: Capital Accumulation and Economic Growth. In: (Lutz, F. A. & Hague, D. C., Hrsg.): The Theory of Capital, Proceedings of a Conference Held by the International Economic Association. London 1961.
- [KNS92] Keller, G., Nuttgens, M. & Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten" (EPK). In: (Scheer, A.-W., Hrsg.) *Veröffentlichungen des Instituts für Wirtschaftsinformatik*. Universität des Saarlandes, Saarbrücken 1992.
- [KK07] Kern, H. & Kühne, S.: Forderungen an hierarchische EPK-Schemata. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 97-109.
- [Ki03] Kindler, E.: On the semantics of EPCs: A framework for resolving the vicious circle. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 7-18.

- [Ko07] Kopp, O., Eberle, H., Unger, T. & Leymann, F.: From Process Models to Business Landscapes. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 7-21.
- [KUL06] Kopp, O., Unger, T. & Leymann, F.: Nautilus Event-driven Process Chains: Syntax, Semantics, and their Mapping to BPEL. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 85-104.
- [Kr09] Krcmar, H.: Innovationen als Voraussetzung für Grundlagenforschung in der Wirtschaftsinformatik. In: *Wirtschaftsinformatik*, 51 (2009) 2, S. 224-225.
- [Kr08] Kruczynski, K.: Prozessmodellierung im Wettbewerb: EPK vs. BPMN. In: *is report*, 6 (2008) S. 30-35.
- [KDW08] Krumnow, S., Decker, G. & Weske, M.: Modellierung von EPKs im Web mit Oryx. In: (Loos, P., Nüttgens, M., Turowski, K. & Werth, D., Hrsg.): Proceedings of the Workshops collocated with the MobIS2008 Conference: Including EPK2008, KobAS2008 and ModKollGP2008. Saarbrücken, Germany 2008, S. 5-17.
- [LT04] Lay, G. & Tegtmeier, S.: Analyse und Visualisierung industrieller Serviceprozesse. In: *Industrie Management*, 1 (2004) S. 40-43.
- [LK06] List, B. & Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages. In: ACM symposium on Applied computing Dijon, France 2006, S. 1532-1539.
- [LF01] Loos, P. & Fettke, P.: Towards an Integration of Business Process Modeling and Object-Oriented Software Development. In: Proceedings of the Fifth International Symposium on Economic Informatics Bucharest 2001, S. 835-843.
- [Lü06] Lübke, D.: Transformation of Use Cases to EPC Models. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 137-156.
- [LMS05] Lübke, D., Marx Gómez, J. & Schneider, K.: Serviceorientierte Architekturen und Prozessmanagement. In: *ERP Management*, 3 (2005) S. 19-22.
- [Me03] Mendling, J.: Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen, Konzeption und Anwendung eines XMLSchemas für Ereignisgesteuerte Prozessketten (EPK). In: (Höpfner, H. & Saake, G., Hrsg.): Beitragsband zum Studierenden- Programm bei der 10. Fachtagung "Datenbanksysteme für Business, Technologie und Web", Fakultät für Informatik. Universität Magdeburg 2003, S. 48-50.
- [MBN04] Mendling, J., Brabenetz, A. & Neumann, G.: Generating SVG Graphics from EPML Processes. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2004 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Luxemburg 2004, S. 55-64.
- [MN02] Mendling, J. & Nüttgens, M.: Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen zur Definition eines XML-Schemas für Ereignisgesteuerte Prozessketten (EPK). In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 87-93.
- [MN03a] Mendling, J. & Nüttgens, M.: EPC Modelling based on Implicit Arc Types. In: (Godlevsky, M., Liddle, S. W. & Mayr, H. C., Hrsg.): Information Systems Technology and its Applications, International Conference ISTA'2003. Kharkiv, Ukraine 2003, S. 131-142.
- [MN03b] Mendling, J. & Nüttgens, M.: EPC Syntax Validation with XML Schema Languages. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 19-30.

- [MN03c] Mendling, J. & Nüttgens, M.: Konzeption eines XML-basierten Austauschformates für Ereignisgesteuerte Prozessketten (EPK). In: Informationssystem Architekturen, Wirtschaftsinformatik Rundbrief der GI Fachgruppe WI-MobIS 10 (2003) 2. 2003, S. 89-103.
- [MN03d] Mendling, J. & Nüttgens, M.: XML-basierte Geschäftsprozessmodellierung. In: (Uhr, W., Schoop, E. & Esswein, W., Hrsg.): Proceedings der 6. Internationalen Tagung Wirtschaftsinformatik 2003: Medien - Märkte Mobilität. Bd. Band II, Dresden 2003, S. 161-180.
- [MN04] Mendling, J. & Nüttgens, M.: Transformation of ARIS Markup Language to EPML. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2004 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Luxemburg 2004, S. 27-38.
- [MA06] Mendling, J. & van der Aalst, W. M. P.: Towards EPC Semantics based on State and Context. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 25-48.
- [MDA07] Mendling, J., van Dongen, B. F. & van der Aalst, W. M. P.: On the Degree of Behavioral Similarity between Business Process Models. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 39-57.
- [Me08] Mendling, J., Verbeek, H. M. W., van Dongen, B. F., van der Aalst, W. M. P. & Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model. In: Data & Knowledge Engineering, 64 (2008) 1, S. 312-329.
- [MZ05] Mendling, J. & Ziemann, J.: EPC Verification in the ARIS for MySAP reference model database. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 41-53.
- [MR00] Moldt, D. & Rodenhagen, J.: Ereignisgesteuerte Prozessketten und Petrinetze zur Modellierung von Workflows. In: (Giese, H. & Philippi, S., Hrsg.): Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme, Proceedings des 8. Workshop des Arbeitskreises "Grundlagen objektorientierter Modellierung" (GROOM) der GI-Fachgruppe 2.1.9 ("Objektorientierte Softwareentwicklung"), Bericht Nr. 24/00-I. Münster 2000, S. 57-63.
- [NC04] Neiger, D. & Churilov, L.: Goal-oriented business process modeling with EPCs and value-focused thinking. In: Business Process Management, 3080 (2004) S. 98-115.
- [NR02] Nüttgens, M. & Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: (Desel, J. & Weske, M., Hrsg.): Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens. Potsdam 2002, S. 64-77.
- [Pe08] Petsch, M., Schorcht, H., Nissen, V. & Himmelreich, K.: Ein Transformationsmodell zur Überführung von Prozessmodellen in eine Simulationsumgebung. In: (Loos, P., Nüttgens, M., Turowski, K. & Werth, D., Hrsg.): MobIS 2008 - Modellierung betrieblicher Informationssysteme – Modellierung zwischen SOA und Compliance Management. Saarbrücken, Germany 2008, S. 209-220.
- [PI04] Plümicke, M.: ARIS meets RUP - The ARIS Unified Information System Development Process In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2004 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Luxemburg 2004, S. 65-86.

- [PSW08] Polyvyanyy, A., Smirnov, S. & Weske, M.: Reducing Complexity of Large EPCs. In: (Loos, P., Nüttgens, M., Turowski, K. & Werth, D., Hrsg.): *MobIS 2008 - Modellierung betrieblicher Informationssysteme – Modellierung zwischen SOA und Compliance Management*. Saarbrücken, Germany 2008, S. 195-207.
- [Po94] Popper, K.: *Logik der Forschung*. Mohr, Tübingen 1994.
- [RD07] 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: *18th Australasian Conference on Information Systems*. Toowoomba, Australia 2007, S. 356-366.
- [Re09] Recker, J., Rosemann, M., Indulska, M. & Green, P.: Business Process Modeling - A Comparative Analysis. In: *Journal of the Association for Information Systems*, 10 (2009) 4, S. 333-363.
- [Ri00a] Rittgen, P.: EMC - A Modeling Method for Developing Web-based Applications. In: *International Conference of the International Resources Management Association (IRMA) 2000*. Anchorage, Alaska, USA 2000, S. 135-140.
- [Ri00b] Rittgen, P.: Paving the Road to Business Process Automation. In: *European Conference on Information Systems (ECIS) 2000*. Vienna, Austria 2000, S. 313-319.
- [Ri00c] Rittgen, P.: Quo vadis EPK in ARIS? Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik. In: *Wirtschaftsinformatik*, 42 (2000) 1, S. 27-35.
- [Ro02] Rodenhagen, J.: Ereignisgesteuerte Prozessketten (EPK) - Multiinstanziierungsfähigkeit und referentielle Persistenz. In: (Nüttgens, M. & Rump, F. J., Hrsg.): *EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 95-107.
- [SDL05] Sarshar, K., Dominitzki, P. & Loos, P.: Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne. In: (Nüttgens, M. & Rump, F. J., Hrsg.): *EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 97-116.
- [SL05] Sarshar, K. & Loos, P.: Modellierung überbetrieblicher Behandlungsprozesse durch Objekt-Petrinetze. In: *Wirtschaftsinformatik*, 47 (2005) 3, S. 203-210.
- [ST05] Scheer, A.-W. & Thomas, O.: Geschäftsprozessmodellierung mit der ereignisgesteuerten Prozesskette. In: *Das Wirtschaftsstudium*, 34 (2005) 8-9, S. 1069-1078.
- [STA05] Scheer, A.-W., Thomas, O. & Adam, O.: Process Modeling Using Event-driven Process Chains. In: (Dumas, M., van der Aalst, W. M. P. & ter Hofstede, A. H. M., Hrsg.): *Process-Aware Information Systems : Bridging People and Software Through Process Technology*. Wiley, Hoboken, New Jersey 2005, S. 119-145.
- [SFG09] Schmidt, W., Fleischmann, A. & Gilbert, O.: Subjektorientiertes Geschäftsprozessmanagement. In: *HMD- Praxis der Wirtschaftsinformatik*, 266 (2009) S. 52-62.
- [SS03] Schneider, K. & Schreiner, P.: Ein methodischer Ansatz zur Messung der Kundenintegration in der Dienstleistungserbringung. In: *Information Management & Consulting*, 18 (2003) 3, S. 51-57.
- [ST03] Schneider, K. & Thomas, O.: Kundenorientierte Dienstleistungsmodellierung mit Ereignisgesteuerten Prozessketten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): *EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, Proceedings des GI-Workshops und Arbeitskreistreffens. Bamberg 2003, S. 87-93.
- [Sc06] Schurz, G.: *Einführung in die Wissenschaftstheorie*. WBG, Darmstadt 2006.
- [Sc01] Schwerin, J.: *Wachstumodynamik in Transformationsökonomien - Strukturähnlichkeiten seit der Industriellen Revolution und ihre Bedeutung für Theorie und Politik*. Böhlau, Köln et al. 2001.

- [SV05] Seel, C. & Vanderhaeghen, D.: Meta-Model based Extensions of the EPC for Inter-Organisational Process Modelling. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 117-136.
- [SS07] Seidlmeier, H. & Scherfler, G.: Modellgetriebene Integration und Migration - vom Fachprozess zur ausführbaren Anwendung. In: HMD- Praxis der Wirtschaftsinformatik, 257 (2007) S. 93-105.
- [SFO06] Simon, C., Freiheit, J. & Olbrich, S.: Using BPEL Processes defined by Event-driven Process Chains. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 121-136.
- [St00] Serman, J. D.: Business Dynamics - System Thinking and Modeling for a Complex World. McGraw-Hill, Boston et al., USA 2000.
- [St06] Störrle, H.: A Comparison of (e)EPCs and UML 2 Activity Diagrams. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 177-188.
- [TD06] Thomas, O. & Dollmann, T.: Fuzzy-EPK-Modelle: Attributierung und Regelintegration. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 49-68.
- [TF06] Thomas, O. & Fellmann, M.: Semantische Integration von Ontologien und Ereignisgesteuerten Prozessketten. In: (Nüttgens, M., Rump, F. J. & Mendling, J., Hrsg.): EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Wien 2006, S. 7-24.
- [THA02] Thomas, O., Hüsselmann, C. & Adam, O.: Fuzzy-Ereignisgesteuerte Prozessketten - Geschäftsprozessmodellierung unter Berücksichtigung unscharfer Daten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Trier 2002, S. 7-16.
- [TKL05] Thomas, O., Kaffai, B. & Loos, P.: Referenzmodellbasiertes Event-Management mit Ereignisgesteuerten Prozessketten. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens Hamburg 2005, S. 74-96.
- [Th04] Thomas, O., Seel, C., Seel, C., Kaffai, B. & Martin, G.: EPK-Referenzmodelle für Verwaltungsverfahren. In: (Nüttgens, M. & Rump, F. J., Hrsg.): EPK 2004 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. Luxemburg 2004, S. 39-54.
- [We07] Wehler, J.: Boolean and free-choice semantics of Event-driven Process Chains. In: (Nüttgens, M., Rump, F. J. & Gadatsch, A., Hrsg.): EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens. St. Augustin 2007, S. 77-96.
- [WL07] Weißenberger, B. E. & Löhr, B.: Planung und Unternehmenserfolg: Stylized Facts aus der empirischen Controllingforschung im deutschsprachigen Raum von 1990-2007. In: Zeitschrift für Planung & Unternehmenssteuerung, 18 (2007) 4, S. 335-363.
- [Wi09] Winter, R.: Was ist eigentlich Grundlagenforschung in der Wirtschaftsinformatik? In: Wirtschaftsinformatik, 51 (2009) 2, S. 223-224.
- [Ze09] Zelewski, S.: Was ist eigentlich Grundlagenforschung in der Wirtschaftsinformatik? In: Wirtschaftsinformatik, 51 (2009) 2, S. 227-229.

On Labeling Quality in Business Process Models

Henrik Leopold[†], Sergey Smirnov[‡], Jan Mendling[†]

[†]Humboldt-Universität zu Berlin, Germany

henrik.leopold@student.hu-berlin.de|jan.mendling@wiwi.hu-berlin.de

[‡]Hasso Plattner Institute, Potsdam, Germany

sergey.smirnov@hpi.uni-potsdam.de

Abstract: Quality assurance is a serious issue for large-scale process modeling initiatives. While formal control flow analysis has been extensively studied in prior research, there is a little work on how the textual content of a process model and its activity labels can be systematically analyzed. It is a major challenge to classify labels according to their quality and consequently assure high label quality. As a starting point we take a recent research on the activity labeling style, which establishes superiority of a so-called verb-object labeling style. Together with the labeling style, the length of an activity label is related to its quality.

In this paper, we investigate how various natural language processing techniques, e.g., part of speech tagging and analysis of phrase grammatical structure, can be used to detect an activity labeling style in an automatic fashion. We also study how ontologies, like WordNet, can support the solution. We conduct a thorough evaluation of the developed techniques utilizing about 20,000 activity labels from the SAP Reference Model.

1 Introduction

Modeling of business processes in large enterprises usually implies a team work of numerous specialists. Such teams may span several organizational departments and even several geographical locations. The staff who takes part in the projects can have different professional background. Hence, organization of efficient work in these teams and assuring an appropriate quality of the produced models become real challenges. This situation has motivated researchers and practitioners in business process management to discuss various aspects of model quality [GL07]. First, there are techniques and tools which assure formal properties like behavioral soundness of process models [Aal97]. Second, there are works on how process model characteristics affect model comprehension by human model readers. The importance of this aspect is motivated by the fact that most process models are created for documentation purposes [DGR⁺06]. It has been shown that large and complex models are more likely to contain errors and are less understandable.

Relatively small attention has been paid to the problem of labeling quality. For instance, a study in [MRR09] showed that the current labeling practice is conducted rather arbitrarily. Meanwhile, labels are the key to understanding the process models by humans. The significance of label quality can be motivated by the dual coding theory [Pai69]. The theory

states that humans grasp information more easily, if it is provided via the auditory and the visual channels. In the context of process modeling the visual channel is represented by the graphical constructs of a modeling language, while the auditory channel—by textual model element labels. As process models use only a few graphical constructs, the use of informative and unambiguous labels improves an overall understanding of a process model. In this way, labels contribute to semantic and pragmatic usefulness of a model (see [KSJ06]).

As real world process model repositories can easily include thousands of process descriptions [Ros06], an efficient quality assurance mechanism has to rely on an automatic classification of models according to a specific quality aspect. In terms of label quality the challenge is to identify the labels of poor quality. From the human user perspective, the designer can be supported by a modeling tool identifying poor labels. This functionality can be extended further to giving suggestions on label reengineering.

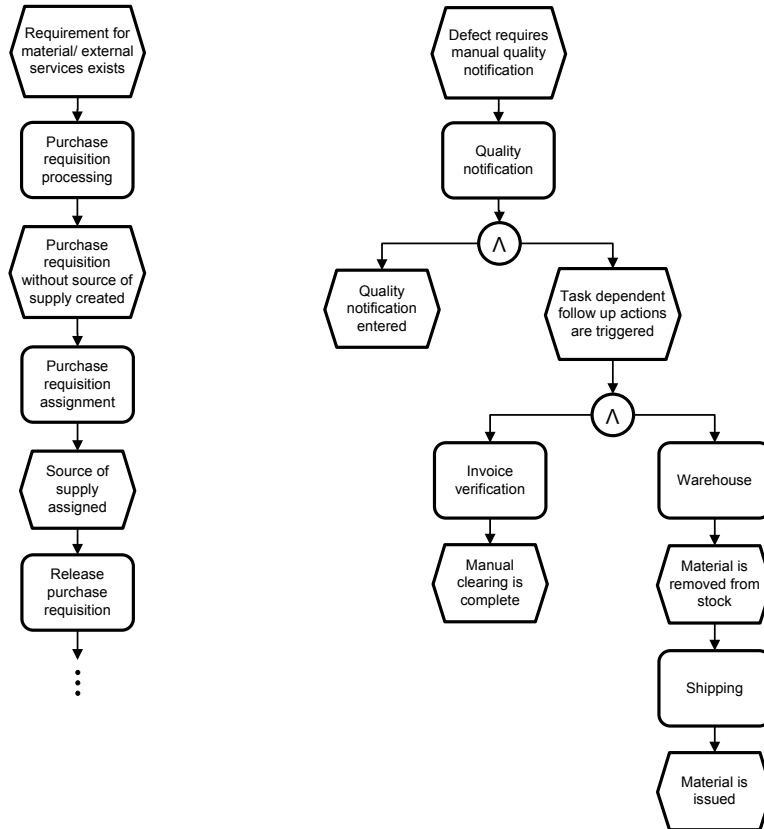
The quality of labels in process models can be discussed along two orthogonal dimensions. On the one hand, appropriate and consistent terms have to be chosen, which is related to a thesaurus. On the other hand, these terms have to be composed according to a particular structure of labels, which can be related to grammatical styles. As the strong influence of label structure on model understanding has been shown in [MRR09], in this work we study the structure of labels. Furthermore, we narrow the scope of our research to the investigation of activity labels. Although a model semantics is defined by the whole constellation of the model elements, activities accumulate its largest part.

The contribution of this paper is the specification and validation of two criteria for assessment of quality of activity labels in process models: frequency of activity labeling style and activity labels length. We propose two metrics on the basis of these criteria and present an automatic method for their quantitative evaluation. While quantitative evaluation of labels length is straightforward, frequency of labeling style requires sophisticated analysis of labels. Furthermore, we perform an evaluation of the SAP Reference Model using these metrics. The SAP Reference Model is a publicly available business process model collection, capturing the processes using Event-driven Process Chains (EPCs) [KT98].

The structure of the paper is as follows. Section 2 motivates this work using the examples of real-world process models. Section 2 discusses what labeling styles can be found in practice and how they correlate with quality and process model understandability. Section 3 introduces the quality metrics for measuring different aspects of structural quality. Subsequently, Section 4 discusses the implementation approach for computing one structural metric automatically. Section 5 presents the results of an evaluation involving the SAP Reference Model. Section 6 discusses the findings in its relationship to the related work. Section 7 concludes the paper and gives an outlook on future research.

2 Labeling Styles

In this section we discuss different styles of labeling. Section 2.1 discusses labeling in the SAP Reference Model, and Section 2.2 the classification of labels. Section 2.3 highlights



(a) Fragment of process model *Purchase Requisition*

(b) Model of process *Return Deliveries*

Figure 1: Labeling in two business process model examples from the SAP Reference Model

implications.

2.1 Labeling in the SAP Reference Model

Fig. 1 shows two examples of business process models from the SAP Reference Model, a process model collection that has been used in several works on process model analysis. Fig. 1(a) depicts a fragment of a business process where a *purchase requisition* is handled. Within this model fragment we observe activity labels *purchase requisition processing*, *purchase requisition assignment* and *release purchase requisition*. In the first two labels the actions are denoted with the nouns *processing* and *assignment*. In the third one the verb *release* corresponds to the action. Obviously, the modelers used several styles for activity labeling. Ambiguity is a potential threat to label understanding. For instance, consider the

word *purchase*, which can be both a noun and a verb. This source of ambiguity is called zero derivation, since a verb is linguistically created from a noun without adding a postfix like *-ize* in *computerize*. It has been pointed out that different styles are prone to different degrees of ambiguity [MRR09], which emphasizes the importance of labeling styles for human understanding. If an action noun is used, there is likely an ambiguity, when it is combined with a zero derivation noun. If we consider the *purchase requisition processing* label, it is hard to tell if *purchase* or *processing* stands for an action. As zero derivation is an essential part of the language that cannot always be avoided, it is a useful strategy to employ and enforce a suitable labeling style.

Fig. 1(b) highlights further potential problems of labels. It captures a model of the *return deliveries* business process. In this model, we can observe activity labels that do not signify any action (e.g. *warehouse*) or activities with actions, but without any object (e.g. *shipping*). Again understanding of such activities requires the reader to interpret the context of the model.

2.2 Classes of Labeling Styles

In order to assess the structural label quality of a process model, it is essential to distinguish different labeling styles, which are found in practice. An analysis of the SAP Reference Model was conducted to identify such different styles and their relationship to potential causes of ambiguity [MRR09]. The activity labels are classified into three major structural categories: verb-object style, action-noun style, and the rest category. Table 1 provides categorization of labels used in models in Fig. 1.

The classification approach is based on the grammatical representation of the action in an activity label. For labels belonging to the verb-object style, the action is captured as a verb and used in the imperative form of the verb at the beginning of the label. Examples are labels like *enter count results* or *compare value dates*. Although the name verb-object style suggests the necessity of an object, also labels like *process* or *follow-up* are subsumed to the verb-object category. In a label following the action-noun style the activity is described in terms of a noun, as in *printing notification* or *check of order*. These nouns reflecting the action are either derived from a verb like in *analysis* or are a gerund like *analyzing*. All remaining labels, referred as the rest category, do not contain a word from which an action can be inferred. This applies to labels like *basic settings*.

A deeper analysis of these labeling styles reveals that all of them are prone to specific types of ambiguity. For example, verb-object labels could potentially be misinterpreted if they are affected by a zero derivation ambiguity. This is the case when a word can be both a verb and a noun without adding any suffixes. An example is *measure* in the label *measure processing*. Thus, this label could either refer to the measurement of a processing or the processing of a measure. But also action-noun labels are affected by ambiguity, which is referred as action-object ambiguity. The label *printing notification* emphasizes this problem, since it could either advice to print a notification or to notify somebody to conduct a printing job. As rest labels do not contain a word representing an action, they

Table 1: Activity labels used in the models in Fig. 1

Activity Label	Labeling Style	Action	Object
Release purchase requisition	verb-object	release	purchase requisition
Purchase requisition processing	action-noun	process	purchase requisition
Purchase requisition assignment	action-noun	assign	purchase requisition
Quality notification	action-noun	notify	quality
Invoice verification	action-noun	verify	invoice
Warehouse	rest		
Shipping	rest		

suffer from the verb-inference ambiguity. This means, that a reader of the process model may not be able to evaluate what to perform when reading a label of the rest category.

2.3 Implications for Labeling Practices in Process Models

Knowing the different styles and their ambiguities, the resulting question is which of these styles should be preferred in practice. Several guidelines for conceptual modeling propose to follow a verb-noun convention in which an action is grammatically captured as a verb [KDV02]. This suggestion is also supported in [MRR09]. On the one hand, the authors counted the ambiguity cases in the SAP Reference Model. They uncovered that among verb-object labels 5.1% and among action-noun labels 9% were affected by ambiguity. Due to the verb-inference ambiguity all labels from the rest category were considered as ambiguous. On the other hand, the same research group conducted an experiment studying the impact of labeling styles on perceived ambiguity. A group of students was asked to assess labels in a given process model regarding their ambiguity and their provided usefulness. The result was that verb-object labels were considered to be the least ambiguous and the most useful in comparison to labels of the other styles. Congruent with the frequency of the ambiguity cases, the rest labels were regarded to be most ambiguous and having the lowest usefulness.

Based on these findings, it can be stated that labeling activities according to the verb-object style is the most desirable from a quality point of view. While the authors of [MRR09] manually inspected the labels, this is hardly an option in industry practice. The SAP Reference Model with its 600 process models already contains about 20,000 activity labels, and it is still much smaller than other model collections with several thousand models. Therefore, we need to investigate how label quality in terms of compliance to a particular labeling style can be determined automatically.

3 Metrics for Measuring Structural Label Quality

In this section we present the metrics for measuring the model quality based on the properties of activity labels. We distinguish two main metrics groups: those considering the length of activity labels and those considering the style.

3.1 Metrics Based on the Length of Activity Labels

An activity label has to capture the essential information about the activity on one hand, but should not overload the reader with unnecessary information on the other hand. Among several other means to achieve this, a label length is a natural regulator for the amount of information in the label. In [MS08] the authors investigated the relation between the label length and label understandability. The study has shown that shorter labels facilitate proper understanding of a process model. However, the terms *short* or *long* are relative and imprecise. In the context of this work we need concrete numbers to tell *short* from *long*.

In [Fle51] Flesch argues that the understandability of a text with sentences of length eight or fewer words can be claimed as *very easy*. The author reports that sentences of length 14 still can be understood *fairly easy*. We adapt these findings to the problem of activity labels and claim that activity labels with the length of eight or less words are easy to interpret. We call the labels with the length greater than eight words labels with excess length. Hence, one metric is the fraction of activities which have labels of excess length. We refer to this metric as excess length fraction and denote it as L_e . The lower the value of excess length fraction, the higher is model understandability.

Although short labels improve model understandability, it is vindicable to state that a label should contain at least two words. In particular, a label should point to an action and an object on which the action is performed. Labels of one word length provide readers too small amount of information (see labels *shipping* and *warehouse* in Section 2). Considering this issue, we propose to treat the fraction of activity labels of one word length as a metric as well. We refer to this metric as one word fraction and denote it as L_o . The lower is L_o , the better is the model understandability.

Finally, a natural metric is the average length of activity labels in the model. We denote it with L_a . Again, the lower is L_a , the better is the model understandability. As a result, we distinguish the following three metrics based on the activity label length:

- excess length fraction L_e ,
- one word fraction L_o ,
- average label length L_a .

3.2 Metrics Based on the Frequency of Activity Labeling Styles

The second group of metrics considers fractions of activity labels adhering to one labeling style. In Section 2 we have identified three labeling styles for activities: verb-object, action-noun and rest. The fractions of activities with labels of particular labeling style result in three metrics for the quality of labels:

- verb-object style fraction S_{vo} ,
- action-noun style fraction S_{an} ,
- rest category fraction S_r .

Activities with labels of verb-object style are easy to comprehend for humans. Hence, high values of S_{vo} imply good model understandability. On the opposite, labels of the rest category are the source of high ambiguity. They harden model comprehension and witness of low model quality. As a result, the lower the value of S_r , the better the model understandability is. Thereafter, the fractions of activities with verb-object style and rest style provide sufficient information about labeling quality in a model. For instance, if the model has a large fraction of activities which adhere to verb-object style and small number of activities with rest style, the model has high labeling quality. At the same time, the number of labels with action-noun style depends on the fractions of activities with labels in verb-object and rest styles.

4 Automatic Identification of Labeling Style

In this section we describe different strategies towards an automatic identification of labeling styles. Every label is analyzed independently from others and from the structure of the process model containing the label. We focus on so-called part of speech tagging [JM08] as a tool to identify the grammatical form of the words of an activity label. Part of speech (or grammatical) tagging is a technique from computational linguistics that assigns the part of speech like verb, subject, or object to words of a text based on the syntactical form of the word and its context within a sentence.

Nowadays, there are powerful algorithms available to automatically perform part of speech tagging. The automatic determination of the grammatical structure of the labels is conducted with part of speech tagging tools. Such tools, referred as tagger or parser, are developed for natural language processing. They provide functionality to assign the according part of speech for each word in a given text. Thus, for instance the input string *process order* will lead to the tagging result *process/VB order/NN*. The tag *VB* indicates that the word *process* is a verb in the base form and the tag *NN* represents a singular noun. There exist about 50 different part of speech tags for providing differentiated information about a word [MMS93]. For example, it is possible to distinguish between different verb types. There are part of speech tags for verbs in the past tense (VBD or VBN), for gerunds (VBG) and for verbs in the third person form (VBZ).

In order to provide the process modeler with the best obtainable results, different part of speech tagging tools are considered regarding their accuracy. In this context, the following tools are further investigated: the Stanford Parser, the Stanford Tagger, and WordNet. The tagger and the parser from the Stanford University [TM00, KM03] both provide the functionality to assign part of speech tags to a given text. The Stanford Parser does additionally analyze the structure and relations between the words in each sentence and is thus able to use more information for its part of speech assessment. WordNet is a lexical database and provides different functionality for analyzing semantic relations between words [Mil95]. Amongst others, WordNet provides a function for determining the most likely part of speech for a single word without considering context.

The *Stanford Tagger* requires its input in terms of a text file, which can include all relevant activity labels separated with a “.” as punctuation mark. As a result the tagger returns a string where each word is enriched with the according part of speech tag. In initial experiments, we observed some problems with tagging activity labels. In the general case, the Stanford Tagger has been tested to work with an accuracy of about 97% [TM00, TKMS03]. Apparently, the structure of the majority of the labels seems to be inappropriate for it as they are not really natural language sentences. We have seen that about one third of all labels (action-noun style plus rest category) do not even contain a verb, such as for instance *asset maintenance* or *order processing*. Even if the labels are proper sentences, they tend to be rather short like *perform posting* or *edit classes*. All these factors might contribute to a poor tagging result.

In order to improve the results the original labels were extended with the prefix *You have to*. First, it increases the label length, and therefore the grammatical context. Second, it yields a proper English sentence for verb-object languages. For instance, the label *process order* is extended to *You have to process order*. While the word *process* plays still the role of a verb, it might be detected with a higher probability since there is now more information for the tagging tool which can be evaluated. Applying this approach, we observed a considerable increase in tagging performance.

The *Stanford parser* expects the input sentences wise. Hence, the parser has to be fed with all activity labels sequentially. Therefore, each label was extended with a “.” as punctuation mark. Experiments by the original authors have shown that the tagger tends to work with an accuracy of about 86% [KM03]. The parser considers the relations between the words. Therefore, the accuracy does not only depend on the part of speech tagging but also on the accuracy of the correct detection of grammatical relations. Accordingly, we expect it to provide a higher recall. However, some initial experiments pointed to some weaknesses. These might stem from the fact that activity labels are not really sentences, which might cause problems.

Beyond the mentioned tagger and parser, we considered the lexical database *WordNet* [Mil95] via its corresponding Java implementation called *Rita*. WordNet provides a function called *getBestPos* that returns the best part of speech for a given word based on its polysemy count. This means that the function will return the part of speech for a given word that captures the most different senses. For instance, *getBestPos* will return *verb* for a word having 8 different senses as a verb and only 6 meanings as a noun. Consequently, WordNet was provided with the first word of each label, as verb-object labels would start

with a verb, and we observed quite good performance in initial experiments. We consider a further improvement by combining WordNet with the tagging approach using the extended labels with the prefix. But as the combination of both requires the union of both result sets, also the amount of incorrectly detected labels increases in the combined result set. This implies a decreasing precision value.

5 Evaluation of the SAP Reference Model

In this section we present an evaluation of a real world collection of business process models against the proposed labeling quality metrics. First we introduce the collection, describing its properties relevant for the experiment. Further, we present the evaluation results.

5.1 SAP Reference Model

The experiment studies the SAP Reference Model [KT98], a process model collection that has been used in several works on process model analysis [Men08]. The collection captures business processes that are supported by the SAP R/3 software in its version from the year 2000. It is organized in 29 functional branches of an enterprise, like sales or accounting, that are covered by the SAP software. The SAP Reference Model includes 604 EPCs.

5.2 Metrics Overview for the SAP Reference Model

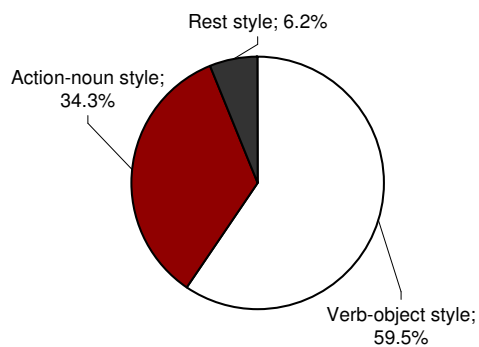


Figure 2: Distribution of activity labeling styles in the SAP Reference Model

Let us first discuss those metrics that are based on the labeling styles. [MRR09] performed an analysis of activity labeling styles employed in the SAP Reference Model. The 19,839 activity labels of the model collection were manually inspected in order to reveal frequencies of labeling styles. About 60% of labels follow the verb-object style, 34% were classified as action-noun labels and only about 6% of the labels belong to the rest category (see Fig. 2). This distribution is quite favorable as a majority of 94% of labels refer an action, while two thirds are verb-object labels. Nevertheless, there are still 6% of all labels which definitely suffer from the verb-inference ambiguity and

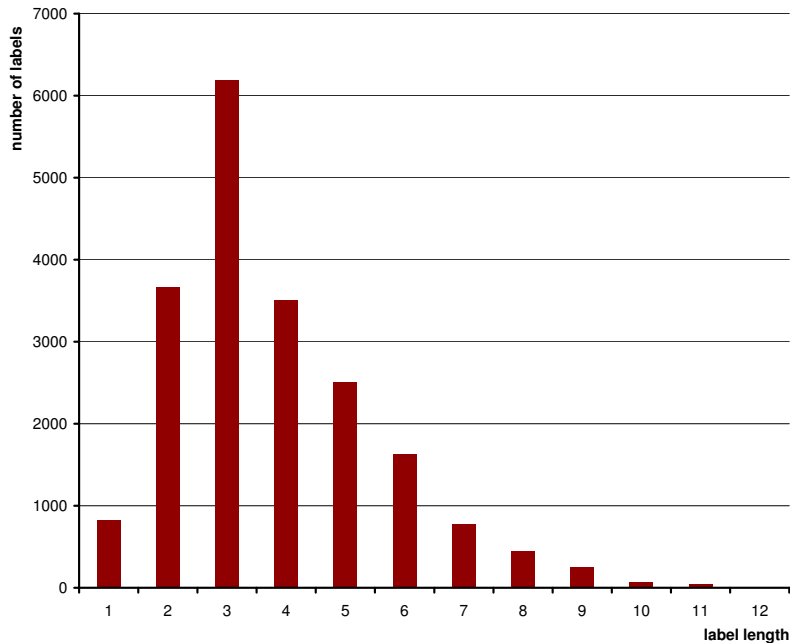


Figure 3: Label length distribution in the SAP Reference Model

might cause misinterpretations.

Figure 3 depicts the length distribution showing that most labels have length of only three words and that there is no label having more than 12 words. The average label length L_a equals to 3.78. Hence, according to [Fle51], most labels can be pronounced *fairly easy* to understand and even *very easy* to understand. The excess length fraction value is $L_e = 3.98\%$. Meanwhile, one word fraction L_o is 4.12%. This fact is important, as these activity labels do not point either to an action or an object, decreasing model label quality.

Summarizing the findings on the SAP Reference Model, the majority of the labels are action orientated or even follow the verb-object style. Moreover, the average label length is below 4 and is thus very short which also supports comprehension. Less favorable are those labels containing only one word or no word referring to an action since those will most likely cause misunderstandings. Table 2 summarizes all introduced metrics for the SAP Reference Model.

5.3 Automatic Part of Speech Identification Results

In this section, we analyze the accuracy of the different strategies to automatic part of speech identification. The manual classification of labels of [MRR09] serves us as a bench-

Table 2: Calculated Metrics for the SAP Reference Model

Metric	Value
Excess Length Fraction (%)	3.98
One Word Label Fraction (%)	4.12
Average Label Length	3.78
Verb-Object-Style Fraction (%)	60
Action-Noun-Style Fraction (%)	34
Rest-Style Fraction (%)	6

mark. We use standard precision (ratio of *found relevant labels* to *all found labels*) and recall (ratio of *found relevant labels* to *all relevant labels*) measurements to assess accuracy.

Figure 4 depicts the results of the different part of speech tagging techniques. It can be seen that the Stanford Parser achieved the lowest recall values. This rather weak performance is likely to be caused by its dependence on accurate contextual information, which is hardly available in short activity labels. The Stanford Tagger showed better results with a recall of about 51%. It was considerably improved by using the *You have to* prefix, which extends verb-object labels to correct English sentences. The rather simple approach of using the probable part of speech function offered by WordNet worked surprisingly well. Combining it with the prefixed tagger yielded almost 99% recall.

Besides the already discussed recall values the Figure shows also the precision value for each approach. This precision value was optimized using two practices. First of all, only those labels were assigned to the verb-object category where the first word in the label has a verb tag indicating a base form. Since only the base form of a verb matches with the imperative form, this approach is reasonable. Thus, for instance labels like *determining protocol proposal/NNP* with a gerund tag *VBG* at the beginning and the label *fixed price billing* starting with a past tense tag *VBN* are excluded from the result set. As Wordnet only differentiates between verb, noun, adjective and adverb, this practice was not applicable for WordNet. By contrast, the second practice for improving the precision value was applicable for all techniques. This approach aims for excluding labels that have a base form verb at the beginning but still do not belong to the verb-object category. Examples are the labels *check of order* or *release of process order*. Apparently, the first word of each label suffers from the zero-derivation ambiguity since considered in isolation they can both be verb and noun. But the preposition *of* uncovers that the first word simply cannot play the role of verb. Therefore, labels with the sequence *verb + of* are also excluded from the result set. These practices excluded up to 5% of all detected labels.

The results suggest two conclusions. If recall and precision are considered in combination, the tagging approach using the prefixes for extending the labels might be an option for an implementation approach. But still, WordNet obtained the best results. Thus, the WordNet approach should be preferred. Although, the combination with the tagger resulted in slightly increased results, the combination is not worthwhile for two reasons. First of all, the precision value decreases. Secondly, the effort for the additional tagging is not reflected in a significant increase of the results.

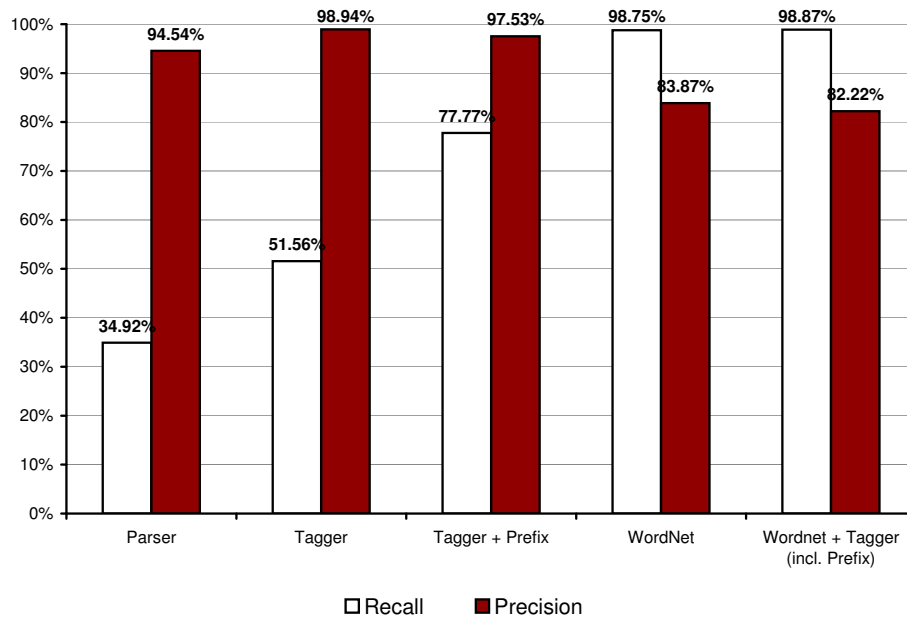


Figure 4: Result of the Analysis on Part of Speech Tagging Techniques

6 Related Work

Our work can be related to three major streams of related work: quality frameworks, process model labeling, and natural language approaches for models.

Process model quality is discussed in different works on *quality frameworks*. The SEQUAL framework builds on semiotic theory and defines several quality aspects [LSS94, KSJ06]. In essence, syntactic quality relates to model and modeling language, semantic quality to model, domain, and knowledge, and pragmatic quality relates to model and modeling and its ability to enable learning and action. The semantic and pragmatic quality clearly point to the relevance of labeling activities. The Guidelines of Modeling (GoM) define an alternative quality framework that is inspired by general accounting principles [BRU00]. The guidelines include the six principles of correctness, clarity, relevance, comparability, economic efficiency, and systematic design, where several of them have implications for good labeling. Also the ISO 9126 [ISO91] quality standard has been suggested as a starting point for model quality [Moo05, GD05].

The verb-object style is widely promoted in the literature for *labeling activities of process models* [Mil61, SM01, MCH03], but rather as informal guidelines. Similar conventions are advocated as guidelines for the creation of understandable use case descriptions, a widely accepted requirements tool in object-oriented software engineering [PVC07]. But

in contrast to its promotion in the process modeling domain, it has been observed that verb-object labeling in real process models is not consistently applied. For instance, the practical guide for process modeling with ARIS [Dav01, pp.66-70] shows models with both actions as verbs and as nouns. It has also been shown that shorter activity labels improve model understanding [MS08], which is consistent with readability assessments on sentence length [Gre00, Fle51]. The concept of part of speech tagging is also investigated for interactive process modeling support. In a recent paper, the authors employ it for auto-completion [BDH⁺09].

The enforcement of the verb-object style might help to close the gap between *natural language and formal language processing*. And indeed, the relationship between process models and natural language has been discussed and utilized in various works. In [FKM05] the authors investigate in how far the three steps of building a conceptual model (linguistic analysis, component mapping, and schema construction) can be automated using a model for pre-design. Further text analysis approaches have been used to link activities in process models to document fragments [IGSR05] and to compare process models from a semantic perspective [EKO07]. Most beneficiary is the verb-object style for model verbalization and paraphrasing, see [HC06, FW06]. Such verbalization is an important step in model and requirements validation [NE00]. For instance, verb-object style labels can easily be verbalized using the *You have to* prefix, which we also used in our analysis. In this way, automatic parsing enables a better validation of process models.

7 Conclusion and Future Work

Recent research revealed the high impact of the labeling quality in business process models on the overall model understanding. These findings motivated us to the work reported in this paper. First, we introduced several metrics that help to assess the quality of activity labels in the model. The metrics are categorized into two groups: metrics based on the activity label length and metrics based on the activity labeling style. Among the proposed metrics, we focused on verb-object fraction, since it is crucial for assessment of model labeling quality. We developed and implemented an approach enabling evaluation of verb-object fraction metrics. For evaluation of the techniques we proposed, we use the SAP Reference Model. The results show that high precision and recall can be achieved automatically by using part of speech tagging techniques and available tools.

In this paper we focused only on the labeling quality of activities in business process models. However, there are other model elements like events and data objects that should be subject to label quality assurance. It is part of our future research agenda to identify how part of speech tagging techniques can be applied for those labels as well. Our results also depend on using English as a language for labeling activities. It will be an interesting task to analyze other languages, like German or Russian, to see whether part of speech tagging can be utilized with the same accuracy. Finally, we plan to use tagging information for building taxonomies for process model collections. Identifying nouns from the label will be a crucial step for this application.

References

- [Aal97] W.M.P. van der Aalst. Verification of Workflow Nets. In Pierre Azéma and Gianfranco Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer Verlag, 1997.
- [BDH⁺09] Jörg Becker, Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. page (forthcoming), 2009.
- [BRU00] J. Becker, M. Rosemann, and C. von Uthmann. Guidelines of Business Process Modeling. In W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors, *Business Process Management. Models, Techniques, and Empirical Studies*, pages 30–49. Springer, Berlin et al., 2000.
- [Dav01] R. Davis. *Business Process Modelling With Aris: A Practical Guide*. Springer, 2001.
- [DGR⁺06] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3):358–380, 2006.
- [EKO07] M. Ehrig, A. Koschmider, and A. Oberweis. Measuring Similarity between Semantic Business Process Models. In J.F. Roddick and A. Hinze, editors, *Conceptual Modelling 2007, Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM 2007)*, volume 67, pages 71–80, Ballarat, Victoria, Australia, 2007. Australian Computer Science Communications.
- [FKM05] G. Fliedl, C. Kop, and H.C. Mayr. From textual scenarios to a conceptual schema. *Data and Knowledge Engineering*, 55(1):20–37, 2005.
- [Fle51] R. Flesch. *How to Test Readability*. Harper & Brothers, New York, NY, USA, 1951.
- [FW06] P.J.M. Frederiks and T.P. van der Weide. Information Modeling: The Process and the Required Competencies of Its Participants. *Data & Knowledge Engineering*, 58(1):4–20, 2006.
- [GD05] A. Selçuk Güceglioglu and O. Demirörs. Using Software Quality Characteristics to Measure Business Process Quality. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science (LNCS)*, pages 374–379. Springer Verlag, 2005.
- [GL07] V. Gruhn and R. Laue. What Business Process Modelers Can Learn from Programmers. *Sci. Comput. Program.*, 65(1):4–13, 2007.
- [Gre00] H. Gretchen. Readability and Computer Documentation. *ACM J. Comput. Doc.*, 24(3):122–131, 2000.
- [HC06] T.A. Halpin and M. Curland. Automated Verbalization for ORM 2. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Montpellier, France, October 29 - November 3. Proceedings, Part II*, volume 4278 of *Lecture Notes in Computer Science*, pages 1181–1190. Springer, 2006.
- [IGSR05] J.E. Ingvaldsen, J.A. Gulla, X. Su, and H. Rønneberg. A Text Mining Approach to Integrating Business Process Models and Governing Documents. In R. Meersman et al., editor, *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA*,

MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005, Proceedings, volume 3762 of *Lecture Notes in Computer Science*, pages 473–484. Springer, 2005.

- [ISO91] International Standards Organisation ISO. Information Technology - Software Product Evaluation - Quality Characteristics and Guide Lines for their Use. Iso/iec is 9126, 1991.
- [JM08] D. Jurafsky and J.H. Martin. *Speech and language processing*. Prentice Hall, 2008.
- [KDV02] H. Koning, C. Dormann, and H. van Vliet. Practical Guidelines for the Readability of IT-architecture Diagrams. In *Proceedings of the 20th Annual International Conference on Documentation, ACM SIGDOC 2002*, pages 90–99, 2002.
- [KM03] D. Klein and Ch. D. Manning. Accurate Unlexicalized Parsing. *41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003.
- [KSJ06] J. Krogstie, G. Sindre, and H.D. Jørgensen. Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems*, 15(1):91–102, 2006.
- [KT98] G. Keller and T. Teufel. *SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping*. Addison-Wesley, 1998.
- [LSS94] O.I. Lindland, G. Sindre, and A. Sølvsberg. Understanding Quality in Conceptual Modeling. *IEEE Software*, 11(2):42–49, 1994.
- [MCH03] T.W. Malone, K. Crowston, and G.A. Herman, editors. *Organizing Business Knowledge: The MIT Process Handbook*. The MIT Press, 2003.
- [Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.
- [Mil61] L.D. Miles. *Techniques of value analysis and engineering*. McGraw-hill, 1961.
- [Mil95] G. A. Miller. WordNet: a Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.
- [MMS93] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 1993.
- [Moo05] D.L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.
- [MRR09] J. Mendling, H. A. Reijers, and J. C. Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems*, 2009.
- [MS08] J. Mendling and M. Strembeck. Influence Factors of Understanding Business Process Models. In W. Abramowicz and D. Fensel, editors, *Proc. of the 11th International Conference on Business Information Systems (BIS 2008)*, volume 7 of *Lecture Notes in Business Information Processing*, page 142–153. Springer-Verlag, 2008.
- [NE00] B. Nuseibeh and S.M. Easterbrook. Requirements engineering: a roadmap. pages 35–46, 2000.
- [Pai69] A. Paivio. Mental Imagery in Associative Learning and Memory. *Psychological Review*, 76:241–263, 1969.

- [PVC07] Keith Thomas Phalp, Jonathan Vincent, and Karl Cox. Improving the Quality of Use Case Descriptions: Empirical Assessment of Writing Guidelines. *Software Quality Journal*, 15(4):383–399, 2007.
- [Ros06] M. Rosemann. Potential pitfalls of process modeling: part A. *Business Process Management Journal*, 12(2):249–254, 2006.
- [SM01] A. Sharp and P. McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development*. Artech House Publishers, 2001.
- [TKMS03] K. Toutanova, D. Klein, Ch. D. Manning, and Y. Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *HLT-NAACL*, pages 252–259, 2003.
- [TM00] K. Toutanova and Ch. D. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *EMNLP*, pages 63–70, 2000.

Ein einfaches Verfahren zur Erkennung häufiger Fehler in EPKs

Volker Gruhn, Ralf Laue

{gruhn, laue}@eбус.informatik.uni-leipzig.de

Lehrstuhl für Angewandte Telematik und E-Business*

Universität Leipzig, Fakultät für Informatik

Abstract: In diesem Beitrag nutzen wir den in [GL06] eingeführten Ansatz, die in einer ereignisgesteuerten Prozesskette (EPK) enthaltenen Informationen in eine Prolog-Faktenbasis zu übersetzen und durch Abfragen an den Prolog-Interpreter mögliche Modellverbesserungen zu lokalisieren.

Dabei werden, anders als in früheren Veröffentlichungen beschrieben, die Beschriftungen von Ereignissen und Funktionen in die Analyse der Modelle einbezogen.

Durch Erzeugen einer textuellen Normalform und die Beachtung von Synonymen und Antonymen wird versucht, Beschriftungen mit gleicher Bedeutung und Beschriftungen, die sich widersprechen, zu finden. Auf diese Weise lassen sich bestimmte Klassen häufig anzutreffender Modellierungsfehler in EPKs automatisch erkennen.

Das Verfahren wurde an 1253 deutschsprachigen Modellen getestet. Dabei wurden mehrere Fehler identifiziert, die mit Hilfe herkömmlicher Validierungstechniken unentdeckt bleiben.

1 Einführung

Die Semantik eines Modells lässt sich nach Esswein et al. [EGS04] stets in zwei Teile gliedern: in die (in der Regel formal wohldefinierte) Semantik der Modellierungssprache sowie die „konkrete Semantik“ jedes einzelnen Modellelements. Bei EPKs wird diese „konkrete Semantik“ durch die Beschriftung von Ereignissen und Funktionen in natürlicher Sprache angegeben. Ähnlich unterteilen Pfeifer und Niehaves [PN05] die Bedeutung graphischer Modelle in die Anordnung der formalen Modellierungselemente („model element structure“) sowie die Bedeutung der Modellelemente in der Sprache der Anwendungsdomäne („terminological structure“). Pfeifer und Niehaves unterstreichen, dass eine Validierung eines Modells immer beide genannten Aspekte berücksichtigen muss.

Bisher vorgeschlagene Verfahren zur Validierung von Geschäftsprozessmodellen beziehen jedoch (von wenigen Ausnahmen wie [ADW08] abgesehen) kaum die Beschriftung von Modellelementen in die Analyse ein. Generell spielen Funktionen und Ereignisse bei den gängigen Validierungsverfahren (siehe etwa [van97, Rum99, Men07]) keine Rolle. Bei

*Der Lehrstuhl für Angewandte Telematik und E-Business ist ein Stiftungslehrstuhl der Deutschen Telekom AG

diesen Verfahren werden lediglich Kontrollflussfehler (z. B. Deadlocks) betrachtet, die sich aus der Anordnung und den Typen der Konnektoren ergeben.

Beim Auswerten publizierter Modelle fielen uns jedoch mehrfach Fehlermuster auf, die nur bei einer Betrachtung der Beschriftung von Modellelementen - insbesondere Ereignissen - erkannt werden können. Ziel unserer Arbeit war es, auch solche Fehler automatisch zu lokalisieren. Durch eine automatische Erkennung soll insbesondere dem ungeübten Modellierer ein Werkzeug zur Hand gegeben werden, um die Qualität seiner Modelle bereits während des Modellierungsprozesses zu prüfen und zu verbessern.

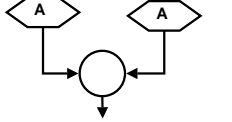
Zur technischen Validierung unseres Verfahrens wurde unser Algorithmus in das quelloffene Modellierungswerkzeug *bflow* Toolbox* integriert. Dieses bietet somit neben der Analyse von Syntax- und Kontrollflussfehlern [GL06, GLKK08] auch eine Modellüberprüfung auf die in diesem Beitrag beschriebenen inhaltlichen Fehler.

Die Fehlermuster, die untersucht werden, sind in Abschnitt 2 beschrieben. Anschließend zeigt Abschnitt 3, wie diese Fehler gefunden werden. Das Ergebnis einer ersten Validierung unserer Verfahren enthält Abschnitt 4. Schließlich werden in Abschnitt 5 die Ergebnisse diskutiert und mit anderen Arbeiten verglichen.

2 Betrachtete Fehlermuster

Bei der manuellen Analyse von Modellen aus der Praxis identifizierten wir einige häufige Modellierungsfehler, die durch gängige Verfahren zur Validierung des Kontrollflusses nicht entdeckt werden können, die jedoch durch einfache Analyse der Beschriftungen von Modellelementen zu identifizieren sind. Diese Muster werden im Folgenden vorgestellt.

2.1 Logisch identische Ereignisse vor/nach einem Konnektor

Muster A	
	<p>Ein Konnektor hat zwei logisch identische Ereignisse als Vorgänger oder Nachfolger. Folgen die logisch identischen Ereignisse auf einen XOR-Split, ist dies ein inhaltlicher Fehler. Andernfalls ist dieses Muster ein Hinweis auf die Möglichkeit, das Modell durch Weglassen redundanter Elemente zu verkleinern.</p>

Stehen die logisch identischen Ereignisse nach einem XOR-Split, ist dies offenbar stets ein inhaltlicher Fehler, da der XOR-Split ja gerade aussagt, dass nur eines der Ereignisse eintreten kann (und das andere nicht).

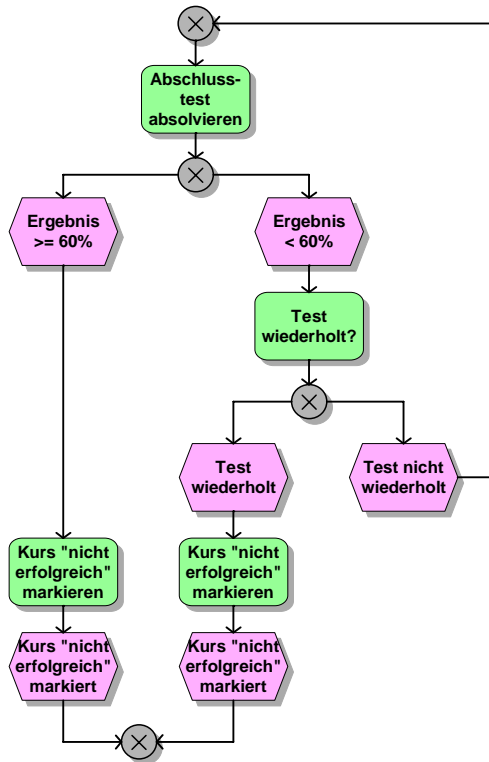


Abbildung 1: inhaltlich fehlerhaftes Modell

In allen anderen Fällen sind verschiedene Situationen möglich: Abb. 1 ist ein Modellausschnitt¹ aus [GKMZ07]. Hier weist das doppelte Vorkommen des Ereignisses “Kurs ’nicht erfolgreich’ markiert” auf einen Modellfehler hin: Vermutlich hätte im linken Zweig die Situation „Kurs erfolgreich“ modelliert werden sollen.

In den meisten Fällen jedoch dürfte das doppelte Vorkommen eines Ereignisses lediglich ein Indiz dafür sein, dass die EPK vereinfacht werden kann. Ein Beispiel zeigt Abb. 2, entnommen einer Diplomarbeit. Eine Modellierung wie im linken Modell von Abb. 2 widerspricht dem Grundsatz, die Zahl der Modellelemente auf das erforderliche Maß zu beschränken. Diese Forderung ist etwa unter der Bezeichnung „Minimalität“ in [Ron97] zu finden. Becker, Rosemann and Schütte fordern in den „Grundsätzen ordnungsgemäßer Modellierung“, dass „das Modell nicht mehr Elemente beinhalten soll, als zum Verständnis und zur Wiedergabe der Intention notwendig sind“ (zitiert aus [BRS95]).

Gesondert zu betrachten sind Ereignissen, die mit trivialen Standardtexten wie „erfolgreich durchgeführt“ oder „OK“ für Ereignisbeschriftungen beschriftet sind. Da hier der Bezug darauf fehlt, *was* erfolgreich durchgeführt wurde, können Ereignisse trotz gleichlautender

¹Die in diesem Beitrag gezeigten Modellbeispiele stellen keine kompletten EPK-Modelle dar, sondern zeigen lediglich unvollständige Modellfragmente.

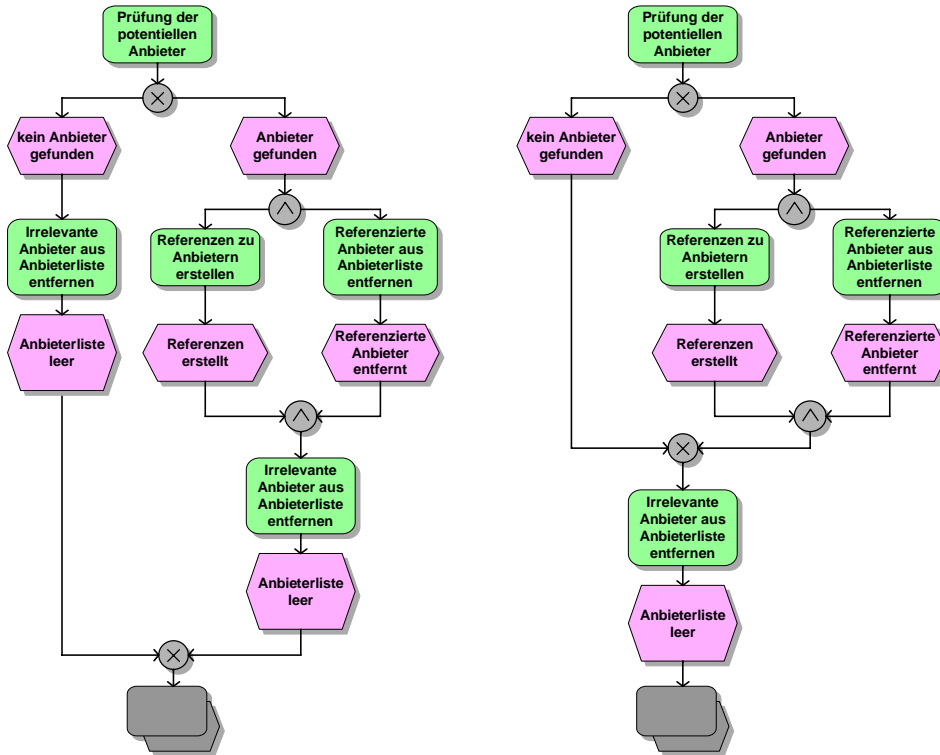


Abbildung 2: links: Originalmodell, rechts: vereinfachtes Modell

Beschriftung verschiedene betriebliche Situationen darstellen. Diese Fälle werden durch unseren Algorithmus, der solche “Trivialereignisse erkennt”, ausgeschlossen. Trotzdem sind gelegentliche “Fehlalarme” wie in dem in Abb. 3 (entnommen aus [KÖ6]) gezeigten Modellfragment möglich.

Weiter ist zu beachten, dass zwei Ereignisse trotz gleichlautender Beschriftung erkennbar verschieden sind, wenn ihnen in einer erweiterten EPK verschiedene Organisationseinheiten zugeordnet sind. Ein Ereignis „Prüfung erfolgreich“, ausgeführt von der Buchhaltung, ist ein anderes Ereignis als „Prüfung erfolgreich“, ausgeführt von der Rechtsabteilung.

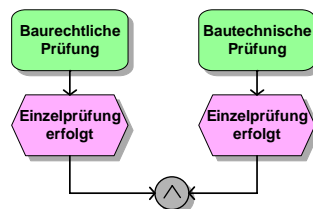
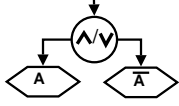


Abbildung 3: Fehlalarm bei Muster A

2.2 Zwei sich ausschließende Ereignisse durch AND-bzw. OR-Konnektor vereint

Muster B	
	<p>Ein Konnektor vom Typ OR oder AND hat zwei Ereignisse, die sich logisch widersprechen, als Vorgänger oder Nachfolger. Dies ist ein logischer Fehler im Modell. Möglicherweise sollte der Konnektor durch einen XOR-Konnektor ersetzt werden.</p>

In dem Modellfragment aus Abb. 4 (das einer Diplomarbeit entnommen wurde), wurde statt des die Situation korrekt beschreibenden XOR-Splits ein OR-Split verwendet. Bei unerfahrenen Modellierern ist eine solche Verwechslung zwischen OR und XOR ein häufiger Fehler.

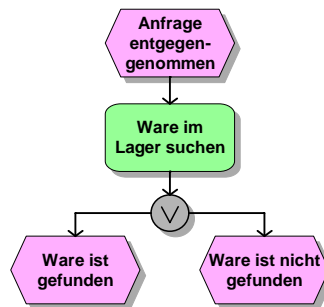


Abbildung 4:

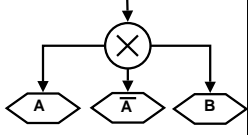
Leitet ein OR- oder AND-Split mehrere Kontrollflusszweige ein, so können diese im Modell parallel durchlaufen werden. Insbesondere heißt das, dass mehrere Ereignisse, die auf einen AND- oder OR-Join folgen, gemeinsam eintreten dürfen.

Schließen sich nun zwei auf einen OR- oder AND-Split folgende Ereignisse logisch aus (z.B. „Genehmigung erteilt“ / „Genehmigung verwehrt“), so ist davon auszugehen, dass ein Fehler im Modell vorliegt. Dieser besteht meist darin, dass statt eines XOR-Splits fälschlicherweise ein OR- oder AND-Split verwendet wurde.

Eine analoge Aussage gilt für den Fall, dass einander widersprechende Ereignisse direkt vor einem OR- bzw. AND-Join stehen.

Es ist anzumerken, dass sich die beiden Ereignisse auch logisch ausschließen können, wenn nicht eines die Negation des anderen ist. Ein Vorliegen des Musters soll auch erkannt werden, wenn z.B. eine Kombination von Ereignissen wie „Der Wert x hat zugenommen“ / „Der Wert x hat abgenommen“ gefunden wird, auch wenn außerdem noch der dritte Fall „Der Wert x ist konstant geblieben“ möglich wäre.

2.3 Ereignis, dessen Negation und weiteres Ereignis am XOR-Konnektor vereint

Muster C	
	<p>Ein Konnektor vom Typ XOR hat die Ereignisse A, $\neg A$ sowie ein weiteres Ereignis B als Vorgänger oder Nachfolger. Da stets entweder A oder $\neg A$ eintritt, ist die Berechtigung des dritten Ereignisses B fraglich (<i>Tertium non datur</i>).</p>

In diesem Muster folgen auf einen XOR-Split ein Ereignis A , dessen Negation $\neg A$ und mindestens ein weiteres Ereignis B . Da stets eines der Komplementäreignisse A und $\neg A$ eintreten muss, sollte es eigentlich keine Berechtigung für ein drittes Ereignis B geben.

Das Modellfragment in Abb. 5 (entnommen dem Zeitschriftenartikel [GK00]) zeigt ein Beispiel für das Auftreten des Musters. In einem solchen Modell leidet zumindest die Verständlichkeit²: Da offenbar stets genau eines der beiden linken Ereignisse eintritt, ist die Berechtigung der beiden anderen Ereignisse unklar.

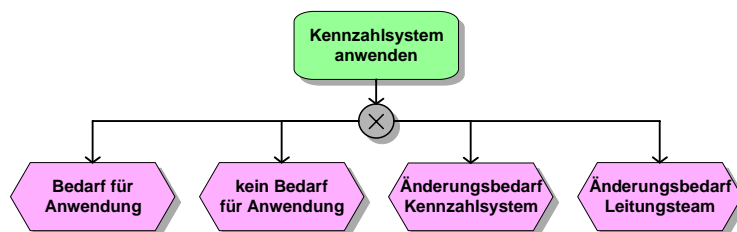


Abbildung 5: Es tritt stets eines der beiden linken Ereignisse ein.

Einen möglichen durch dieses Muster erzeugten „Fehlalarm“ zeigt Abb. 6. Hier entsteht der Eindruck eines Fehlers dadurch, dass die Aussage „CR ist vorhanden und (un)vollständig“ verkürzt wurden auf „CR ist (un)vollständig“. Für den Leser des Modells dürfte diese verkürzte Schreibweise sogar leichter verständlich sein.

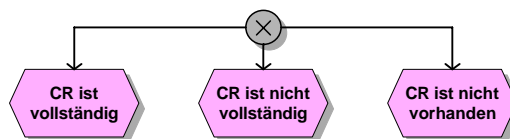


Abbildung 6: Fehlalarm bei Muster C

In Fällen wie in Abb. 6 ist es kaum möglich, Fehlalarme auszuschließen. Dies gelingt jedoch in einem häufigen Spezialfall, nämlich dann, wenn die Ereignisbeschriftungen A , $\neg A$ und B Entscheidungen der Art „ja“ / „nein“ / „unter Umständen“ beschreiben. Um auch solche Situationen richtig zu behandeln, meldet unsere Mustersuche in den Fällen,

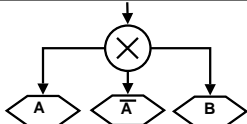
²Den fachlichen Inhalt des Modellausschnitts erlauben wir uns nicht zu beurteilen.

in denen B einschränkende Begriffe wie „zum Teil“, „eventuell“ oder „unter Vorbehalt“ enthält, kein Auftreten des.

In den im Rahmen der Validierung untersuchten Praxis-Modellen (vgl. Abschnitt 4) fanden sich die folgenden Kombinationen von Ereignissen nach einem XOR-Split, für die *kein* Auftreten des Musters gemeldet wird:

- „Berichte sind in Ordnung“ / „Berichte sind nicht in Ordnung“ / Berichte sind nur teilweise in Ordnung“
- „Bestellstatus zurückgewiesen“ / „Bestellstatus zugestimmt“ / „Bestellstatus teilweise zugestimmt“

2.4 Vergessen des Falles „Gleichheit“ beim Vergleich von Werten

Muster D	Ereignisse nach einem Konnektor beschreiben Größenvergleiche der Form " $a < b$ " / " $a > b$ " oder Größenveränderungen der Form "a ist gestiegen" / "a ist gesunken"
	Es besteht die Möglichkeit, dass bei der Modellierung der Fall der Gleichheit (" $a < b$ ") bzw. des Gleichbleibens ("a blieb unverändert") vergessen wurde.

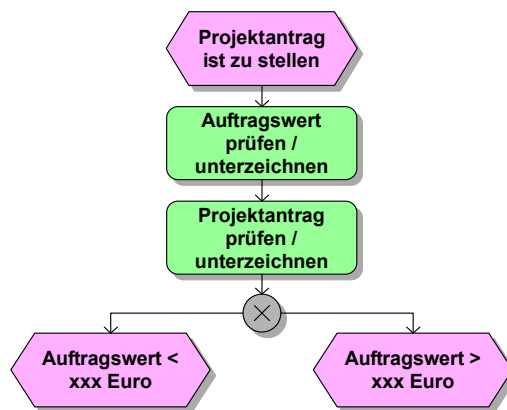


Abbildung 7: Vergessen des Falles „Gleichheit“ beim Vergleich von Werten

Als Ereignisse, deren Eintreten über die weitere Abarbeitung eines EPK-Modells nach einem Konnektor entscheidet, dienen in manchen Fällen Vergleiche von Werten.

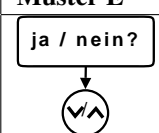
Im Beispiel von Abb. 7 (entnommen dem Buch [BKR08, Seite 634]) wird der Projektantrag sofort eingereicht, wenn der Auftragswert kleiner als ein bestimmter Betrag ist. Ist er größer als der genannte Betrag, muss eine zusätzliche Prüfung erfolgen. Im beschriebenen

Falle ist zu vermuten, dass der Modellierer den Fall „Auftragswert ist genau xxx Euro“ vergessen hat (vgl. [Amb03], Regel 233).

Unser Werkzeug identifiziert solche Situationen in einem Modell und weist auf ein mögliches Problem hin.

Analog untersuchen wir Aussagen zur Veränderung von Größen. Folgen etwa auf einen Split die beiden Ereignisse „Bedarf ist angestiegen“ und „Bedarf ist gesunken“ erfolgt ein Hinweis, dass der Fall „Bedarf ist gleich geblieben“ eventuell bei der Modellierung vergessen wurde.

2.5 AND- oder OR-Split nach einer Entscheidungsfrage

Muster E	
	<p>Auf eine Entscheidungsfrage folgt ein AND- oder OR-Split, so dass laut Modell mehrere Ereignisse zugleich auftreten können.</p>

In der Regel sollte nach einer Entscheidungsfrage (also einer Frage, auf die entweder mit „ja“ oder mit „nein“ geantwortet werden kann), ein XOR-Split folgen. Gleiches gilt für Fragen der Art „Prüfe, ob x oder y“. Unser Algorithmus identifiziert Fragen der genannten Art, denen ein AND- oder OR-Split folgt und gibt einen entsprechenden Hinweis aus.

Unser Algorithmus findet z.B. den Modellabschnitt von Abb. 8, bei dem nach der Überprüfung korrekterweise ein OR-Split stehen sollte³.

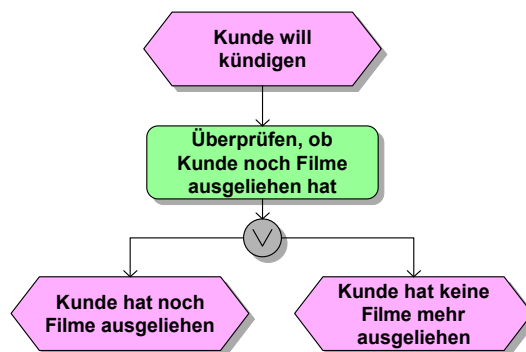


Abbildung 8: Nach einer solchen Entscheidung sollte immer ein XOR-Split stehen

³Im gezeigten Modell liegt außerdem Muster B vor; dies muss jedoch nicht immer der Fall sein.

3 Algorithmus

Wir benutzen den in [GL06] eingeführten Ansatz, die in einer ereignisgesteuerten Prozesskette (EPK) enthaltenen Informationen in eine Prolog-Faktenbasis zu übersetzen und durch Abfragen an den Prolog-Interpreter Fehler im Modell zu lokalisieren.

In bisherigen Arbeiten wurde auf diese Weise die syntaktische Korrektheit [GL06], Freiheit von Deadlocks und Kontrollflussfehlern [GLKK08] sowie mögliche Modellvereinfachungen, die zu einer leichteren Erfassbarkeit einer EPK führen [GL09], untersucht.

Die grundsätzlichen Schritte bei einer logikbasierten Analyse sind:

1. Übersetzung der im Modell enthaltenen Informationen in eine Prolog-Faktenbasis. Diese enthält dann Aussagen wie `event(i_3)` (es gibt ein Ereignis mit der ID `i_3`) oder `elementname(i_3, "Fahrzeug betanken")` (das Modellelement mit der ID `i_3` ist mit dem Text "Fahrzeug betanken" beschriftet).
2. Überführen der Beschriftungen der Modellelemente in eine Normalform, so dass Beschriftungen gleicher Bedeutung in dieselbe Normalform überführt werden
3. Suche nach den beschriebenen Mustern mittels Abfragen an den Prolog-Interpreter

3.1 Schritt 1: Überführen des Modells in eine Prolog-Faktenbasis

Die Überführung von EPK-Modellen in eine Prolog-Faktenbasis erfolgt mittels eines XSLT-Skripts, das eine EPML-Datei in Prolog-Regeln überführt. Details sind in [GL06] zu finden.

3.2 Schritt 2: Erzeugen einer Normalform der Modellbeschriftungen

In Schritt 2 werden die Beschriftungen der Modellelemente in eine Normalform gebracht.

Zweck dieser Normalform ist es, dass Beschriftungen gleicher Bedeutung in die gleiche Normalform überführt werden. Zu diesem Zwecke werden Stoppwörter, Synonyme und Antonyme beachtet. In unserem prototypischen Werkzeug haben wir 210 Synonyme und 70 Antonyme zu Begriffen aufgenommen, die sich sehr häufig in Modellen betrieblicher Abläufe finden.

Um die Population des Synonym/Antonym-Katalogs aufzubauen, wurde zunächst untersucht, welche Begriffe in Modellen eines uns vorliegenden EPK-Katalogs am häufigsten vorkommen. Hierfür wurden insgesamt 1154 deutschsprachige EPKs (darunter die 749 EPKs des deutschsprachigen SAP R/3-Referenzmodells) ausgewertet. Deren Beschriftungen von Ereignissen und Funktionen enthielten insgesamt 66088 Wörter, darunter 7599 verschiedene. Unter den am häufigsten gefundenen Begriffen wurden, sofern sinnvoll, Wörter zu Gruppen von Synonymen zusammengefasst. Die in einer solchen Gruppe enthaltenen Wörter werden dann bei der Normalformbildung alle durch die selbe Zeichenkette dargestellt.

Um die Normalform zu erhalten, bildet unser Algorithmus durch wiederholte Ersetzung von Zeichenketten eine Normalform einer Modellbeschriftung, indem Begriffe mit gleicher Bedeutung in die selbe Zeichenkette überführt werden. In den beiden Ereignisbeschriftungen „Der Antrag ist genehmigt“ und „Dem Antrag wurde zugestimmt“ werden zunächst Stoppwörter wie *der, die, das, ein, eine* u.ä. durch die leere Zeichenkette ersetzt, da sie für die Erfassung der Bedeutung der Zeichenkette verzichtbar sind. Da die Wörter „genehmigt“ und „zugestimmt“ beide im Synonymkatalog enthalten sind, werden weiterhin beide Zeichenketten in die selbe Normalform „Antrag nf_genehmigt“ überführt.

Auf die gleiche Weise werden Zeichenketten ersetzt, die zum Antonym-Katalog enthalten sind. In diesem Falle wird außerdem ein Flag gesetzt, das ausdrückt, dass die Normalform das Gegenteil der ursprünglichen Zeichenkette darstellt. Auf diese Weise wird etwa die Zeichenkette „Der Antrag wurde abgelehnt“ ebenfalls in die Normalform „Antrag nf_genehmigt“ überführt. Am gesetzten Flag ist jedoch erkennbar, dass sie das Gegenteil aussagt.

Tabelle 1 zeigt die in unserem Korpus von EPK-Modellen am häufigsten gefundenen Wörter sowie (falls zutreffend) ihre Ersetzung im Zuge der Normalform-Bildung:

Vorkommen	Wort	Behandlung
5719	ist	ersetzen durch leere Zeichenkette (Stoppwort)
1774	sind	ersetzen durch leere Zeichenkette (Stoppwort)
1400	zu	
821	für	
718	der	ersetzen durch leere Zeichenkette (Stoppwort)
696	und	ersetzen durch nf_und
645	vor	in Kombinationen wie „liegt vor“ ersetzen durch nf_vorhanden
536	wurde	ersetzen durch leere Zeichenkette (Stoppwort)
527	nicht	ersetzen durch leere Zeichenkette und Flag „Aussage ist negiert“ setzen
494	liegt	in Kombinationen wie „liegt vor“ ersetzen durch nf_vorhanden
477	an	
455	durchgeführt	ersetzen durch nf_beendet
404	werden	ersetzen durch leere Zeichenkette (Stoppwort)

Tabelle 1: Behandlung der häufigsten Begriffe bei der Normalformbildung

Eine besondere Behandlung erfahren Ereignisbeschriftungen der Form $x \diamond y$ mit $\diamond \in \{<, >, =, \leq, \geq, \}$ sowie Formen wie „x hat sich erhöht“, „x verringerte sich“, „x ist gefallen“, „x ist gestiegen“, „x blieb konstant“ etc. Auf eine Beschreibung der Details soll an dieser Stelle verzichtet werden. Die folgenden Beispiele illustrieren jedoch, dass als Resultat der Normalformbildung festgestellt werden kann, dass

- „ $x > 1000$ “ ebenso wie „ $1000 > x$ “ den Aussagen „ $x < 1000$ “ oder „x ist gleich1000“ widerspricht.

- “x ist gestiegen” und “x hat sich erhöht” die gleiche Aussage darstellen, jedoch im Widerspruch zu “x blieb konstant” und “x verringerte sich” stehen.
- die Aussagen “x ist kleiner als y” und “x ist größer als y” zwei der drei möglichen Fälle “größer/kleiner/gleich” beschreiben.

3.2.1 Schritt 3: Abfragen an die Faktenbasis

Aus der Beschreibung der in Abschnitt 2 betrachteten Muster wird deutlich, dass zur Identifikation der beschriebenen Muster die folgenden logischen Abfragen ausreichend sind:

1. **Vorgänger- und Nachfolgerbeziehung zwischen Ereignis und Konnektor:**
Diese wird einfach durch das Vorhandensein eines Kontrollflusspfeiles ausgedrückt, was sich in der Prolog-Repräsentation des Modells durch ein Prädikat $\text{arc}(x, y)$ widerspiegelt.
2. **Zwei Ereignisse beschreiben den gleichen Sachverhalt:**
Dies führt, wie in Schritt 2 dargestellt, dazu, dass beide Beschriftungen in die selbe Normalform überführt wurden. Eine Meldung zu Problem A wird nur ausgegeben, wenn die Ereignisse nicht eine triviale Beschriftung wie “erledigt” oder “fertig” haben.
3. **Zwei Ereignisbeschriftungen A und B widersprechen einander:**
Dies ist der Fall, wenn A und B in die selbe Normalform überführt wurden, jedoch bei der Ersetzung das Flag gesetzt wurde, das auf eine Ersetzung aus dem Antonymkatalog hinweist.
Ebenso ist dies der Fall, wenn A aus B hervorgeht, indem eine der negierenden Zeichenketten “un”, “nicht” bzw. “nicht-” eingefügt wurde.
Schließlich widersprechen sich Ereignisbeschriftungen, wenn sie genau zwei der drei Fälle “kleiner/größer/gleich” oder “verringert/vergrößert/unverändert” darstellen.
4. **Eine Beschriftung weist darauf hin, dass ein Ereignis nur teilweise eintritt:**
Dies wird angenommen, wenn die Ereignisbeschriftung bestimmte einschränkende Zeichenketten wie z.B. “möglicherweise”, “eventuell”, “vielleicht”, “möglichenfalls”, “unter Umständen”, “u.U.”, “unter Vorbehalt(en)”, “zum Teil”, “z.T.”, “teilweise”, “partiell” oder “unvollständig” enthält.
5. **Eine Beschriftung stellt eine Entscheidungsfrage (ja/nein-Frage) dar:**
Dies wird angenommen, wenn die Beschriftung einer Funktion die Zeichenkette “, ob” enthält (“Prüfe, ob der Auftrag ausgeführt wurde”) sowie wenn die Beschriftung mit einem Fragezeichen endet, jedoch nicht mit einem Fragewort (“wer”, “womit”, etc.) beginnt. Als Entscheidungsfrage wird somit z.B. “Erfolgte der Widerspruch rechtzeitig?” eingestuft, jedoch nicht “Welche Zusatzoptionen werden gewünscht?”.

Mit diesen genannten Prädikaten sowie weiteren einfachen Prädikaten, die z.B. bestimmen, ob ein Konnektor Split oder Join ist (siehe auch [GL06]) lässt sich z.B. eine Abfrage nach Muster A für zwei Ereignisse nach einem XOR-Split (was auf einen ernststen Modellierungsfehler hinweist) in Prolog wie folgt formulieren:

```
findemuster(E1,E2) :-
  split(C),type(C,xor),           \% C ist ein XOR-Split...
  arc(C,E1),arc(C,E2),           \% von dem aus es Pfeile zu E1 und E2 gibt
  event(E1),event(E2),          \% E1 und E2 sind Ereignisse
  E1 @< E2,                       \% bedeutet insbesondere: E1 ungleich E2
  elementname(E1,NameE1),        \% E1 hat die Beschriftung NameE1
  elementname(E2,NameE2),        \% E2 hat die Beschriftung NameE2
  equivalent(NameE1,NameE2),     \% NameE1 und NameE2 sind logisch äquivalent
  not(trivialereignis(NameE1)),   \% NameE1 ist kein Trivialereignis
  not(trivialereignis(NameE2)).   \% NameE2 ist kein Trivialereignis
```

4 Validierung

Wir überprüften unser Verfahren an 1253 EPK-Modellen in deutscher Sprache, die wir aus verschiedenen Quellen zusammengetragen haben.

Dabei stammten:

- 591 EPKs aus dem deutschsprachigen SAP R/3 Referenzmodell
- 127 EPKs aus Büchern (vornehmlich Lehrbücher zur EPK-Methode oder zu SAP R/3)
- 48 EPKs aus Dissertationsschriften
- 70 EPKs aus wissenschaftlichen Veröffentlichungen in Zeitschriften und Konferenzbänden
- 252 aus Bachelor-, Diplom- und Seminararbeiten
- 84 EPKs aus Praxisprojekten
- 22 EPKs aus Vorlesungsskripts
- 13 EPKs aus Software-Handbüchern

Bei den verbleibenden 37 Modelle handelt es sich um im Internet veröffentlichte EPKs, deren Einordnung in die o.g. Kategorien nicht möglich ist. Durch die Einbeziehung der verschiedenen Quellen ist zu erwarten, dass Modelle von Autoren mit höchst unterschiedlichen Erfahrungen mit der EPK-Modellierungsmethode berücksichtigt wurden.

Alle vom Werkzeug gemeldeten Problemmeldungen wurden durch manuelle Prüfung der betroffenen EPK untersucht, um zu entscheiden, ob tatsächlich ein Modellierungsproblem vorliegt. Insbesondere bei Muster C war dies nicht immer eindeutig zu entscheiden; in Zweifelsfällen wurde die Fehlermeldung als “unberechtigt” eingestuft. Es ergab sich die folgende Verteilung von gefundenen Fehlern bzw. “Fehlalarmen”:

Insgesamt wurden 114 tatsächliche sinnvolle Hinweise in 84 EPKs erkannt. Dem stehen 14 “Fehlalarme” in 13 EPKs entgegen. Die zahlreichen Fehlalarme bei Muster C legen nahe, dass eine Untersuchung dieses Musters in der Praxis nicht unbedingt sinnvoll ist. Bei allen anderen Mustern zeigen die gemeldeten Musterinstanzen fast ausnahmslos tatsächliche

Muster	gemeldete Vorkommen des Musters	davon unberechtigte Fehlermeldungen
Muster A	71 in 55 EPKs	1 in 1 EPK
Muster B	31 in 19 EPKs	1 in 1 EPK
Muster C	21 in 18 EPKs	12 in 11 EPKs
Muster D	3 in 3 EPKs	keine
Muster E	2 in 2 EPKs	keine

Tabelle 2: Gefundene Fehler in den einzelnen Fehlerklassen

Fehler bzw. Modellierungsprobleme. Dem Modellierer dürfte somit geholfen sein, wenn er zur Modellierungszeit eine Rückmeldung über Auftreten der genannten Muster und ggf. möglichen Modellverbesserungen erhält.

Bemerkenswert ist der Zusammenhang zwischen den gefundenen Fehlern und der Herkunft der Modelle. So stellt Muster B einen typischen Anfängerfehler dar - statt eines XOR-Splits wird das umgangssprachlich naheliegende OR verwendet. Während dieser Fehler in den studentischen Arbeiten (wie auch in einem Praxisprojekt aus dem Bereich Medien) recht häufig auftritt, wurde kein einziges Vorkommen dieses Musters im SAP R/3-Referenzmodell gefunden. Diese Beobachtung zeigt, dass von einem automatischen Erkennen der beschriebenen Fehlermuster hauptsächlich Anfänger profitieren dürften.

5 Diskussion und Vergleich mit verwandten Arbeiten

Das im vorangehenden Abschnitt beschriebene Ergebnis belegt, dass sich durch eine Analyse der Beschriftung von Modellelementen eine nennenswerte Zahl von Modellierungsfehlern aufspüren lässt. Diese Modellierungsfehler bleiben bei bloßer Betrachtung des Kontrollflusses unentdeckt.

Unser Algorithmus setzt keine Beschränkung der im Modell zu verwendenden Elemente der natürlichen Sprache voraus. Ereignisse und Funktionen können durch beliebigen Freitext beschrieben werden, was der heute meist gängigen EPK-Modellierungsmethode entspricht. Weit bessere Ergebnisse dürften zu erwarten sein, wenn die natürliche Sprache, die zur Beschreibung von Ereignissen und Funktionen verwendet wird, beschränkt wird.

Eine Vereinheitlichung der Beschriftungen von Modellierungselementen leistet etwa das Werkzeug Semtalk [FW05], das Ontologien verwendet, um eine einheitliche Verwendung von Substantiven und Verben über ein oder mehrere Modelle hinweg zu gewährleisten. Bei Verwendung eines solchen ontologiebasierten Konzeptes ist es auch möglich, echte inhaltliche Prüfungen von Geschäftsregeln automatisiert vorzunehmen. Fillies und Weichhardt führen in [FW03] ein Beispiel an, in dem zwei Geschäftsprozessmodelle „Bestellungseingang“ und „Bestellungsbearbeitung“ betrachtet werden. Sie nennen die Geschäftsregel „Nur bestätigte Bestellungen dürfen ausgeführt werden“ als Beispiel einer Regel, die man mit Hilfe solcher ontologiebasierter Modellierung modellübergreifend prüfen kann. Ein ähnliches Beispiel nennen Thomas und Fellmann. Sie beschreiben in in [TF07] einen Ansatz, Modellierung betrieblicher Abläufe mit EPKs mit Ontologien zu verknüpfen.

Wir sind davon überzeugt, dass durch die Einbindung von Ontologien in Geschäftsprozessmodellierungsmethoden mächtige Werkzeuge zur Konsistenzprüfung und Validierung von Geschäftsprozessmodellen sowie für Abfragen in Modellkatalogen geschaffen werden können. Vorhandene Ansätze werden beispielsweise in [WHM08] oder [GHSW08] beschrieben.

Wir sind jedoch ebenso davon überzeugt, dass sich in der betrieblichen Praxis solche ontologiebasierte Verfahren schwer durchsetzen werden, da dem Ziel (Verbesserung der Modellqualität) ein zumindest in der Einführungsphase erhöhter Aufwand im Modellierungsprozess entgegensteht. Letzlich sehen die beschriebenen Verfahren vor, eine Domänenontologie zusätzlich zum Modell zu erstellen, was in der Regel per Hand erfolgt (vgl. etwa die Beschreibung zur Erstellung von sog. semantischen EPKs in [FKS08]).

Unser Verfahren verzichtet auf eine aufwendige Erstellung einer Domänenontologie. Lediglich die im Standard-Synonym/Antonym-Katalog enthaltenen Begriffe stellen eine einfache Art einer (allerdings unvollständigen) Ontologiebeschreibung dar. In der aktuellen Form hat unser Synonym/Antonymkatalog noch einen recht geringen Umfang, so dass wir keinen Anspruch auf vollständige Erkennung der untersuchten Problemmuster erheben können. Ebenso gibt es sicher neben den betrachteten häufigen Problemmustern noch weitere.

Dem Nachteil dieser Unvollständigkeit steht der Vorzug der für den Modellierer einfachen Verfügbarkeit gegenüber. In der Validierung wurde gezeigt, dass sich bereits mit einem kleinen Katalog von Synonymen und Antonymen eine bemerkenswerte Zahl von Modellfehlern finden lässt. Im Unterschied zu Verfahren, die eine ontologiebasierte Modellierung verlangen, erhält der Modellierer Informationen zu diesen Fehlern, ohne dass die Modellierungsmethode komplexer wird. Im Werkzeug *bflow* Toolbox* ist die Mustersuche fest eingebaut und kann "auf Knopfdruck" gestartet werden (siehe Bildschirmfoto in Abb. 9). Dies erweitert die in [GLKK08] beschriebenen Möglichkeiten der *bflow* Toolbox*, dem Modellierer zur Modellierungszeit Rückmeldungen über mögliche Modellverbesserungen zu geben.

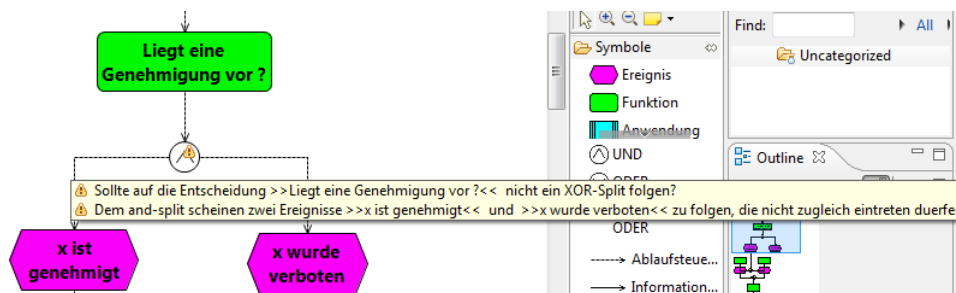


Abbildung 9: Integration in die bflow Toolbox, gemeldet werden hier Muster B und Muster E

In [ADW08] verwenden Awad et al. Verfahren aus dem Gebiet des Information Retrieval, um ein Ähnlichkeitsmaß zwischen Namen von Aktivitäten innerhalb eines BPMN-Diagramms zu definieren. Dieser Ansatz, der auf die englische Wortschatz-Datenbank WordNet [Fel98] zurückgreift, kommt ohne Beschränkung des zur Beschreibung von Aktivitäten verwendbaren Wortschatzes aus. Er erzielt gute Ergebnisse beim Erkennen gleicher oder ähnlicher Aktivitäten, auch wenn diese nicht identisch bezeichnet sind. Eine Kombination des in [ADW08] beschriebenen Verfahrens mit dem von der gleichen Forschungsgruppe beschriebenen Validierungsansatz [ADW08] erlaubt auch eine Überprüfung inhaltlicher Aussagen wie „Es darf kein Konto eröffnet werden, bevor die Identität des Inhabers überprüft wurde“.

Einen zu [ADW08] ähnlichen Ansatz beschreibt [KO07], hier steht jedoch nicht die Validierung von Modellen sondern das Erkennen von Modellvarianten im Vordergrund.

Ein wesentlicher Unterschied zwischen unserem Ansatz und [ADW08] sowie [KO07] besteht darin, dass wir bereits mit einem sehr kleinen Katalog von Synonymen und Antonymen beachtliche Ergebnisse erzielen können. Generell ist jedoch eine Verbesserung der Fehlererkennung zu erhoffen, wenn statt unseres eher einfachen Ansatzes zur Erkennung von identischen bzw. negierten Aussagen mächtigere Verfahren (wie in [ADW08], [KO07] oder [GZ05] beschrieben) verwendet werden.

Ein weiteres Feld für künftige Erweiterungen ist die Überprüfung der Einhaltung von Modellierungskonventionen für die Beschriftung von Ereignissen und Funktionen. So sollten Ereignisse etwa durch ein adjektivisch verwendetes Partizip („Der Antrag wurde genehmigt“) und Funktionen durch den Infinitiv eines Verbs („Antrag genehmigen“) dargestellt werden. Abweichende Modellierungskonventionen („Der Antrag ist zu genehmigen“ / „Der Antrag wird genehmigt“) sind möglich. Bögl et al. zeigen in [BSPW08], wie mit Hilfe von Wortdatenbanken und semantischen Mustern für Modellbeschriftungen die Einhaltung von Modellierungskonventionen wirkungsvoll überprüft werden kann.

Wir planen, unseren Ansatz in Zukunft um weitere Muster, die mögliche Modellverbesserungen beschreiben, zu erweitern. Forscher und Praktiker sind eingeladen, die im Werkzeug *bflow*Toolbox* bereits integrierten Tests zu nutzen und zu erweitern. Rückmeldungen und Erweiterungsvorschläge sind herzlich willkommen. Der jeweils aktuelle Entwicklungsstand kann von der Website www.bflow.org⁴ heruntergeladen werden.

⁴Das Prolog-Programm befindet sich im Plugin `org.bflow.toolbox.prolog`, dort findet der interessierte Leser auch die Liste der Synonyme und Antonyme.

Literatur

- [ADW08] Ahmed Awad, Gero Decker und Mathias Weske. Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In *BPM '08: Proceedings of the 6th International Conference on Business Process Management*, Seiten 326–341, Berlin, Heidelberg, 2008. Springer-Verlag.
- [Amb03] Scott W. Ambler. *The Elements of UML Style*. Cambridge University Press, 2003.
- [BKR08] Jörg Becker, Martin Kugeler und Michael Rosemann. *Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. Springer-Verlag, 6. Auflage, 2008.
- [BRS95] Jörg Becker, Michael Rosemann und Reinhard Schütte. Grundsätze ordnungsgemäßer Modellierung. *Wirtschaftsinformatik*, 37(5):435–445, 1995.
- [BSPW08] Andreas Bögl, Michael Schrefl, Gustav Pomberger und Norbert Weber. Semantic Annotation of EPC Models in Engineering Domains by Employing Semantic Patterns. In *ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Volume AIDSS, Barcelona, Spain*, Seiten 106–115, 2008.
- [DBL07] *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management held in conjunction with the 3rd European Semantic Web Conference (ESWC 2007), Innsbruck, Austria, June 7, 2007*, Jgg. 251 of *CEUR Workshop Proceedings*, 2007.
- [EGS04] Werner Esswein, Andreas Gehlert und Grit Seiffert. Towards a Framework for Model Migration. In *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004, Proceedings*, Jgg. 3084 of *Lecture Notes in Computer Science*, Seiten 463–476. Springer, 2004.
- [Fel98] Christiane Fellbaum, Hrsg. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [FKS08] Agata Filipowska, Monika Kaczmarek und Sebastian Stein. Semantically Annotated EPC within Semantic Business Process Management. In Danilo Ardagna, Massimo Mecella und Jian Yang, Hrsg., *Business Process Management Workshops*, Jgg. 17 of *Lecture Notes in Business Information Processing*, Seiten 486–497. Springer, 2008.
- [FW03] Christian Fillies und Frauke Weichhardt. Towards the Corporate Semantic Process Web. In *Berliner XML Tage*, Seiten 78–90, 2003.
- [FW05] Christian Fillies und Frauke Weichhardt. On Ontology-based Event-driven Process Chains. In *EPK 2005, Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, 2005.
- [GHSW08] Guido Governatori, Jörg Hoffmann, Shazia Sadiq und Ingo Weber. Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. In *BPD-08: 4th International Workshop on Business Process Design*, September 2008.
- [GK00] Frank Giesa und Herbert Kopfer. Management logistischer Dienstleistungen der Kontraktlogistik. *Logistik Management*, 2(1):43–53, 2000.
- [GKMZ07] Guido Grohmann, Wolfgang Kraemer, Frank Milius und Volker Zimmermann. Modellbasiertes Curriculum-Design für Learning Management Systeme: Ein Integrationsansatz auf Basis von ARIS und IMS Learning Design. In *Wirtschaftsinformatik*, Seiten 795–812. Universitätsverlag Karlsruhe, 2007.

- [GL06] Volker Gruhn und Ralf Laue. Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In *EPK 2006, Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 5. Workshop der Gesellschaft für Informatik e.V.*, Seiten 69–84, 2006.
- [GL09] Volker Grund und Ralf Laue. Reducing the Cognitive Complexity of Business Process Models. In *IEEE International Conference on Cognitive Informatics, Hong Kong 2009*, 2009.
- [GLKK08] Volker Gruhn, Ralf Laue, Heiko Kern und Stefan Kühne. EPK-Validierung zur Modellierungszeit in der bflow* Toolbox. In Peter Loos, Markus Nüttgens, Klaus Turowski und Dirk Werth, Hrsg., *MobIS*, Jgg. 141 of *LNI*, Seiten 181–194. GI, 2008.
- [GZ05] Vincenzo Gervasi und Didar Zowghi. Reasoning about inconsistencies in natural language requirements. *ACM Trans. Softw. Eng. Methodol.*, 14(3):277–330, 2005.
- [KÖ6] Markus König. Workflow-Management in der Baupraxis. In *4. Tag des Baubetriebs 2004 - Tagungsbeiträge Nachtragsmanagement in Praxis und Forschung, Schriften der Professur Baubetrieb und Bauverfahren*. Bauhaus-Universität Weimar, 2006.
- [KO07] Agnes Koschmider und Andreas Oberweis. How to detect semantic business process model variants? In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, Seiten 1263–1264, New York, USA, 2007. ACM.
- [Men07] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. Dissertation, Wirtschaftsuniversität Wien, 2007.
- [PN05] Daniel Pfeiffer und Björn Niehaves. Evaluation of Conceptual Models - A Structuralist Approach. In *Proceedings of the 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy, ECIS 2005, Regensburg, Germany, May 26-28, 2005*, 2005.
- [Ron97] Ron Weber. *Ontological Foundations of Information Systems*. Bericht 4, Coopers and Lybrand Accounting Research Methodology monograph, 1997.
- [Rum99] Frank J. Rump. *Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozeßketten*. B. G. Teubner Verlag Stuttgart Leipzig, 1999.
- [TF07] Oliver Thomas und Michael Fellmann. Semantic EPC: Enhancing Process Modeling Using Ontology Languages. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management held in conjunction with the 3rd European Semantic Web Conference (ESWC 2007), Innsbruck, Austria, June 7, 2007* [DBL07].
- [van97] Wil M. P. van der Aalst. Verification of Workflow Nets. In *Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN '97, Toulouse, France, June 23-27, 1997, Proceedings*, Seiten 407–426, 1997.
- [WHM08] Ingo Weber, Jörg Hoffmann und Jan Mendling. Semantic Business Process Validation. In *SBPM-08: 3rd international workshop on Semantic Business Process Management at ESWC-08*, Juni 2008.

Using Glossaries to Enhance the Label Quality in Business Process Models

Nicolas Peters and Matthias Weidlich

Hasso Plattner Institute at the University of Potsdam, Germany

`nicolas.peters@student.hpi.uni-potsdam.de`

`matthias.weidlich@hpi.uni-potsdam.de`

Abstract: Conceptual models are mostly used for human to human communication. Besides several other aspects, that is, the chosen modelling notation or the model layout, the labelling has a strong influence on the understandability and, therefore, quality of a process model. Consequently, labels should be reused and aligned across different process models, whereas similar labels such as homonyms should be avoided. In order to support these goals, in this paper, we describe an approach that applies a glossary for process modelling. On the one hand, we show how such a glossary that considers structural as well as control flow aspects is generated from an existing collection of process models. The applicability of our glossary generation and the appropriateness of the chosen structural and behavioural aspects is evaluated with the SAP reference model. On the other hand, we introduce two prototypes, the label checker and the label suggester, that illustrate the application of the glossary in the course of modelling.

1 Introduction

Business processes modelled with the Business Process Modeling Notation (BPMN) [OMG09] or as Event-driven Process Chains (EPC) [KNS92] are often used for human to human communication. Therefore, process models have to be easily understandable for the audience in order to benefit by using these models for discussions. Understandability of a process model always depends on its context, i.e., the purpose of the model and the involved stakeholders. For instance, process models designed for software developers as a basis for a system implementation or configuration will differ significantly from those that are used by managers for high-level decision making. Depending on the context, many specific factors affect the understandability of a process model, among them the chosen notation [RD07], the number of different elements used, as well as the model structure [MRC07].

Besides these aspects, the labelling of process model elements heavily influences the understandability [MRR09]. In order to illustrate this causality, Figure 1 depicts three process models of the same structure and notation with different styles of labelling. Without meaningful labels as in model a) one cannot even guess what the process model is about. Slightly better, in model b), one already knows the actions in the process model. Still, the domain the process model relates to is subject for interpretation. In contrast, using

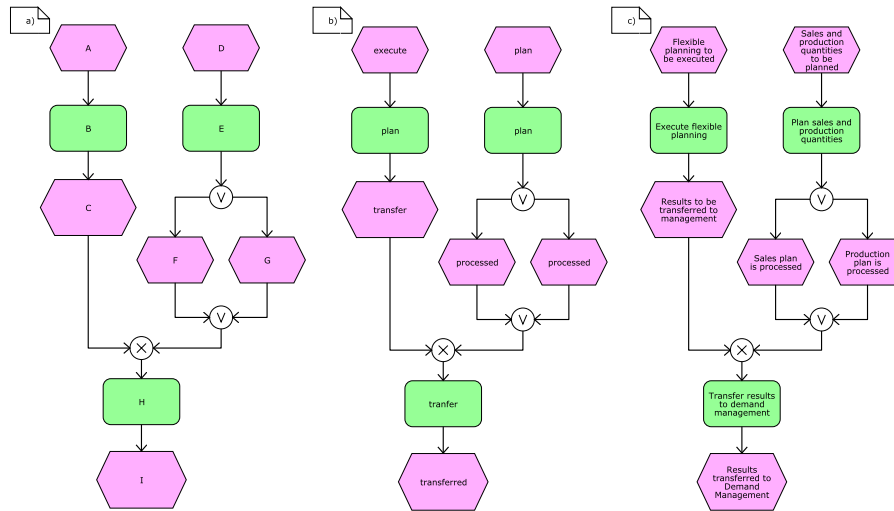


Figure 1: Impact of labelling on the understandability. a) nonsense labelling, b) only verbs, and c) verb-object style labelling

the verb-object style labelling [MRR09] for functions as in model c) allows for a good understanding of the described process. Based thereon, it is easy to infer that the model originates from the sales domain and defines a process of planning sales quantities.

Glossaries have proved useful in team-based environments and, therefore, are an inherent part of many project management methods [Kar88]. Of course, this holds also true for process modelling initiatives [Ros03]. In general, a glossary defines a centralized terminology for a specific domain. Such a glossary usually contains a list of terms and a description for each term. By using a glossary one can ensure that all participants of a collaborative effort have the same understanding of the terms they are using. That, in turn, reduces costs by preventing misunderstandings and shortening discussion times. Furthermore, glossaries are usually controlled by experts and contain terms and descriptions of high quality. This makes glossary entries ideal candidates for the labels of process model elements. Usage of a glossary in the course of modelling results in process models with a high labelling quality. The understandability of these process models increases, because the labels in the different process models are aligned to each other.

In this paper, we describe an approach to use a glossary for labelling process model elements to enhance their labelling quality. In particular, we answer the following two questions: Where does the glossary come from? And how can the glossary be applied to help a modeller creating process models? The first question is tackled by generating a glossary from the labels of an existing process model repository. This approach is motivated by the fact that there exist several reference process models for different domains (cf., [CKL97, Ste01]). These reference models are generic conceptual models that formalise recommended practices [FL03, Fra99, RvdA07]. They are domain-specific and have been

created to streamline existing process models or to improve the understanding of a technical system. Therefore, we assume these models to have a high labelling quality, which, in turn, qualifies them for acting as the basis of a glossary. Furthermore, we do not only consider the labels of the reference model, but also structural and control flow aspects of the given process models. That is, the glossary is enriched with element type information and behavioural characteristics. We show that these aspects are of high value for modelling support by analysing the characteristics of the SAP reference model [CKL97].

With respect to the application of the glossary in the course of modelling, we present two prototypes supporting the modeller. The first application is a label checker that marks all elements with potentially invalid labels and provides feedback on how to resolve the error. With this tool existing models (or whole process repositories) can be checked against the glossary. The second application is a label suggester that proposes labels based on the entered characters, as well as the label's context. The suggester enables a modeller to easily adopt the glossary labels in his models, as well as to find a substitute in the glossary for an invalid label.

Against this background, the remainder of this paper is structured as follows. Section 2 introduces the preliminary concepts used in this paper. Section 3 introduces our approach of generating a glossary for a given collection of process models, while Section 4 reports on findings that stem from the application of our approach to the SAP reference model. Section 5 illustrates the application of the glossary in the course of modelling by presenting two prototypes. Finally, we review related work in Section 6 and conclude in Section 7.

2 Preliminaries

This section provides preliminaries for our work. First, Section 2.1 shortly introduces EPCs as the process modelling language used throughout this paper. Note that, however, our approach for glossary based modelling itself does not rely on specific features of EPCs. Therefore, it can be transferred to other modelling languages directly. Second, Section 2.2 gives details on behavioural profiles as means to capture control flow characteristics of process models.

2.1 Event-driven Process Chains (EPC)

Event-driven process chains (EPCs) [KNS92, NR02] are a popular notation for modelling business processes. They are widely used for human to human communication and have also been applied in the field of reference models. For instance, the SAP reference model consists of hundreds of EPCs. In general, EPC models are a graph comprising functions and events in alternating order. While the former describe elementary actions, the latter specify the process state. Further on, control flow dependencies are expressed using directed flow arcs as well as split and join connectors that are typed as XOR, OR, or AND. A formal definition of EPC syntax can be found in [KNS92]. Note that there are various different

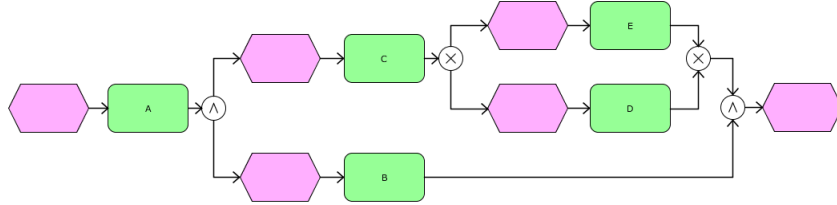


Figure 2: An EPC process model example.

formalisations of execution semantics for EPCs (cf., [KNS92, Men08, Kin04]), as the synchronisation behaviour of the converging OR-connector raises numerous questions (e.g., in cyclic structures). However, the differences of these semantics can be neglected in our context.

2.2 Behavioural Profiles

As mentioned before, our approach assumes a glossary that takes control flow aspects into account. In order to formalise these aspects, we apply the notion of behavioural profiles [WMW09]. These profiles have been introduced as a consistency notion in the field of process model alignment and capture behavioural characteristics of a process model by three different relations, i.e., *strict order*, *exclusiveness*, and *observation concurrency*. All of these relations are defined based on the set of possible traces of a process model.

Strict Order Relation The strict order relation holds between two process elements x and y , if x might happen before y , but not vice versa. In other words, x will be before y in all traces that contain both elements. Moreover, the *reverse strict order relation* holds for any inverted element pair that is in strict order. Note that both relations do not enforce a direct causality. That is, the occurrence of one of the elements in a trace does not enforce the occurrence of the other element.

Exclusiveness Relation Nomen est omen, the exclusiveness relation holds for two process elements, if they never occur together in any process trace.

Observation Concurrency Relation The observation concurrency relation holds for two process elements x and y , if x might happen before y and y might also happen before x . Thus, observation concurrency might be interpreted as the absence of any specific order between two process elements. It is worth to mention that this relation does not imply actual concurrent activation of the process elements. In particular, two process elements that are part of the same control flow cycle are also considered to be observation concurrent.

We illustrate these relations by means of the example EPC model in Figure 2. For instance, functions A and B are in strict order, whereas D and E are exclusive to each other, as there is

no trace of the EPC that contains both functions. Further on, B and C are in the observation concurrency relation, due to their concurrent activation. That is, B might happen before C or vice versa.

Initially, these relations have been defined for free-choice workflow nets [DE95, vdA98] in [WMW09]. There, it was also shown that the four relations (including the reverse strict order relation) partition the Cartesian product of process elements, i.e., every pair of process elements is in one of the four relation. We can easily lift these concepts to the level of EPCs under the assumption of execution semantics that are defined unambiguously. In particular, instantiation semantics for EPCs with multiple start events (cf., [DM09]) and semantics of the converging OR-connector have to be defined properly. Note the latter is an issue solely for complex synchronisation dependencies. For a block-structured joining OR-connector (all incoming arcs originate from a single splitting OR-connector), the behavioural profile would be the same as if the connectors are of type AND. That is, all elements in between would be considered to be observation concurrent to each other.

3 Generation and Setup of a Glossary

Glossaries may contain thousands of entries, which raises the question of how such a glossary is created. Manually adding all terms to a glossary is very time consuming, while it can be done by domain experts only. Thus, if there is existing data in a non-glossary format available for the domain of interest, it saves time and cost to automatically generate the glossary from that data. As mentioned above, we consider reference models consisting of a collection of process models as ideal candidates, as we assume these models to be consistent, precise, and contain labels of high quality.

In this section the structure of the glossary and its features as well as the process of generating a glossary from a collection of process models is described in detail. Section 3.1 discusses the question of which kind of label should appear in the glossary, while Section 3.2 gives details on the technical representation of the glossary. Finally, Section 3.3 and Section 3.4 show how the glossary is enriched with structural and control flow aspects.

3.1 The Choice of Glossary Terms

In general, a glossary is based on a list of terms, which might contain single words or complete phrases. The decision on the appropriate level of granularity for glossary items depends on the primary use case of the glossary. For instance, a glossary might contain names of data objects and a list of actions (verbs) that can be applied on the data objects. Such a glossary would allow to control the labelling of activities in a process model effectively, i.e., an activity label would be a combination of a verb and a data object name. This approach would allow for a glossary that is easier to manage than a glossary that contains all valid combinations of verbs and data object names. However, that would also require the definition of all valid phrase structures. One could imagine to apply association

rules mining [AIS93] in order to derive all valid combinations. Nevertheless, that would require automatic speech tagging [Bri92] in order to identify verbs and objects. Especially for short phrases (like those found in reference models), existing part of speech taggers are not very reliable. Thus, automatic generation is hard to accomplish for this kind of glossary.

In contrast, a glossary might also contain complete phrases that are directly applied as labels for process model elements. Such a glossary is useful, when the set of possible labels is rather small, i.e., the glossary is applied for a distinct domain. In particular, process models that are the basis for process execution might be build from a set of predefine actions.

Due to the obstacle of automatic part of speech tagging, we focus on glossaries that contain full phrases in the remainder of this paper.

3.2 Term Processing and Index Construction

In order to generate a list of labels for our glossary, all process models of the given collection are parsed in order to extract their element labels. However, a pure list of labels is not sufficient. For any reasonable kind of modelling support, the glossary has to offer a querying mechanism in order to search for labels. Similar to information retrieval systems [Kur04] or search engines [CMS09], the glossary has to provide a full text search for labels. Consequently, the basic components of search engines, i.e., text acquisition, text transformation, and index creation, have to be adapted for the use case of a glossary.

First, all labels are tokenized and each token is preprocessed before the search index is created. Such preprocessing aims at increasing the quality of search results by removing leading and trailing white spaces, as well as new line characters. In addition, all label tokens are lower-cased, stop words are filtered, and a stemming algorithm is applied. It is worth to mention that stemming algorithms are language dependant and error-prone. Thus, the stemming result is not always an actual word of that language, e.g., the porter stemmer [Por80] reduces the verb 'create' to 'creat'. However, search queries are preprocessed in the very same way, so that the stemmed terms of a query fit to the stemmed terms in the search index.

After the extracted labels have been preprocessed a search index is created for the glossary. The search index is an inverted index that maps preprocessed terms to the labels in which the terms occur. A very important task of the indexing process is the weighting of terms. Given a search query, these weights are the basis for the calculation of a ranking of result entries. For our glossary, we use the well-known TF-IDF scheme (see [ES07] for further information). The TF-IDF scheme computes for each term in a given label their relevance with regard to a corpus, i.e, the count of term occurrences over all labels. Based on the search index, a query to the glossary is answered using a Vector Space Model (VSM) retrieval model [SWY75].

Based thereon, the glossary can be queried for labels of process elements. Thus, assuming that the process model c) in Figure 1 is part of the model collection from which the glossary is derived, the query 'flex' would be answered with the labels 'Flexible planning to be executed' and 'Execute flexible planning'.

Table 1: Derivation of behavioural profile relations for the glossary (SO: strict order, RSO: reverse strict order, EX: exclusiveness, CON: observation concurrency relation)

		Relation 2			
Relation 1		SO	RSO	EX	CON
	SO	SO	CON	SO	CON
	RSO	CON	RSO	RSO	CON
	EX	SO	RSO	EX	CON
	CON	CON	CON	CON	CON

3.3 Element Types in the Glossary

The glossary that is build as introduced above offers full-text search over all labels. However, it is a common observation that labels for different element types have structural differences in process models. In case of EPCs, functions are often labelled with the verb-object style for describing an action (e.g., ‘Execute flexible planning’), whereas events describe the state of the process and, therefore, are often labelled with a passive sentence (e.g., ‘Flexible planning to be executed’). This distinction is not reflected in the glossary as described above (the search query ‘flex’ would return both labels). However, such type information might be used to improve glossary-based modelling support. If a search query also contains information about the element type for which a label is searched in the glossary, this information can be used to narrow the result set. That is, all labels that are not assigned to the element type of interest are removed from the result. Therefore, our glossary stores for each label the types of elements for which the label is used. In order to provide a ranking in case of labels that are used for more than one element type, the number of occurrences of a label in a certain element type is also stored.

Therefore, considering element types ensures that glossary labels are always applied in a *type consistent* manner.

3.4 Behavioural Profiles in the Glossary

Structural information such as element types improve the glossary-based modelling support by reducing the set of retrieved labels for a given query significantly. Similar improvements can be expected when considering the control flow characteristics of the process models from which the glossary is created. Here, the underlying assumption is that labels typically follow some kind of implicit ordering. For instance, ‘Receive invoice’ will typically occur before ‘Archive invoice’, whereas ‘Handle standard customer’ and ‘Handle VIP customer’ can be expected to never occur both in one process instance. In order to consider these information for modelling support, we also store the relations of the behavioural profile for all pairs of labels in the glossary.

Of course, there might be cases, in which more than one relation is found for a pair of

labels. That, in turn, might be due to fact that a label pair is found in more than one process model. In such a case, the relation to store in the glossary is selected according to Table 1. The idea behind this table is an order of the behavioural relations based on their strictness. We consider the exclusiveness relation as the strongest relation, as it completely disallows two labels to occur in one process trace. In contrast, the observation concurrency relation can be seen as being the weakest relation. It allows two labels to occur in any order in a process trace. Consequently, the strict order and reverse strict order relation are intermediate relations, as they disallow solely a certain order of two labels. Given two labels with different behavioural relations in two process models, the weakest of the two behavioural relations will be stored in the glossary (cf., Table 1). A behavioural relation between two labels is a constraint based on which the result set for a search query is reduced. Therefore, it is reasonable to use solely the weakest of all behavioural relations found for two labels in the respective model collection.

In order to leverage the behavioural relations for labels stored in the glossary, a query against the glossary might specify a so called search context. This context is given by two sets of labels of process model elements that precede or succeed the model element for which the search query is run. Based thereon, the set of results derived based on the full-text search (cf., Section 3.2) and the structural information (cf., Section 3.3) is further reduced. We remove all labels of the result set that fulfil one of the following requirements.

- They are in an exclusiveness relation with one of the labels in the search context.
- They are not in strict order with the succeeding labels in the search context.
- They are not in reverse strict order with the preceding labels in the search context.

As a result, the glossary returns solely these labels for a search query that can be applied for a certain model element without violating the behavioural relations as stored in the glossary for the respective labels. Consequently, the usage of a label from the query result is always *behaviour consistent* with respect to the information stored in the glossary.

4 Case Study: Generating a Glossary from the SAP Reference Model

This section demonstrates the applicability of the glossary generation and setup by presenting an implementation of our approach for the SAP reference model. In addition, we prove the appropriateness of the structural and control flow aspects that are part of our glossary by an experimental setup based on the SAP reference model.

In general, our implementation is based on two components. On the one hand, we use the well-known framework Apache Lucene¹, which provides a full-featured text search engine library written in Java. That is, it offers an implementation of the VSM model, as well as basic term preprocessing as introduced above. While the framework could easily be adapted to take element type information into account, we used the JBPT library² for taking the control flow aspects into account. The library provides an implementation of the behavioural profiles for free-choice workflow nets along with an EPC parser and an EPC to

¹<http://lucene.apache.org/>

²<http://code.google.com/p/jbpt/>

Petri net mapping. The mapping follows on standard EPC formalisations (e.g., [vdA99]). As mentioned in Section 2.2, however, a certain preprocessing of EPCs has to take place in order to map them to free-choice workflow nets. In particular, start and end events need to be normalized, such that there is a single start event and a single end event. Further on, block-structured OR-connectors are replaced with AND connectors as that enables a mapping to free-choice Petri nets without affecting the resulting behavioural profile. These normalisation are also provided by the JBPT library.

The SAP reference model [CKL97] that is used to generate the glossary describes the functionality of the SAP R/3 system in its version 4.6. It comprises 604 process diagrams, which are expanded to 737 EPC models as some diagrams contain multiple disconnected EPCs. These EPC models capture different functional aspects of an enterprise, such as sales or accounting. That allows us to assess the amount of reused labels in the reference model and to determine the consistency with respect to structural and control flow aspects. Note that it is well-known that the SAP reference model contains models that are erroneous [MVvD⁺08]. That is, these models contain deadlocks or livelocks, or even syntactical errors that preclude any reasonable interpretation. Therefore, we exclude these models from the behavioural analysis.

Label reuse. Generation of the glossary based on every second process model of the SAP reference model (that is a set of 368 models) yields a glossary containing 2565 unique labels. If the other half of the reference model is checked against this glossary, 1319 out of 2508 unique labels are also defined in the glossary. That corresponds to a rate of 52.59%. It is obvious that not all labels can be found in the glossary as the test set is an extension to the set of models used for generating the glossary. However, one out of two labels is reused, which indicates how common it seems to reuse labels in reference models.

Element type consistency. For the same glossary and test set, we also analysed the types of elements that have the same label. It is worth to mention that only four labels of the glossary are used for both, functions and events:

- Invoice Verification
- Information System
- Order Settlement
- Shipment Cost Calculation and Settlement

These labels contain no verbs so that an application for both types of process elements is useful in general. Still, ‘Information System’ does neither describe an activity nor a state and has probably been used accidentally as a label for functions and events, respectively. Besides these four exceptional cases, we see that, despite their enormous quantity, all labels can be identified as being either a function label or an event label. That, in turn, underpins the usefulness to consider such type information in the glossary. As a consequence, it is no surprise that we observed a high consistency value regarding our experimental setup. There is not a single label in the test set that is used for another element type than defined in the glossary, i.e., all labels are type consistent. This result further emphasizes that element types should be considered in a glossary for process modelling.

Behavioural profile consistency. Finally, we evaluated the consistency of behavioural profile relations for labels in the glossary and in the test set. Note that we removed all EPCs

Table 2: Consistency matrix of the behavioural profile relations for our experimental setup.

Rel. in Test Set \ Rel. in Glossary	SO	RSO	EX	CON
SO	yes	no	no	yes
RSO	no	yes	no	yes
EX	yes	yes	yes	yes
CON	no	no	no	yes

that have been identified as erroneous (cf., [MVvD⁺08]) from the set for the generation of the glossary. As a consequence, behavioural profiles were generated for 268 process models, which led to behavioural relations for 2244 unique pairs of labels. Regarding the test set (again, erroneous EPCs are removed), behavioural profiles are computed for 243 models, yielding behavioural relations for 4732 label pairs (note that these pairs are not unique). Out of these 4732 label pairs, 498 were already defined in the glossary. For these pairs we checked consistency with the behavioural relation stored in the glossary according to Table 2. Following on our discussion of an order of strictness of the behavioural relations (cf., Section 3.4), a relation in the test set is consistent, if the same relation or a weaker relation is defined in the glossary. Again, we observe a high consistency between the relations of the glossary and those of the test set. Only two of the 498 label pairs of the test set showed a behavioural relation that is inconsistent with the glossary. That corresponds to the rate of 99.60%. It is worth to mention that for 494 out of 498 label pairs, the relation in the test set was even equivalent to the relation in the glossary. Thus, our assumption of an implicit ordering between labels seems to hold for the SAP reference model. Therefore, considering control flow aspects between labels based on behavioural profiles is a useful feature for a process modelling glossary.

5 Application of a Glossary

After we introduced our approach of generating and setting up a glossary, the question of its application during process modelling has to be answered. In this section, we present two ways of applying the glossary to support process modelling aligned to a glossary, which, in the end, increases understandability of the models. Section 5.1 shows how the labelling of a complete process model is analysed based on the glossary, while Section 5.2 shows how label suggestion can be integrated in the process of modelling. For both modelling support features, we also show prototypical implementations based on the Oryx editor³ [DOW08], a web-based open source framework for process modelling.

³<http://www.oryx-project.org>

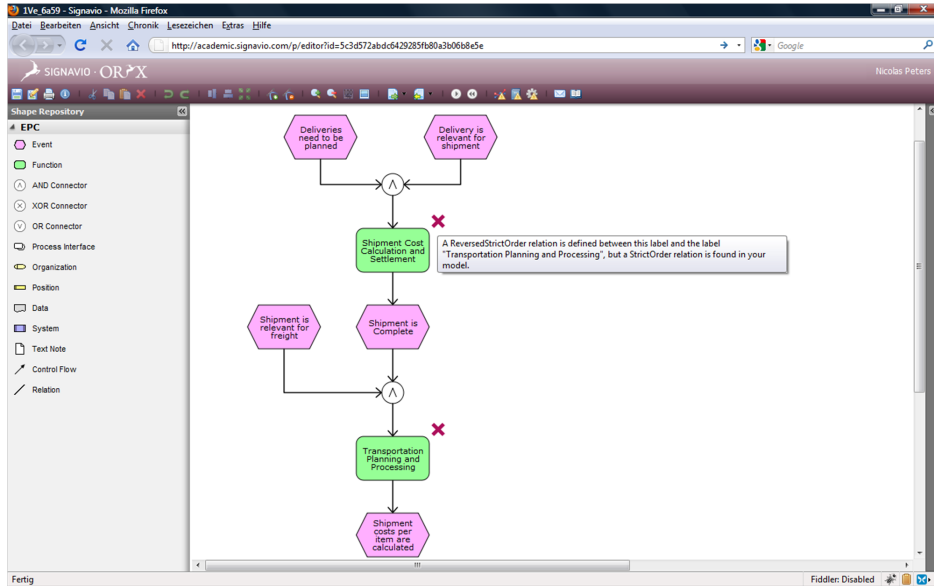


Figure 3: Label analysis reveals an inconsistent behavioural profile relation between two functions.

5.1 Label Analysis

The label analysis feature is a one click application that checks the labelling of a complete model against the glossary. The label analysis follows a two step approach. First, the consistency between the model and the glossary is calculated with respect to the used labels (how many labels are found in the glossary?), the element types (how many of the labels found in the glossary are used for the correct element types?), and behavioural profile relations (how many of the relations between labels in the model are consistent with the respective relations in the glossary?). Such a calculation provides simple consistency metrics that provide the modeller with a first feeling on how well a model is aligned with the glossary.



Figure 4: Label analysis reveals an inconsistent label of an event.

In the second step, the label analysis feature provides detailed feedback on labelling inconsistencies. As illustrated in Figure 3 and 4, process elements with inconsistent labels are highlighted directly in the editor with a red cross. Further on, a mouse over message provides details on the type of labelling inconsistency. That is, either the label cannot be found in the glossary, the label is assigned to a wrong element type, or the label and another label in the process model have an inconsistent behavioural profile relation. Figure 3 depicts an example for inconsistent behavioural profile relations. Here, the glossary defines a strict

order relation between the two function labels marked with red crosses, whereas they are related by reverse strict order in the process model. Figure 4, in turn, depicts the case of an event label that is not defined in the glossary.

5.2 Label Suggestion

In order to use the glossary for labelling elements after their creation, the label suggestion feature can be applied. It uses the information retrieval functionality of the glossary that has been introduced in Section 3. Whenever the modeller starts editing the label of an element, suggested labels from the glossary are immediately presented as depicted in Figure 5. These labels are retrieved by a query that contains the current element's label (or the modeller's input, respectively), the element type, and all preceding and succeeding labels. Each time the modeller enters a new character, the suggested labels are updated. Of course, the modeller might also enter several words, which narrows the result set. An interesting possibility is to enter parts of several words. For instance, using 'proc acqui' or 'acqui proc' as a query, a glossary based on the SAP reference model would return 'Processing of Asset Acquisition' as the first result. Thus, this feature allows to search for possible combinations of verbs and objects.

Note that the label suggestion feature might also be applied in order to resolve inconsistent labels that have been detected in the label analysis. In this case, the inconsistent label is queried against the glossary, such that labels similar to the inconsistent one are presented to the modeller. Based thereon, the modeller can replace the label with a similar one that is close to the intended semantics.

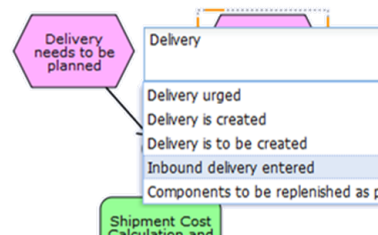


Figure 5: Label suggestion during modelling.

6 Related Work

Our approach of using a glossary for process modelling aims at increasing the model quality by providing a centralized terminology. There has been a lot of research on the quality aspects of process models (cf., [HMR08, SRG02, MRvdA09, BGR04]). Although quality of process models is affected by a whole spectrum of different factors, there is no doubt about the importance of the element labelling for the model understandability and, therefore, model quality.

Based on the SAP reference model that we used to generate our glossary, Mendling et al. have investigated common phrase structures [MRR09]. They found out that the verb-object style is the most common phrase structure for EPC functions, a labelling style that is often

referred to as a best practise (e.g., in [MCH03]). They also propose different approaches for a controlled object vocabulary and a controlled verb vocabulary. Such an approach would result in a one word glossary, instead of a complete label glossary as in our approach. As mentioned above these types of glossaries are fundamentally different, as, e.g., the one word glossary raises the question of automatic part of speech tagging. It is worth to mention that not only the functions of the EPCs in the SAP reference model, but also the start events show a set of dedicated phrase structures [DM09]. In particular, the distinction of start events (in the sense of events of the real world) and start conditions (EPC start events that express a condition) is reflected in the label structure. These findings are in line with our observation regarding the clear distinction between event and function labels.

Similar to our approach, Delfmann et al. describe a generic framework for defining a glossary of terms and phrase structures [DHLS08, BDH⁺09]. Their work is motivated by naming conflicts in process models that are created in distributed teams. Still, the generation of the glossary is regarded as a manual task, which might require serious efforts. Our approach is more lightweight in the sense that only complete labels instead of grammars are considered in order to benefit from automatic glossary generation. In addition, our approach considers structural as well as control flow aspects of process models to ensure a high degree of labelling consistency and to increase the usefulness of term suggestions.

Other work aims at providing modelling support based on a repository of model patterns that are extracted based on the element labelling. In [TRI09] the authors propose a set of generic activity patterns that might be used as basic building blocks of process models. While this approach requires a priori knowledge about the patterns, similar patterns might also be extracted from a model repository using association rules mining techniques [SWMW09]. Here, the patterns are lifted from the level of activities to the more abstract level of actions in order to enable reuse in a broader context. Again, this relies on part of speech tagging.

Support for process modelling might also be based on search techniques [HKL08]. Here, the main idea is to search a process repository for similar models in order to suggest extension of the current model. Of course, such a similarity search considers control flow and structural aspects of a process model, which resembles our idea of taking such information into account when querying a glossary. Similar approaches for modelling support might be based on ontology knowledge, e.g. [KO05]. Obviously, such approaches require the existence of a domain specific ontology. However, automatic generation of such an ontology imposes various challenges that go beyond the aforementioned part of speech tagging issue.

7 Conclusion

In this paper, we presented an approach that aims at increasing the labelling quality of business process models based on a glossary. We argued that the existence of reference models enables us to generate such a glossary automatically for a dedicated domain. Further on, we advocated the enrichment of a glossary with structural and behavioural information in terms of control flow aspects for the labels. The applicability of our approach as well

as the appropriateness of our choice on such structural and behavioural information was demonstrated using the SAP reference model. Further on, we illustrated the application of such a glossary in the course of process modelling by two modelling support features, which we implemented based on an open source modelling framework.

We showed that effective modelling support can be achieved once reference models are available. In particular, our approach of narrowing the set of potential labels for a certain element based on structural and behavioural information proved to be valuable. Even though our experiments provided evidence for the usefulness of the approach, future research has to evaluate the presented prototypes empirically in a user study.

We mentioned before that our approach is independent of the EPC notation and might be applied for other modelling languages as well. Still, languages with a huge set of element types (e.g., BPMN) might require further investigation. Probably, not all types imply differences in the labelling structure, so that clustering of element types has to be explored.

Further on, we based our approach on the assumption of high-quality labels in the collection of models from which the glossary is generated. Therefore, consistency checks for such a model collection (cf., [KMS08]) and quality metrics for the glossary itself have to be defined and evaluated. For instance, the number of homonyms used in a glossary can be regarded as such a metric, as usage of homonyms causes misunderstandings. Such homonymy might be detected using existing tools such as WordNet⁴.

References

- [AIS93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *COMAD*, pages 207–216, Washington, D.C., 1993.
- [BDH⁺09] J. Becker, P. Delfmann, S. Herwig, L. Lis, and A. Stein. Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In *ECIS*, June 2009.
- [BGR04] Wasana Bandara, Guy G. Gable, and Michael Roseman. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, 14(4):347 – 360, 2004.
- [Bri92] Eric Brill. A Simple Rule-Based Part of Speech Tagger. In *ANLP*, pages 152–155, 1992.
- [CKL97] Thomas A. Curran, Gerhard Keller, and Andrew Ladd. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Prentice-Hall, 1997.
- [CMS09] Croft, Metzler, and Strohman. *Search Engines: Information Retrieval in Practice*. Pearson Education, 2009.
- [DE95] Jörg Desel and Javier Esparza. *Free choice Petri nets*. Cambridge University Press, New York, NY, USA, 1995.

⁴<http://wordnet.princeton.edu/>

- [DHLS08] Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle. In *MobIS*, volume 141 of *LNI*, pages 23–38. GI, 2008.
- [DM09] Gero Decker and Jan Mendling. Process Instantiation. *Data & Knowledge Engineering (DKE)*, 68:777–792, 2009.
- [DOW08] Gero Decker, Hagen Overdick, and Mathias Weske. Oryx - An Open Modeling Platform for the BPM Community. In *BPM*, volume 5240 of *LNCS*, pages 382–385. Springer, 2008.
- [ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.
- [FL03] P. Fettke and P. Loos. Classification of reference models - a methodology and its application. *Information Systems and e-Business Management*, 1(1):35–53, 2003.
- [Fra99] U. Frank. Conceptual Modelling as the Core of the Information Systems Discipline - Perspectives and Epistemological Challenges. In *Proceedings of the America Conference on Information Systems (AMCIS)*, pages 695–698, 1999.
- [HKL08] Thomas Hornung, Agnes Koschmider, and Georg Lausen. Recommendation Based Process Modeling Support: Method and User Experience. In *ER*, volume 5231 of *LNCS*, pages 265–278. Springer, 2008.
- [HMR08] Mitra Heravizadeh, Jan Mendling, and Michael Rosemann. Dimensions of Business Processes Quality (QoBP). In *Business Process Management Workshops*, volume 17 of *LNBIP*, pages 80–91. Springer, 2008.
- [Kar88] J. Karimi. Strategic Planning for Information Systems: Requirements and Information Engineering Methods. *Journal of Management Information Systems*, 4:5–24, 1988.
- [Kin04] Ekkart Kindler. On the semantics of EPCs: A framework for resolving the vicious circle. In J. Desel, B. Pernici, and M. Weske, editors, *Business Process Management (BPM)*, volume 3080 of *Springer, LNCS*, pages 82–97, Potsdam, Germany, June 2004.
- [KMS08] E. Knauss, S. Meyer, and K. Schneider. Recommending Terms for Glossaries: A Computer-Based Approach. In *First International Workshop on Managing Requirements Knowledge*, pages 25–31, 2008.
- [KNS92] G. Keller, M. Nüttgens, and A.-W. Scheer. Semantische Prozeßmodellierung auf der Grundlage ‘Ereignisgesteuerter Prozeßketten (EPK)’. Veröffentlichungen des instituts für wirtschaftsinformatik (iwi), Universität des Saarlandes, January 1992.
- [KO05] Agnes Koschmider and Andreas Oberweis. Ontology Based Business Process Description. In *EMOI-INTEROP*, volume 160 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
- [Kur04] D. Kurupka. *Modelle zur Repraesentation natürlichsprachlicher Dokumente. Ontologie-basiertes Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos Verlag, 2004.
- [MCH03] Th. W. Malone, K. Crowston, and G. A. Herman. *Organizing Business Knowledge: The MIT Process Handbook*. The MIT Press, Cambridge, MA, USA, 1st edition, September 2003.
- [Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.

- [MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In *BPM*, volume 4714 of *LNCIS*, pages 48–63. Springer, 2007.
- [MRR09] J. Mendling, H. Reijers, and J. Recker. Activity labeling in process modeling: empirical insights and recommendations. *Information Systems (IS)*, 2009. to appear.
- [MRvdA09] J. Mendling, H.A. Reijers, and W.M.P. van der Aalst. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology (IST)*, 2009. to appear.
- [MVvD⁺08] Jan Mendling, H. M. W. Verbeek, Boudewijn F. van Dongen, Wil M. P. van der Aalst, and Gustaf Neumann. Detection and prediction of errors in EPCs of the SAP reference model. *Data Knowl. Eng.*, 64(1):312–329, 2008.
- [NR02] Markus Nüttgens and Frank J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise*, volume 21 of *LNI*, pages 64–77. GI, 2002.
- [OMG09] OMG. *Business Process Modeling Notation (BPMN) 1.2*, January 2009.
- [Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [RD07] J. Recker and A. Dreiling. Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. In *18th Australasian Conference on Information Systems*, pages 356–366, 2007.
- [Ros03] M. Rosemann. *Process Management: A Guide for the Design of Business Processes*, chapter Preparation of process modeling, pages 41–78. Springer, 2003.
- [RvdA07] Michael Rosemann and Wil M. P. van der Aalst. A configurable reference modelling language. *Inf. Syst.*, 32(1):1–23, 2007.
- [SRG02] Wasana Sedera, Michael Rosemann, and Guy G. Gable. Measuring Process Modelling Success. In *ECIS*, 2002.
- [Ste01] S. Stephens. The Supply Chain Council and the Supply Chain Operations Reference Model. *Supply Chain Management*, 1:9–13, 2001.
- [SWMW09] S. Smirnov, M. Weidlich, J. Mendling, and M. Weske. Action Patterns in Business Process Models. In *7th International Joint Conference on Service Oriented Computing (ICSOC)*, Stockholm, Sweden, 2009.
- [SWY75] Gerard Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620, 1975.
- [TRI09] Lucinea Thom, Manfred Reichert, and Cirano Iochpe. Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence. *International Journal of Business Process Integration and Management (IJBPIIM)*, 2009. to appear.
- [vdA98] Wil M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.
- [vdA99] Wil M. P. van der Aalst. Formalization and verification of event-driven process chains. *Information & Software Technology*, 41(10):639–650, 1999.
- [WMW09] Matthias Weidlich, Jan Mendling, and Mathias Weske. Computation of Behavioural Profiles of Processes Models. Technical report, BPT Technical Report 07, 2009.

Migration von EPK zu BPMN

Gero Decker, Willi Tscheschner
Signavio GmbH
(gero.decker,willi.tscheschner)@signavio.com

Jörg Puchan
Hochschule München
puchan@hm.edu

Abstract: Prozessmodelle sind wichtige Grundlagen zur effizienten und effektiven Gestaltung von Organisationen. Darüber hinaus fordern zahlreiche nationale und internationale Standards und Normen die Modellierung und Dokumentation der Geschäftsprozesse. Daher verfügen viele Unternehmen heute über eine große Menge an ausgereiften und aufwändig erstellten Prozessmodellen. Im deutschsprachigen Raum sind ereignisgesteuerte Prozessketten (EPK) weit verbreitet. International wird der Standard Business Process Modeling Notation (BPMN) zunehmend eingesetzt. Die internationale Zusammenarbeit und Verflechtung der Organisationen ist ein verstärkender Grund dafür, dass Unternehmen ihre EPKs in die BPMN-Notation überführen möchten. Hierzu wird in diesem Beitrag dargestellt, inwieweit dies automatisiert abgewickelt werden kann und welche Grenzen der Automatisierung gesetzt sind. Nach einem kurzen Überblick über die Konstrukte von erweiterten EPKs und BPMN werden sukzessive deren Transformationsmöglichkeiten dargestellt. Während bei zahlreichen Konstrukten die Transformation automatisiert erfolgen kann, ergeben sich in einigen Fällen Schwierigkeiten bzw. Mehrdeutigkeiten. In diesen Fällen ist i.d.R. nur ein semi-automatisches Vorgehen möglich. Durch die Einhaltung geeigneter Modellierungsrichtlinien für EPKs lassen sich einige dieser Problemfälle vermeiden. Eine Teilmenge der dargestellten Transformationsregeln wurde bereits in einer prototypischen Implementierung auf Grundlage des Modellierungswerkzeugs Signavio Process Editor umgesetzt.

1 Einleitung

Die Gründe für die Prozessmodellierung in Unternehmen sind vielfältig. Häufig werden Prozesse modelliert, um diese nachfolgend zu optimieren, im Unternehmen verbindlich zu machen oder prozessunterstützende Systeme zu entwickeln bzw. anzupassen. Weitere Motivationen zur Prozessmodellierung resultieren aus der Notwendigkeit, gesetzliche, aufsichtsrechtliche, revisorische oder ähnliche Anforderungen zu erfüllen (Compliance), freiwillig bzw. marktgetrieben die Prozessorientierung zu fördern (z.B. Zertifizierungen) oder Vergleiche mit Standardprozessmodellen (z.B. CMMI, ITIL) zu ermöglichen bzw. diese schrittweise einzuführen. In jedem dieser Fälle wird in Unternehmen und Behörden eine große Menge an Ressourcen, Zeit und Kosten aufgewandt, um diese Prozessmodelle zu erstellen, zu optimieren, zu implementieren und sie aktuell zu halten. Im deutschsprachigen Raum sind dabei seit Mitte der neunziger Jahre des vergangenen Jahrhunderts ereignisgesteuerte Prozessketten (EPK) und deren Erweiterungen (eEPK) auf der Grundlage bzw. nach den Vorgaben der "Architektur rechnergestützter Informationssysteme" (ARIS) zunehmend und heute weit verbreitet. Die Modellierungsmethode und -sprache wurde ursprünglich von Scheer entwickelt und dokumentiert [Sch92, Sch98] und dann im

Rahmen zahlreicher Projekte weiterentwickelt und verfeinert [SJ02, IDS06]. Heute sind EPKs bei zahlreichen Unternehmen im Einsatz. Es kann angenommen werden, dass ein Großteil der ca. 7.500 ARIS¹-Kunden der IDS Scheer AG EPKs zur Modellierung ihrer Prozesse einsetzen. Genaue Statistiken zum Einsatz und zum Bestand von EPKs liegen den Autoren allerdings nicht vor. Die internationale Entwicklung vor allem im englischsprachigen Raum ist zwar erst später in Gang gekommen, gewinnt aber durch die weltweite, unternehmensübergreifende Kooperation, Verflechtung und Standardisierung immer mehr an Bedeutung. Die Business Process Modeling Notation (BPMN [bpm09]) wurde 2004 erstmalig von der Business Process Management Initiative (BPMI) veröffentlicht. Der Standardisierungsprozess wurde von der Object Management Group (OMG) übernommen und eine überarbeitete Version wurde im Juni 2005 veröffentlicht. Die aktuelle Version 1.2 steht seit Februar 2009 zur Verfügung. Momentan wird BPMN 2.0 vorbereitet und eine Veröffentlichung ist im Laufe des Jahres 2010 zu erwarten.

Unternehmen arbeiten heute zunehmend international zusammen. Viele Unternehmen sind selbst internationale Konzerne. Auch internationale Standards und regulatorische Anforderungen verlangen zunehmend standardisierte und dokumentierte Prozesse, z.B. das US-Gesetz Sarbanes-Oxley Act 2002 (SOX, aufsichtsrechtliche Anforderungen für in den USA börsennotierte Unternehmen oder Unternehmen, deren Anteile in den USA außerbörslich gehandelt werden sowie deren Tochtergesellschaften), COBIT (Control Objects for Information and related Technology, internationaler Wirtschaftsprüferstandard [IT-05]) oder CMMI (Capability Maturity Model Integration, Best Practice-Modell des Software Engineering Instituts der Carnegie Mellon University mit 22 Referenzprozessen/sog. "Prozesskategorien" für die Optimierung der IT-Organisation [CKS08]). Daraus folgt, dass ein international anerkannter Standard wie BPMN zunehmend an Bedeutung gewinnt und gewinnen wird. Auch Unternehmen des deutschsprachigen Raums werden durch internationale Kooperationen bzw. Verflechtungen, aus Gründen der leichteren Gewinnung von qualifiziertem, international tätigen Personal oder aus anderen Gründen zunehmend BPMN als Modellierungssprache einsetzen wollen. Die Autoren begleiten derzeit einige Unternehmen, bei denen diese Entwicklung zu verzeichnen ist.

Anwenderunternehmen, die also wertvolle Prozessmodelle in Form von ereignisgesteuerten Prozessketten erstellt haben und durch externen Druck oder aus eigenem Antrieb zur Modellierungsmethode BPMN wechseln wollen, müssen also überlegen, wie sie diese "Wertgegenstände" möglichst aufwandsarm überführen können. In diesem Beitrag wird dargestellt, nach welchen Regeln ereignisgesteuerte Prozessketten möglichst automatisierbar in BPMN-Modelle überführt werden können und bei welchen Modellelementen eine automatisierte Lösung nicht unmittelbar möglich bzw. sicher ist, da die Semantik der Sprachen dies nicht zulässt.

Dieser Beitrag stellt somit eine direkte Transformation von EPK zu BPMN vor und ist wie folgt strukturiert. Nach einer kurzen Vorstellung der verwandten Arbeiten werden eEPKs und BPMN jeweils in einem eigenen Abschnitt vorgestellt. Die Transformation wird in Abschnitt 5 vorgestellt. Die Ergebnisse werden anschließend diskutiert. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick.

¹ ARIS ist eine geschützte Marke der IDS Scheer AG

2 Verwandte Arbeiten

Die Transformation von EPKs nach BPMN wird in der Literatur bisher kaum diskutiert. Ein Beitrag von Hoyer et al. [HBS07] ist ein erster Schritt in Richtung einer (semi-)automatischen Transformation. Neben einer Abbildung der Kontrollflusskonstrukte wird hier vor allem auf die Unterscheidung zwischen internen Aktivitäten und Kommunikationsaktivitäten (d.h. Senden und Empfangen von Nachrichten).

Grundlage für eine saubere Transformation ist eine präzise Definition der Semantik der Quell- und Zielsprache. Existierende Arbeiten zu EPK und BPMN konzentrieren sich hierbei vor allem auf die Kontrollflussesemantik. Während die Semantik von XOR- und UND-Konnektoren unstrittig ist, ist eine rege Diskussion bzgl. ODER-Konnektoren (OR-Joins) entstanden. Sowohl auf der EPK-Seite als auch auf der BPMN-Seite gibt es eine Reihe von Arbeiten, die jeweils über Transformationen zu einem Formalismus die Kontrollflussesemantik der einzelnen Modellierungskonstrukte abbilden [DGHW07, Men07]. Eine Übersicht über die Diskussion findet sich in [KMWL09].

Neben der Kontrollflussesemantik wurde in der Literatur auch die Prozessinstanzierungssemantik von EPK und BPMN beleuchtet. Dies ist besonders relevant, da sowohl EPK als auch BPMN mehrere Eintrittspunkte in einen Prozess zulassen und die Anzahl der Modellierungsfehler stark mit der Anzahl an Startknoten in einem Prozessmodell korreliert [DM09].

Als Alternative zu einem direkten Mapping von EPK nach BPMN kann auch ein zweistufiges Mapping vorgenommen werden, welches zunächst ein EPK-Modell in eine dritte Sprache abbildet und dieses Zwischenergebnis wiederum nach BPMN übersetzt. Ein solcher Ansatz wird propagiert in [VZHA05] unter Verwendung einer speziellen XML-Sprache. Zwar werden EPK und BPMN als motivierendes Beispiel in diesem Beitrag genannt, die genauen Mappingvorschriften werden allerdings nicht näher erläutert. Besonderes Augenmerk gebührt bei dieser Strategie der Frage, ob die dritte Sprache tatsächlich eine semantische Obermenge der Quell- oder Zielsprache bildet. Ist dies nicht der Fall, kann durch den "Umweg" mehr Information verloren gehen als bei einer direkten Abbildung. Insofern kommt der Einsatz von Petrinetzen oder BPEL nicht in Frage, obwohl entsprechende Mappings von EPKs / zu BPMN existieren [SKI08, KWL09, WDGW08, SKLN09].

Gerade in Situationen, bei denen die Zielsprache eine höhere Ausdrucksmächtigkeit besitzt als die Quellsprache, sind Spracherweiterungen / -verfeinerungen in der Quellsprache eine häufig verwendete Strategie, um eine genauere Abbildung zu erreichen. Für EPKs existieren Erweiterungen, die genau aus diesem Grund hinzugefügt wurden, z.B. für ein Mapping in Richtung Web-Service-Orchestrierungen [HW08].

3 EPKs im Überblick

Im Folgenden werden die Grundelemente der ereignisgesteuerten Prozessketten (EPKs) grob skizziert. Für Details wird auf das Methodenhandbuch ARIS [IDS06] bzw. die zahlreich vorhandene Literatur (z.B. [Sch98] oder [Gad08]) verwiesen. Ereignisgesteuerte

Prozessketten (EPKs) bzw. deren Erweiterungen sind nicht formal normiert und unterliegen pragmatischer, kontinuierlicher Verbesserung bzw. Veränderung, wie es für Industriestandards häufig der Fall ist. Darüber hinaus werden von unterschiedlichen Beratern und Anwendern gelegentlich eigene Erweiterungen oder Ergänzungen entwickelt und propagiert. Im Kern stimmen die bekannten Grundelemente der EPKs jedoch im Wesentlichen überein. Bei ereignisgesteuerten Prozessketten werden die Elemente der Funktionssicht (Funktionen) über Ereignisse verkettet. Ereignisse repräsentieren einen “betriebswirtschaftlich relevanten Zustand eines Informationsobjektes, der den weiteren Ablauf des Geschäftsprozesses steuert oder beeinflusst. Ereignisse lösen Funktionen aus und sind Ergebnisse von Funktionen. Im Gegensatz zu einer Funktion, die ein zeitverbrauchendes Geschehen darstellt, ist ein Ereignis auf einen Zeitpunkt bezogen.” (vgl. [IDS06]). Die Verkettung der Elemente erfolgt entweder sequenziell (repräsentiert durch eine gerichtete Kante) oder durch einen logischen Verknüpfungsoperator (UND, ODER, exklusives ODER) mit gerichteten Kanten.

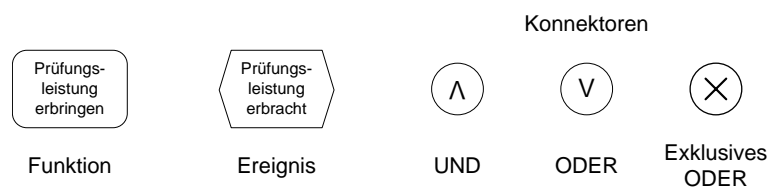


Abbildung 1: EPK-Konstrukte

Abbildung 1 zeigt die drei Hauptkategorien von EPK-Konstrukten: Funktionen, Ereignisse und Konnektoren. Die zulässigen Verknüpfungsregeln für Konnektoren sind in Abbildung 3 dargestellt (vgl. [IDS06], S. 110).

EPKs starten und enden stets mit einem Ereignis. In der Praxis kommt es auch vor, dass Modelle mehrere Startereignisse und mehrere Endereignisse haben.

Um die Ausdrucksfähigkeit nicht auf die Kontrollflussspezifikation zu beschränken, wurden die EPKs erweitert. Erweiterte EPKs (eEPKs) zeigen auch beteiligte Organisationseinheiten, Stellen oder Rollen (Organisationssicht), Ein- und Ausgabedaten (Datensicht), sowie die verwendeten Systeme (ein spezieller Aspekt der Funktionssicht) auf. Abbildung 2 zeigt ein eEPK-Beispiel für einen Teil des Prüfungsprozesses einer Hochschule.

Zur Beschränkung von Prozessen auf handhabbare Größen und auf überschaubare Ausschnitte bzw. Abstraktionsebenen werden Wertschöpfungskettendiagramme und Hinterlegungen verwendet. Wertschöpfungsketten abstrahieren EPKs und zeigen Abfolgen und Strukturen betrieblicher Funktionen auf höherer Ebene auf. Hinterlegungen sind Verfeinerungen von Funktionen, die diese mit den Mitteln der eEPK auf nächster Verfeinerungsstufe beschreiben.

Für die Nennung von Nachfolgeprozessen und Vorgängerprozessen in EPKs werden in der Praxis oft Prozessschnittstellen eingesetzt, die diese Prozesse symbolisieren. In manchen existierenden Prozessmodellen werden Prozessschnittstellen auch zur Darstellung von Unterprozessen eingesetzt. Diese Verwendung wird allerdings nicht empfohlen, da hierfür bereits das Konstrukt der Hinterlegung existiert. Generell sind Prozessschnittstellen

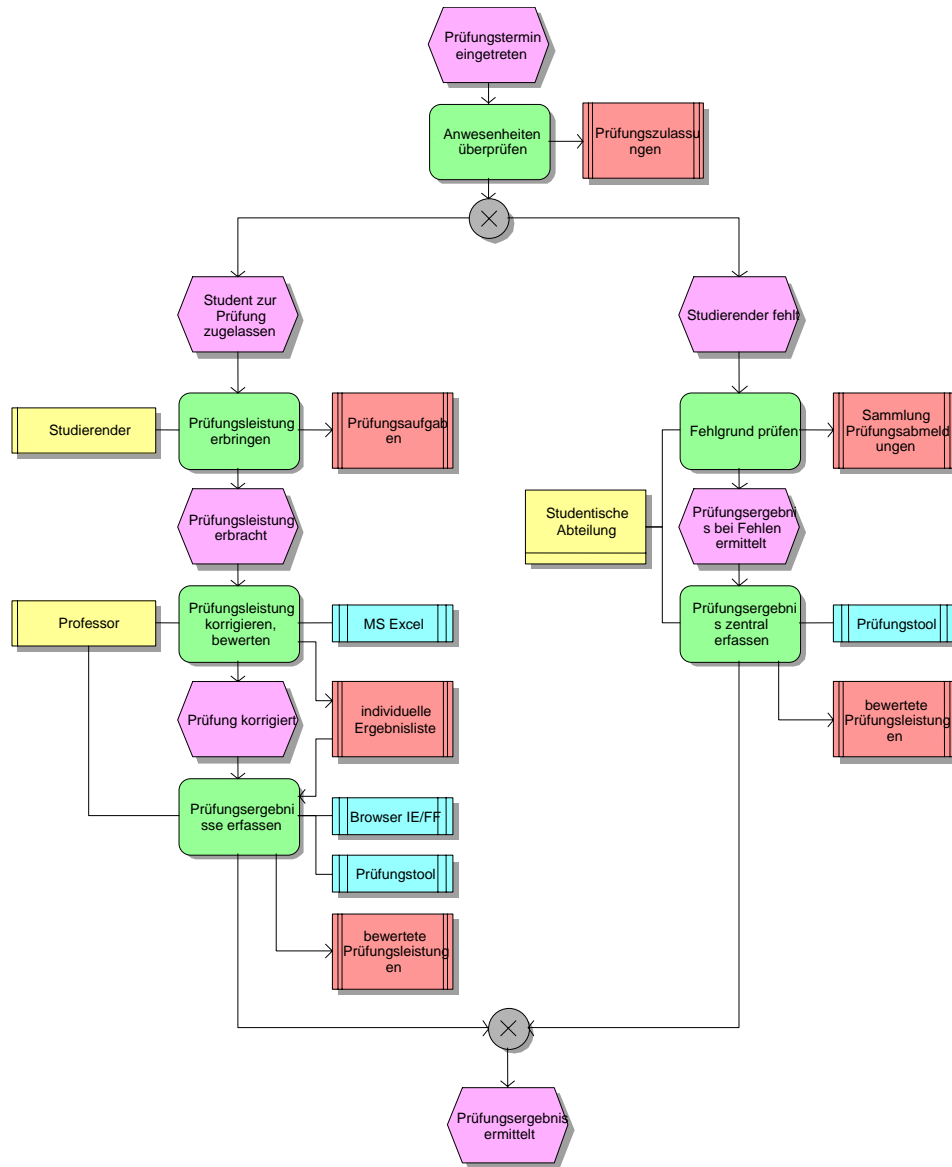


Abbildung 2: EPK-Beispiel

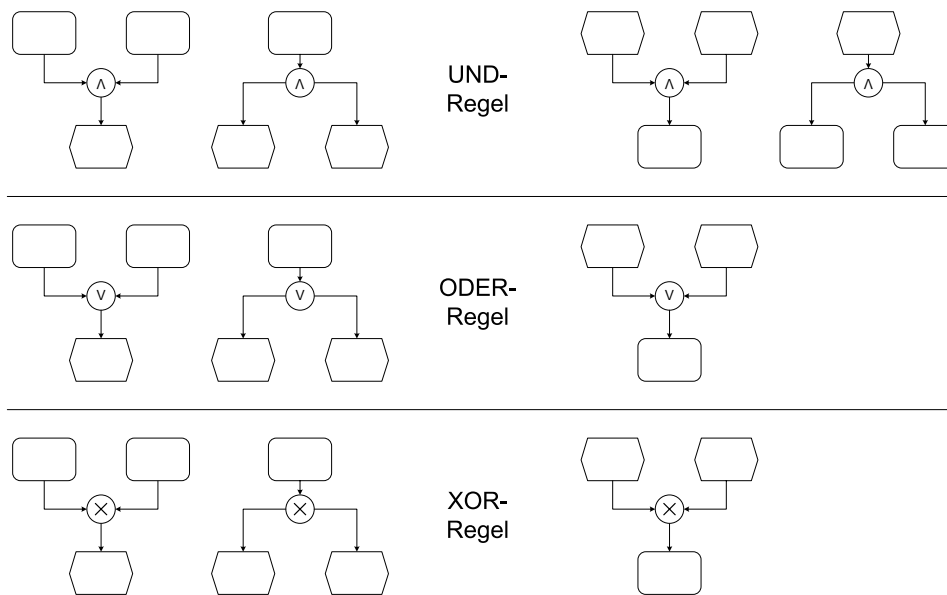


Abbildung 3: Verknüpfungsregeln nach [IDS06]

zurückhaltend einzusetzen, da sie Informationen redundant darstellen. Eine Definition des Konstrukts Prozessschnittstelle findet sich im Methodenhandbuch von ARIS [IDS06] nicht.

4 BPMN im Überblick

Der folgende Überblick über die Business Process Modeling Notation (BPMN) basiert auf der im Februar 2009 veröffentlichten Version 1.2 [bpm09].

Abbildung 4 zeigt die vier Hauptkategorien an BPMN-Konstrukten: Swimlanes, verknüpfende Objekte, Flussobjekte und Artefakte. Swimlanes repräsentieren Organisationseinheiten, Rollen oder Systeme. Eine Unterscheidung, ob es sich um eine Organisationseinheit oder eine Rolle handelt, wird nicht unterstützt. Die häufigste Verwendung ist dabei jedoch, dass Pools Organisationen darstellen und Lanes Organisationseinheiten darin.

BPMN unterscheidet Sequenzfluss und Nachrichtenfluss. Sequenzfluss stellt die Verhaltensabhängigkeiten innerhalb eines Pools dar, während Nachrichtenfluss die Kommunikation verschiedener Pools miteinander repräsentiert. Nachrichtenfluss innerhalb des gleichen Pools ist nicht erlaubt.

Es gibt drei Arten von Flussobjekten in BPMN: Aktivitäten repräsentieren die Arbeitsschritte innerhalb eines Prozesses. Diese sind entweder atomar (Tasks) oder können in Unterschritte (Unterprozesse) weiter unterteilt werden.

Ereignisse gibt es in den Ausprägungen "auslösend" und "eintretend". Bei auslösenden Er-

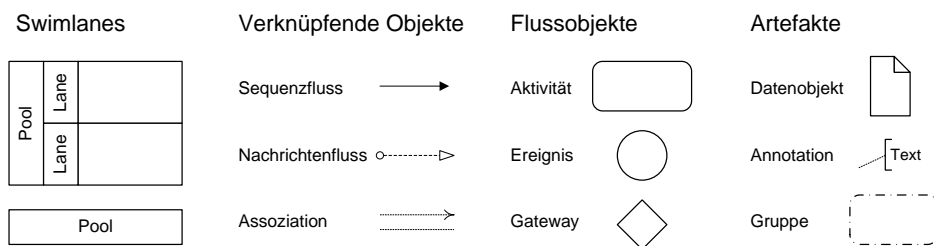


Abbildung 4: BPMN Modellierungskonstrukte

eignissen handelt es sich um spezielle Aktionen, z.B. werden Nachrichten an einen anderen Pool verschickt. Bei eintretenden Ereignissen wird der Prozessfluss solange blockiert, bis das entsprechende Ereignis eingetreten ist, z.B. bis eine Nachricht eingetroffen ist oder bis eine Zeitbedingung erreicht wurde.

Gateways werden für die Spezifikation von Kontrollflusslogik verwendet. Hier gibt es die fünf Typen daten-basiert exklusiv, ereignis-basiert exklusiv, parallel, inklusiv und komplex. Auf ein ereignis-basiertes Gateway folgt immer eine Reihe von eintretenden Ereignissen. Wird das Gateway erreicht, blockiert der Prozessfluss und es wird auf die Ereignisse gewartet. Sobald ein Ereignis eingetreten ist, wird nicht weiter auf die anderen Ereignisse gewartet und der Prozessfluss setzt sich an der Verzweigung fort entsprechend dem eingetretenen Ereignis. Dieses Verhalten wird im Workflow-Patterns-Katalog als “Deferred Choice” bezeichnet [AtHKB03].

BPMN bietet einige besondere Kontrollflusskonstrukte, wovon wir zwei ausgewählte kurz erwähnen. (1) Es können “Multiple Instanzen” spezifiziert werden, bei denen eine Menge von Instanzen von der gleichen Aktivität parallel ausgeführt werden können. Beispielsweise kann so spezifiziert werden, dass für jeden Bestellposten innerhalb einer Bestellung eine bestimmte Aktion nötig ist. (2) Weiterhin können mittels angehefteter Zwischen-Ereignisse Abbruchkriterien für Aktivitäten spezifiziert werden. Zum Beispiel können Eskalationsmaßnahmen definiert werden für den Fall, dass eine Aktivität nicht innerhalb einer bestimmten Zeit abgeschlossen wird.

Bei BPMN ist es möglich, mehrere Ende-Ereignisse zu verwenden. Abgesehen vom speziellen Terminierungs-Ende-Ereignis, bei dem die laufende Prozessinstanz als Ganzes abgebrochen wird, läuft eine Prozessinstanz solange weiter, bis keine Aktion mehr möglich ist. Dies wird auch “Implicit Termination” genannt [AtHKB03].

Abbildung 5 zeigt das Beispiel aus dem vorigen Abschnitt in BPMN-Darstellung.

5 Transformation von EPK zu BPMN

Dieser Abschnitt untersucht die Transformation von EPK zu BPMN. Dabei sind für jedes Konstrukt jeweils zwei Aspekte zu prüfen.

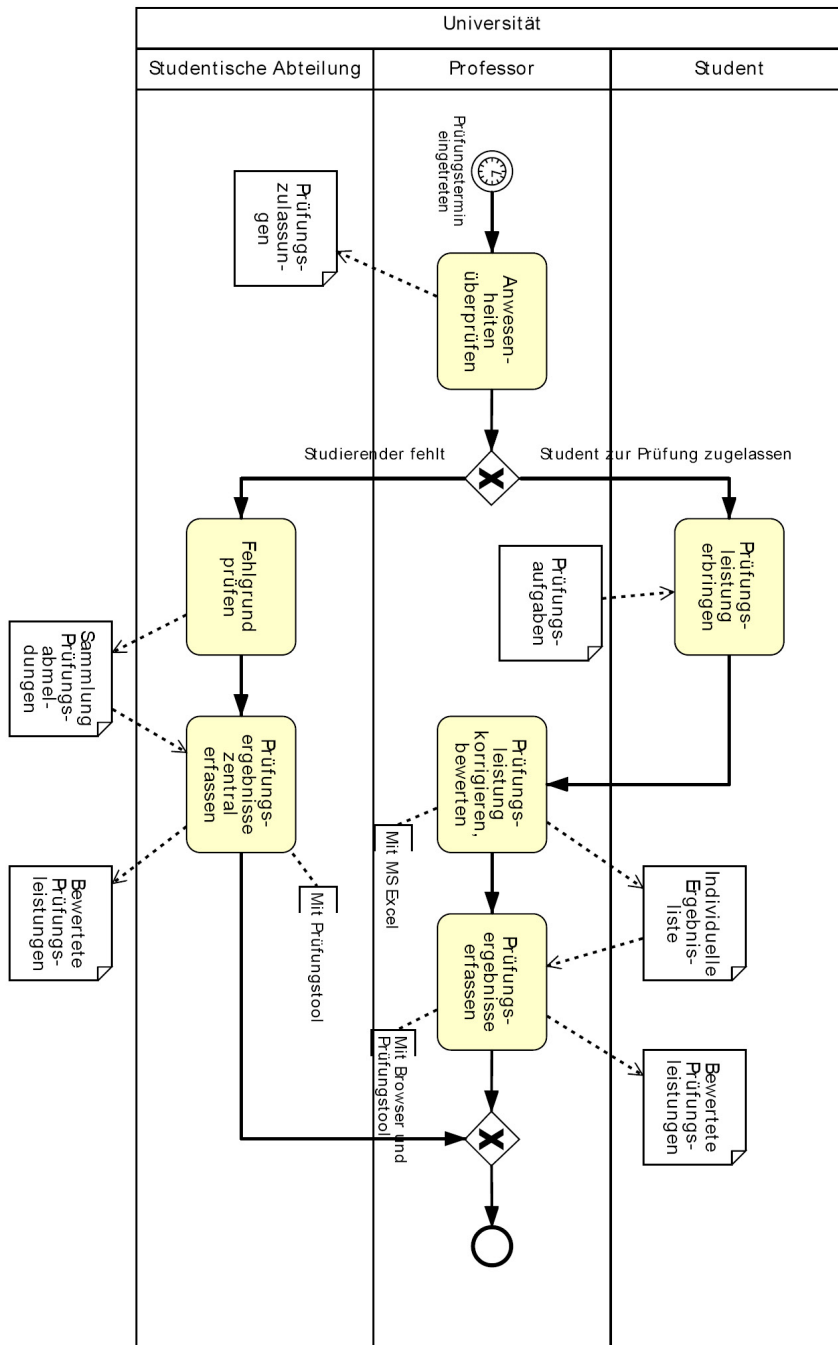


Abbildung 5: BPMN-Beispiel

1. Existiert ein BPMN-Konstrukt (oder eine Kombination von BPMN-Konstrukten), das die Semantik des EPK-Konstruktes abdeckt? Falls nicht, welche Konstrukte kommen der Semantik möglichst nahe?
2. Ist die Abbildung injektiv, d.h. werden keine zwei EPK-Konstrukte auf das gleiche BPMN-Konstrukt abgebildet? Falls eine nicht-injektive Abbildung vorliegt, handelt es sich um einen echten Informationsverlust?

Beide Fragen beschäftigen sich damit, ob der Informationsgehalt aus der EPK vollständig erhalten werden kann. Dies ist wichtig, da Modelle vor allem zur Beantwortung von Fragen dienen. Gehen Informationen verloren, kann dies zur Folge haben, dass bestimmte Fragen im Zielmodell nicht im gleichen Maße beantwortet werden können, wie dies im Quellmodell der Fall war.

Abbildung 6 zeigt eine tabellarische Übersicht über alle betrachteten EPK-Konstrukte und deren Entsprechungen in BPMN.

5.1 Funktionen

Funktionen repräsentieren einen Arbeitsschritt innerhalb des Prozesses und finden somit in BPMN-Tasks ihre Entsprechung. Ein Sonderfall ergibt sich im Falle von Hinterlegungen. Ist der Funktion ein Prozessmodell hinterlegt, so entspricht dies einem zugeklappten Subprozess in BPMN, der auf ein weiteres Diagramm zeigt.

Aktionen werden in BPMN nicht ausschließlich über Tasks und Subprozesse abgebildet. Auslösende Ereignisse repräsentieren kurzlebige Aktionen innerhalb eines Prozesses. Somit erlaubt BPMN eine weiterführende Typisierung von Aktionen. In ausgewählten Fällen können daher Funktionen mit auslösenden Zwischen- und Ende-Ereignissen übersetzt werden. Beispielsweise kann eine Funktion "Bescheid verschicken" als Zwischen- oder Ende-Nachrichten-Sende-Ereignis dargestellt werden. Eine detaillierte Diskussion zu dieser Unterscheidung ist in [HBS07] zu finden.

5.2 Ereignisse

Ereignisse in EPKs repräsentieren zumeist Zustände. So ist es ein typisches Beispiel, dass die Funktion "Prüfungsleistung erbringen" von einem Ereignis "Prüfungsleistung erbracht" gefolgt wird. In diesem Fall trägt das Ereignis keine nennenswerte Mehrinformation und scheint nur ein syntaktisches Füllelement zu sein. In BPMN ist eine explizite Darstellung von Prozesszuständen im Regelfall nicht vorgesehen. Der Fokus liegt vielmehr auf den Aktivitäten. Daher werden die meisten EPK-Ereignisse in einem Mapping auf BPMN ignoriert.

Soll ein Prozesszustand zwingend dargestellt werden, so stehen in BPMN Datenobjekte mit Zustandsattributen zur Verfügung. Ein Datenobjekt "Akte" könnte sich dabei im Zustand

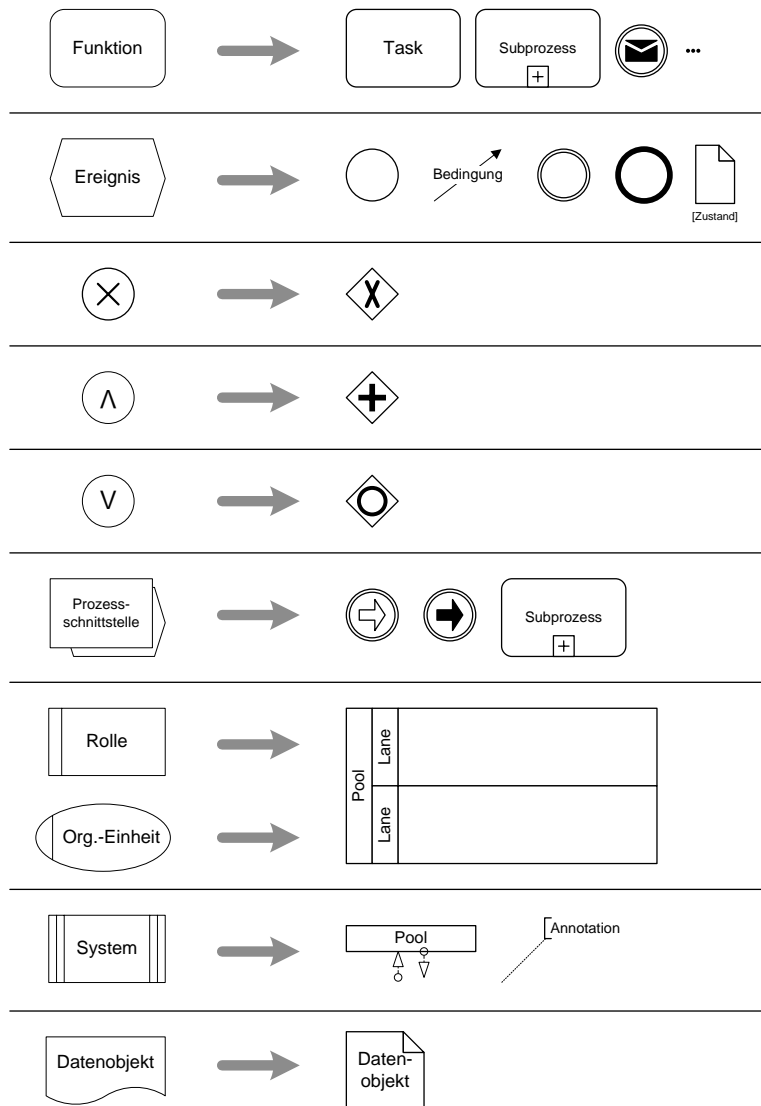


Abbildung 6: Transformation von EPK nach BPMN – Übersicht

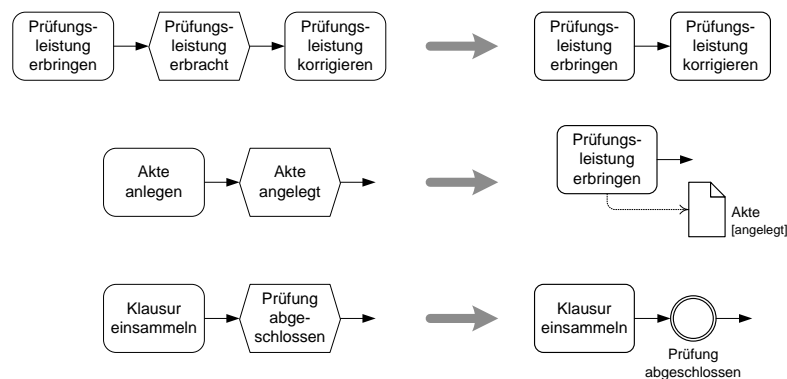


Abbildung 7: Abbildung von Zwischenereignissen

“angelegt” befinden. Als Alternative für die Darstellung von Prozesszuständen bieten sich weiterhin Blanko-Zwischenereignisse an. Diese Verwendung ist in der Praxis jedoch selten anzutreffen. Ein guter Hinweis darauf, dass Prozesszustände tatsächlich eine wichtige Information im Modell darstellen, besteht in folgendem Fall: Eine Funktion wird gefolgt von einem XOR- oder UND-Konnektor, dieser von einer Reihe von Ereignissen und diese wiederum von einem weiteren XOR- oder UND-Konnektor, der den Prozessfluss wieder zusammenführt. Hier haben die Konnektoren keinerlei Einfluss auf den Prozessfluss. Vielmehr findet hier eine explizite Aufzählung von möglichen Prozesszuständen statt. Zu diskutieren wäre hier allerdings, ob es sich hier um guten Modellierungsstil handelt, da die Aufspaltung im Kontrollfluss zu keinem Unterschied bzgl. der Funktionsausführungen führt.

Spannend wird es bei Startereignissen. Hier repräsentieren EPK-Ereignisse entweder Trigger oder Zustände [DM09]. Bei “Prüfungstermin eingetreten” handelt es sich um einen Trigger. In dem Moment, wo der Termin eintritt, wird der Prozess ausgelöst. In BPMN werden mehrere Typen von Triggern unterschieden. Der Typ kann lediglich durch die Beschriftung des Ereignisses oder durch den Prozesskontext ermittelt werden. Beispielsweise würde “Jeden Montagmorgen” auf ein zeitliches Ereignis hindeuten, “Zahlung ist eingegangen” auf eine eingehende Nachricht oder “Wasserstand hat kritisches Level erreicht” auf eine Business-Regel. Ist ein Ereignis-Typ nicht eindeutig auszumachen, so kann auch einfach ein untypisiertes Start-Ereignis (Blanko-Ereignis) verwendet werden.

“Kunde ist volljährig” ist ein Beispiel für eine Zustandsbeschreibung. Hier wird der Prozess nicht in dem Moment ausgelöst, wo der Kunde das 18. Lebensjahr vollendet. Eher ist diese Zustandsbeschreibung als Vorbedingung für die Prozessinstanziierung zu verstehen.

Sofern eine EPK nur über ein einziges Startereignis verfügt, ist eine Transformation offensichtlich: Im BPMN-Modell wird ein Startereignis erzeugt. Die einzige Frage ist, ob eine weitere Typisierung des Triggers sinnvoll ist.

Verfügt eine EPK über mehrere Startereignisse, so ist eine gültige Transformation nur in bestimmten Fällen möglich. Idealerweise handelt es sich um alternative Startereignisse, d.h. es erfolgt später eine Zusammenführung mittels XOR-Konnektoren. In diesem Fall

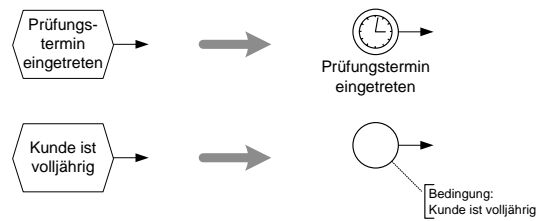


Abbildung 8: Abbildung von Startereignissen

kann jedes EPK-Startereignis auf ein BPMN-Startereignis abgebildet werden. Ein Beispiel für alternative Eintrittspunkte in einen Vertriebsprozess wären “Kunde ruft an” vs. “Messekontakt wurde protokolliert”.

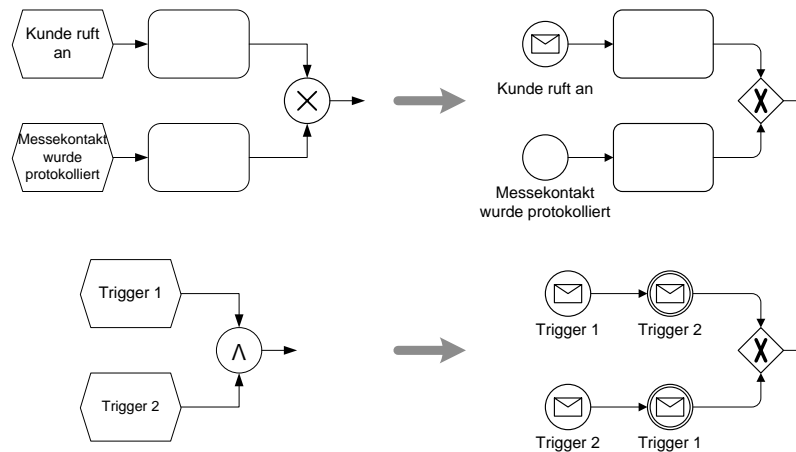


Abbildung 9: Abbildung von verknüpften Startereignissen

Sofern es sich nicht um alternative EPK-Startereignisse handelt, so ist anhand der Beschriftungen zu prüfen, ob es unter den Startereignissen mehr als einen Trigger gibt. Ist dies nicht der Fall, so wird das triggernde EPK-Startereignis wiederum mit einem BPMN-Startereignis übersetzt. Gibt es mehrere Trigger, die später über UND- oder ODER-Konnektoren zusammengeführt werden, ist eine korrekte Transformation in ein BPMN-Modell unter Beibehaltung einer vergleichbaren syntaktischen Struktur nicht möglich. Die parallele Struktur muss entsprechend sequenzialisiert werden, was eine Duplizierung von Ereignissen im Diagramm zur Folge hat (siehe Abbildung 9) und das Diagramm schwer lesbar macht. Eine detaillierte Diskussion zu diesem Fall findet sich bei Decker und Mendling [DM09].

BPMN-Prozessmodelle schließen mit Ende-Ereignissen ab. Im häufigsten Fall wird ein Blanko-Ende-Ereignis verwendet, welches optional mit einer kurzen Zustandsbeschreibung versehen ist, z.B. “Vertrag ist abgeschlossen”. Daher können EPK-Endereignisse einfach in BPMN-Endereignisse übersetzt werden.

An dieser Stelle sei darauf hingewiesen, dass Zwischen- und Endereignisse auch in bestimm-

ten Fällen auf Nachrichten-Empfangs-Ereignisse abgebildet werden können [KWL09].

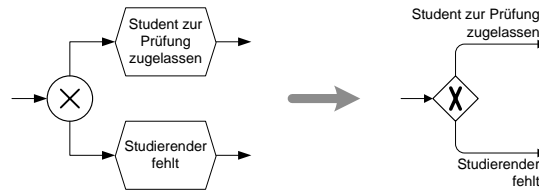


Abbildung 10: Abbildung von Ereignissen (4)

Bei denjenigen EPK-Ereignissen, die weder Start- noch Endereignisse sind, d.h. eingehende oder ausgehende Kontrollflusskanten besitzen, gibt es einen speziellen Fall, in dem das Ereignis auf jeden Fall in der Transformation beachtet werden muss: Entscheidungen werden in EPK über XOR- und ODER-Konnektoren mit mehreren ausgehenden Kanten modelliert. In diesem Fall geben die nachfolgenden Ereignisse die Verzweigungsbedingungen an, z.B. “Student zur Prüfung zugelassen”. Verzweigungsbedingungen werden in BPMN über Kantenbeschriftungen realisiert.

5.3 Konnektoren

Konnektoren realisieren Kontrollflussbeziehungen in EPKs. XOR-Konnektoren finden in daten-basierten exklusiven Gateways ihre Entsprechung, UND-Konnektoren in parallelen Gateways und ODER-Konnektoren in inklusiven Gateways. Während bei der semantischen Entsprechung im Falle von XOR- und UND-Konnektoren mit den entsprechenden BPMN-Konstrukten keine Zweifel bestehen, so ist der ODER-Konnektor insofern problematisch, da sowohl auf EPK- als auch auf BPMN-Seite Unsicherheit bzgl. der Semantik besteht (siehe Abschnitt 2). Praktische Relevanz haben die verschiedenen Nuancen in der Ausführungssemantik von ODER-Konnektoren / inklusiven Gateways aber wahrscheinlich nicht: Es besteht die Vermutung, dass für die allermeisten Prozessmodelle diese Nuancen keinen Unterschied in der Semantik bedeuten. Ein entsprechender empirischer Nachweis fehlt in der Literatur allerdings noch.

Zu beachten bei einem Mapping von Konnektoren ist lediglich, dass die Verzweigungsbedingungen in EPKs in Form von Ereignissen gegeben sind, während BPMN hierfür Kantenbeschriftungen vorsieht. Darüber hinaus sieht BPMN alternative Darstellungen von XOR-Joins, AND-Splits und OR-Splits vor: Hier werden mehrere Sequenzflusskanten mit Flussobjekten verbunden. Beispielsweise haben mehrere eingehende Kanten in einen Task XOR-Semantik. Eine Abbildung auf BPMN-Gateways, wie oben beschrieben, kann jedoch als üblicher Weg verwendet werden.

5.4 Prozessschnittstellen

Prozessschnittstellen bieten die Möglichkeit, ein EPK-Modell mit anderen Modellen in Beziehung zu setzen. Hier gibt es jedoch zwei semantische Ausprägungen, die jeweils unterschiedlich in BPMN abgebildet werden müssen. Zum einen werden Prozessschnittstellen verwendet, um Prozesshierarchien darzustellen. Hierbei verweist eine Prozessschnittstelle auf ein Modell als Ganzes oder auf ein Startereignis in einem anderen Modell. Die Semantik hierbei ist, dass der zweite Prozess beendet sein muss, bevor die auf die Prozessschnittstelle folgenden Funktionen angestoßen werden. Die Entsprechung auf BPMN-Seite wäre damit ein Unterprozess.

Alternativ realisieren Prozessschnittstellen zuweilen jedoch auch einfache Zerschneidungen von Prozessmodellen. Hierbei wird auf eine Prozessschnittstelle in einem anderen Modell verwiesen. Semantisch ist ein Pärchen von Link-Ereignissen also einer Kontrollflusskante, sofern die beiden Modelle in einem großen Modell zusammengeführt werden. BPMN sieht Link-Ereignisse zur Darstellung solcher Modell-Zerschneidungen vor.

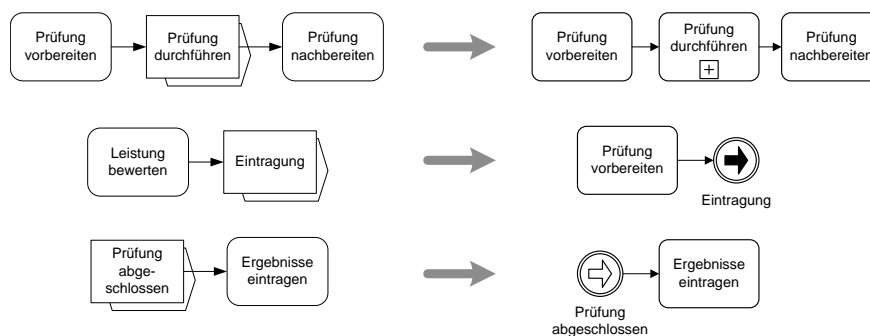


Abbildung 11: Abbildung von Prozessschnittstellen

Bei der Wahl, ob nun ein Unterprozess oder ein Link-Ereignis die passende Entsprechung ist, helfen eine Reihe von Indizien. Besitzt eine Prozessschnittstelle Vorgänger- und Nachfolgerelemente, so handelt es sich um einen Unterprozess. Wird eine Prozessschnittstelle jedoch als Eintrittspunkt in einen Prozess oder als Austrittspunkt verwendet, so sind wahrscheinlich Link-Ereignisse die passende Entsprechung. Informationen darüber, was von der Schnittstelle verlinkt wird (ganzes Modell vs. Prozessschnittstellen) gibt einen zusätzlichen Hinweis.

5.5 Organisationseinheiten / Rollen

Organisationseinheiten und Rollen werden in BPMN durch Pools und Lanes dargestellt. Anders als in EPK, wo durch Enthaltenseinsbeziehungen von Pools, Lanes und Sub-Lanes eine hierarchische Beziehung ausgedrückt werden kann, werden Organisationseinheiten und Rollen in EPKs nicht weiter in Beziehung gesetzt. Dies geschieht zumeist außerhalb

des Modells in separaten Organigrammen. Da weiterhin keine Unterscheidung zwischen Sequenz- und Nachrichtenfluss in EPKs zu finden ist, können die Organisationseinheiten und Rollen am besten als nebeneinander angeordnete Lanes des gleichen Pools abgebildet werden. Ein Bezeichner für diesen Pool steht nicht zur Verfügung. Als Behelfslösung können generische Namen wie “Hauptpool” oder “Organisation” gewählt werden.

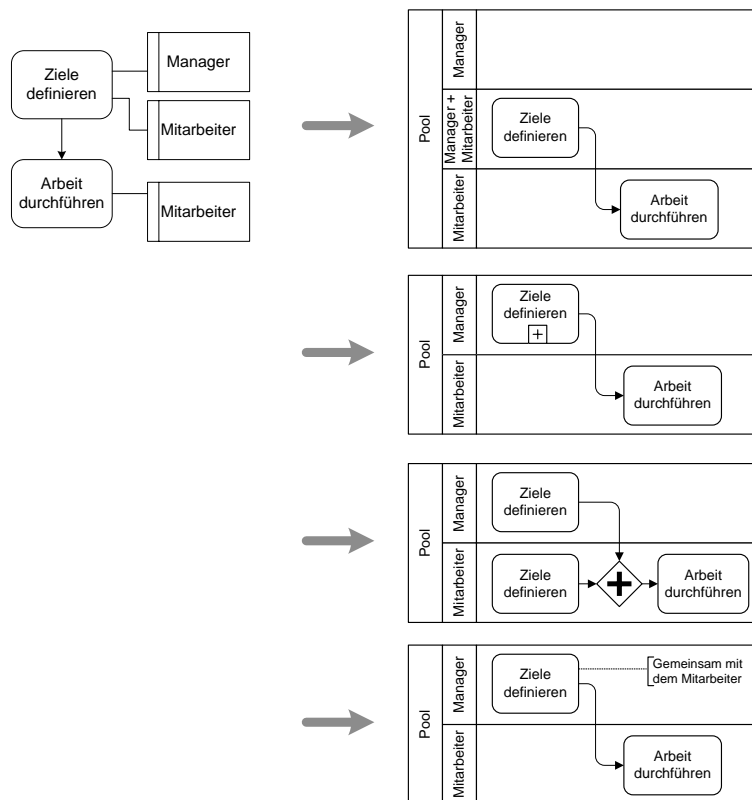


Abbildung 12: Abbildung von Rollen (und Organisationseinheiten)

Problematisch wird es, sobald mehrere Organisationseinheiten und/oder Rollen mit einer Funktion verknüpft sind. Dieser Sachverhalt kann in BPMN nicht sauber abgebildet werden. In BPMN-Diagramme sieht man des öfteren, dass eine Aktivität mehrere Lanes abdeckt. Dies ist von BPMN syntaktisch allerdings nicht vorgesehen. Verschiedene Workarounds sind hier in BPMN denkbar. (1) Zum einen können separate Lanes geschaffen werden, die explizit die Zusammenarbeit mehrerer Organisationseinheiten oder Rollen modellieren, z.B. “Team”. (2) Zum anderen ist es möglich, Unterprozesse zunächst einer bestimmten Lane zuzuordnen und bei einer Verfeinerung in einem separaten Diagramm die Aufgaben entsprechend den verschiedenen Lanes zuzuordnen. (3) Manchmal sieht man, dass eine gemeinschaftliche Aktivität dupliziert wird und mit gleicher Bezeichnung in verschiedenen Lanes erzeugt wird. (4) Die Aktivität wird nur einer Lane zugeordnet, und zwar der verantwortlichen oder führenden Organisationseinheit/Rolle, und über entsprechende

Textannotationen (Kommentare) wird die Beteiligung der anderen erwähnt. Abbildung 12 illustriert diese vier Strategien.

5.6 Systeme

Systeme werden typischerweise auf eine von vier Arten abgebildet. (1) Sofern das Modell eine reine Systemsicht liefert, d.h. Organisationseinheiten und Rollen kommen im Modell nicht vor, so können Systeme als Pools und Lanes dargestellt werden. Hier stellt sich wiederum die gleiche Frage wie bei Organisationseinheiten und Rollen: Wie kann dargestellt werden, dass mehrere Systeme einer Funktion zugeordnet sind. Schwieriger wird es außerdem, falls sowohl Systeme als auch Organisationseinheiten / Rollen in einem Modell definiert sind. (2) Alternativ können Systeme als zugeklappte Pools dargestellt werden, mit denen der Prozess Nachrichten austauscht. Von der Intuition her würde so z.B. repräsentiert, dass ein Mensch im Laufe einer Aktivität mit dem System über Eingabemasken und entsprechende Bildschirmanzeigen interagiert. (3) Als dritte Option kann ein Modellierer auch auf die Darstellung von Systemen in einem Modell verzichten. Mit Hilfe weiterer Modelle, die sich dann speziell mit einer Systemzuordnung beschäftigen, kann die Systemsicht wiederum abgedeckt werden. Es entstehen somit mehrere Perspektiven auf den gleichen Prozess (organisatorische Sicht vs. IT-Sicht). (4) Abschließend, und wahrscheinlich am häufigsten verwendet, können auch Textannotationen verwendet werden, um die Systemzuordnung widerzuspiegeln. Diese vier Optionen sind in Abbildung 13 illustriert.

5.7 Datenobjekte

Für Datenobjekte gibt es eine offensichtliche Abbildung von EPKs auf BPMN: Über BPMN-Datenobjekte und entsprechend gerichtete Assoziationen kann die EPK-Semantik direkt abgebildet werden.

6 Diskussion

Im vorigen Abschnitt wurden alternative Optionen vorgestellt, wie die verschiedenen EPK-Konstrukte abgebildet werden können. Die Wahl einer geeigneten Option kann bei einer Migration einer größeren Anzahl von Modellen oft ein einziges Mal für alle Modelle entschieden werden. Dies ist vor allem bei der Abbildung von mehreren Rollen / Organisationseinheiten oder bei der Abbildung von Systemen der Fall. Auch könnte z.B. entschieden werden, dass Ereignisse wenn immer möglich in einer Abbildung ignoriert werden.

Auch kann bei einer Abbildung behilflich sein, wenn die EPKs definierten Modellierungsrichtlinien folgen. Dies kann z.B. klären, wie Prozessschnittstellen bei einer Abbildung zu behandeln sind.

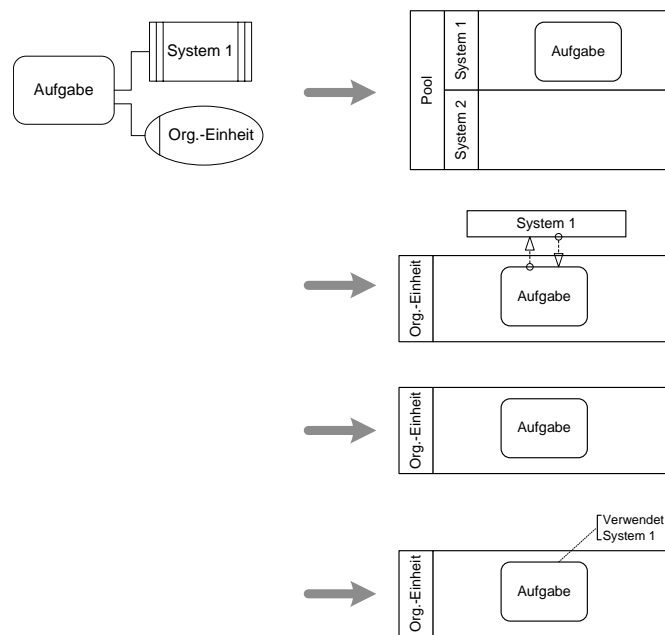


Abbildung 13: Abbildung von Systemen

In der ARIS-Methode sind EPKs nur eine von zahlreichen Diagrammtypen. Weitere häufig verwendete Diagrammtypen sind z.B. Wertschöpfungsketten und Organigramme. Organigramme können in BPMN nicht dargestellt werden. Zwar bietet BPMN die Möglichkeit, Hierarchien an Lanes zu definieren. Dies deckt jedoch nicht die Ausdrucksmächtigkeit von Organigrammen ab. Wertschöpfungskettendiagramme, in denen Prozesse landkartenartig aufgezählt werden, können in BPMN insofern abgebildet werden, als dass eine Menge von Subprozessen nebeneinander im Diagramm platziert werden.

Dieser Beitrag beschäftigt sich lediglich mit dem unidirektionalen Mapping von EPK nach BPMN. Es wird keine Antwort darauf gegeben, wie ein BPMN-Diagramm in eine EPK überführt werden kann. Ein Round-Tripping, bei dem das Modell in der Zielsprache editiert wird und wieder zurück in die Quellsprache überführt wird, bleibt somit erst recht undiskutiert.

Auch beantwortet dieser Artikel nicht die Frage "Wann setze ich welche Sprache ein?", bzw. die Frage nach Vor- und Nachteilen der beiden Sprachen in den unterschiedlichen Phasen des Entwurfs prozessorientierter Informationssysteme. Diese Fragen werden zu Beginn von BPM-Initiativen häufig gestellt. Dieser Beitrag hilft bei dieser Diskussion insofern, als dass er einen ersten Anhaltspunkt dafür bietet, wie im Falle der Entscheidung, zunächst mit EPKs zu arbeiten und später auf BPMN umzusteigen, hinterher eine Migration der Modelle aussehen kann.

Dieser Beitrag hat sich auf BPMN in der Version 1.2 konzentriert. Im Laufe des Jahres 2010 kann mit der Veröffentlichung der neuen Version 2.0 gerechnet werden. Die allermeisten

Konvertierungsregeln bleiben davon unberührt. Lediglich bei der Darstellung von Systemen erleichtert BPMN 2.0 eine Abbildung. Durch “Data Stores” sowie über selbst definierbare Artefakte können Systeme damit abgebildet werden.

7 Zusammenfassung und Ausblick

Dieser Beitrag hat gezeigt, dass weite Teile der EPKs automatisch bzw. semi-automatisch in bedeutungsgleiche BPMN-Modelle überführt werden können. Eine vorherige sachkundige Qualitätssicherung der EPKs sowie geeignete Modellierungskonventionen, die wiederum automatisch (z.B. mit sog. Reports) überprüft werden können, erhöhen zusätzlich das Einsatzpotenzial der automatischen Umstellung. Durch dieses Vorgehen können die in EPKs getätigten Investitionen gesichert werden, wenn das Anwenderunternehmen von EPKs auf BPMN wechseln möchte oder muss.

Eine Organisation sollte jedoch prüfen, ob nicht auch eine erneute BPMN-Modellierung, gepaart mit einer Anpassung und Aktualisierung der Modelle eine valide Alternative bildet. Dies hat gleichzeitig den Effekt, dass die Verwendung der BPMN eingeübt werden kann.

Eine Teilmenge der vorgestellten Konvertierungsregeln wurde im Rahmen der Signavio-Oryx Academic Initiative² als automatische Transformation realisiert. Dabei wird jede Funktion als Task, jedes Starterereignis als Blanko-Start-Ereignis und jedes Endereignis als Blanko-Ende-Ereignisse umgesetzt. Zwischenereignisse werden ignoriert – außer im Falle von Verzweigungsbedingungen. Prozessschnittstellen werden bislang nicht unterstützt und bei Rollen/Organisationseinheiten wird pro Funktion einfach nur eine Rolle/Organisationseinheit ausgewählt (und die anderen werden ignoriert). Als nächster Schritt wird eine semi-automatische bzw. durch Konventionen konfigurierbare Transformation realisiert, um den Entscheidungspunkten gerecht zu werden, die in diesem Beitrag beschrieben wurden.

Literatur

- [AtHKB03] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski und Alistair P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.
- [bpm09] Business Process Modeling Notation (BPMN), Version 1.2. Bericht, Object Management Group (OMG), Feb 2009. <http://www.omg.org/spec/BPMN/1.2/>.
- [CKS08] Mary Beth Chrissis, Mike Konrad und Sandy Shrum. *CMMI, Guidelines for Process Integration and Product Improvement, Second Edition*. Addison Wesley, 2008.
- [DGHW07] Marlon Dumas, Alexander Grosskopf, Thomas Hettel und Moe Wynn. Semantics of BPMN Process Models with OR-joins. In *Proceedings of 15th International Conference on Cooperative Information Systems (CoopIS)*, Jgg. 4803 of LNCS, Seiten 41–58, Vilamoura, Portugal, November 2007. Springer Verlag.

²See <http://www.signavio.com/academic>

- [DM09] Gero Decker und Jan Mendling. Process Instantiation. *Data & Knowledge Engineering*, 2009.
- [Gad08] Gadatsch. *Grundkurs Geschäftsprozess-Management, 5. Auflage*. Vieweg, 2008.
- [HBS07] Volker Hoyer, Eva Bucherer und Florian Schnabel. Collaborative e-Business Process Modelling: Transforming Private EPC to Public BPMN Business Process Models. In *Business Process Management Workshops*, Jgg. 4928 of LNCS, Seiten 185–196, Brisbane, Australien, 2007. Springer Verlag.
- [HW08] Stefan Huth und Thomas Wieland. *Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK*, Seiten 61–76. 2008.
- [IDS06] ARIS Platform, Methode ARIS 7.0. Bericht, IDS Scheer, Saarbrücken, 2006.
- [IT-05] COBIT 4.1. Bericht, IT-Governance Institute, 2005.
- [KMWL09] Oliver Kopp, Daniel Martin, Daniel Wutke und Frank Leymann. The Difference Between Graph-Based and Block-Structured Business Process Modelling Languages. *Enterprise Modelling and Information Systems*, 4(1):3–13, June 2009.
- [KWL09] Oliver Kopp, Matthias Wieland und Frank Leymann. External and Internal Events in EPCs: e2EPCs. In *2nd International Workshop on Event-Driven Business Process Management (edBPM09)*. Springer Verlag, 2009.
- [Men07] Jan Mendling. On the Detection and Prediction of Errors in EPC Business Process Models. *EMISA Forum*, 27(2):52–59, 2007.
- [Sch92] August-Wilhelm Scheer. *Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2. Auflage*. Berlin, 1992.
- [Sch98] August-Wilhelm Scheer. *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen, 3. Auflage*. Berlin, 1998.
- [SJ02] August-Wilhelm Scheer und W. Jost. *ARIS in der Praxis, Gestaltung, Implementierung und Optimierung von Geschäftsprozessen*. Berlin, Heidelberg, New York, 2002.
- [SKI08] Sebastian Stein, S. Kühne und K. Ivanov. Business to IT Transformations Revisited. In *MDE4BPM 2008*, 2008.
- [SKLN09] David Schumm, Dimka Karastoyanova, Frank Leymann und Jörg Nitzsche. On Visualizing and Modelling BPEL with BPMN. In *Proceedings of the 4th International Workshop on Workflow Management (ICWM2009)*. IEEE Computer Society, 2009.
- [VZHA05] D. Vanderhaeghen, S. Zang, A. Hofer und O. Adam. XML-based Transformation of Business Process Models–Enabler for Collaborative Business Process Management. *XML4BPM*, 2005.
- [WDGW08] Matthias Weidlich, Gero Decker, Alexander Grosskopf und Mathias Weske. BPEL to BPMN: The Myth of a Straight-Forward Mapping. In *Proceedings 16th International Conference on Cooperative Information Systems (CoopIS)*, Jgg. 5331 of LNCS, Seiten 265–282, Monterrey, Mexico, Nov 2008. Springer Verlag.

Evaluating the Effect of Feedback on Syntactic Errors for Novice Modellers

Ralf Laue

Chair of Applied Telematics/e-Business,* University of Leipzig
Klostergasse 3, 04109 Leipzig, Germany
laue@ebus.informatik.uni-leipzig.de

Stefan Kühne

Business Information Systems, University of Leipzig
Johannisgasse 26, 04103 Leipzig, Germany
kuehne@informatik.uni-leipzig.de

Andreas Gadatsch

University of Applied Sciences Bonn-Rhein-Sieg
Grantham-Allee 20, 53757 Sankt Augustin, Germany
Andreas.Gadatsch@fh-bonn-rhein-sieg.de

Abstract: In this paper, we present the results of a controlled human experiment where students in a business process modelling course had to model a business process from a case study as part of their coursework. One group could take advantage of the continuous validation feature that is implemented in the *bflow* modelling tool, i.e. they were provided with alerts about modelling errors. A control group had to create a model for the same case study without using continuous validation. The results of the experiment indicate that the presence of continuous validation indeed has had a positive effect on the number of syntactic errors in business process models.

1 Introduction and Related Work

Modern development environments like Eclipse or Visual Studio provide immediate feedback on syntactic and semantic errors by performing continuous parsing and compilation. This continuous compilation feature helps to detect coding errors very quickly and can reduce the time between making an error and fixing it.

An extension is continuous testing [Saf04] where regression tests are run in the background, providing the developer an instant feedback about test failures.

It has been shown experimentally that continuous compilation and continuous testing have a statistically significant effect on the success in completing a programming task [SE04, Saf04].

*The Chair of Applied Telematics/e-Business is endowed by Deutsche Telekom AG.

The idea of continuously validating the correctness has been adapted to the domain of graphical modelling as well. *ARGO/UML* [RR00] allows to run so-called *design critics* in background and can give suggestions for improvements. *MetaModelAgent* [BI06], which is an extension of *IBM Rational Software Architect*, contains a continuous model verification feature. With each alteration of the model, it is verified against several guidelines. Violations are reported, and suggestions for a correction are provided. *UMLAnalyzer* [Egy06] provides feedback on model consistency.

The aforementioned tools allow to check a model for meta-model compliance, violations of modelling style conventions and intra-model consistency.

Literature on experiments testing the usefulness of continuous validation for graphical models is rare. Dranidis [Dra07] showed that the consistency checking feature of the educational UML modelling tool *StudentUML* helps students in identifying and correcting their modelling mistakes.

Lange et al. [LBCD06] conducted an experiment in which they compared three groups of modellers: One group did not use modelling conventions, a second one used modelling conventions and a third group used a tool that checked adherence to modelling conventions. They found that the group using tool-supported modelling conventions had less defects, although the results were not statistically significant.

In this paper, we describe an experiment with a small group of business administration students. We gave the students the task to build a business process model from a case study. The language that had to be used for this purpose were Event Driven Process Chains. A short introduction into this language is given in Sect. 2. The modelling tool used for the experiment, *bflow Toolbox*, comes with a continuous validation feature which will be described in Sect. 3. We describe our experiment to evaluate the effect of using this feature in Sect. 4. The results of a group that used this feature and the results of a control group which modelled without continuous validation are given in Sect. 5 and statistically compared in Sect. 6. Afterwards, we discuss the findings in Sect. 7.

2 Event-Driven Process Chains

For our experiment, we used the modelling language Event Driven Process Chains (EPC). EPCs consist of functions (activities which need to be executed, depicted as rounded boxes), events (pre- and postconditions before / after a function is executed, depicted as hexagons) and connectors (which can split or join the flow of control between the elements). Arcs between these elements represent the control flow. The connectors are used to model parallel and alternative executions. There are two kinds of connectors: Splits have one incoming and at least two outgoing arcs, joins have at least two incoming arcs and one outgoing arc. AND-connectors (depicted as $\textcircled{\wedge}$) are used to model parallel execution. When an AND-split is executed, the elements on all outgoing arcs have to be executed in parallel. The corresponding AND-join connector waits until all parallel control flows that have been started are finished. XOR-connectors (depicted as $\textcircled{\times}$) can be used to model alternative execution: A XOR-split has multiple outgoing arcs, but only one of

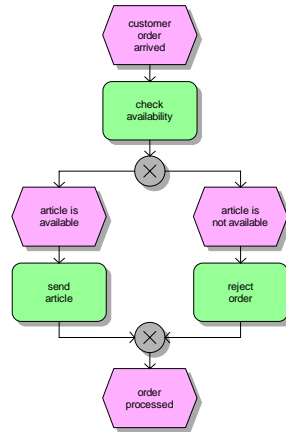


Figure 1: Simple Business Process modelled as EPC

them will be processed. A corresponding XOR-join waits for the completion of the control flow on the selected arc. Finally, OR-connectors (depicted as \odot) are used to model parallel execution along one or more control flow arcs. An OR-split starts the processing of one or more of its outgoing arcs. The corresponding OR-join waits until all control flows that have been started by the OR-split are finished.

The EPC elements described above are sufficient for modelling simple business processes like the following one: “When a request from a customer arrives, the availability of the product has to be checked. If it is available, the item will be sent; otherwise the customer will get a negative reply.” Fig. 1 shows this business process modelled as EPC diagram.

A detailed definition of the syntax of the EPC language can be found in [NR02, van99]. For the purpose of this article, we just mention the most important syntax requirements of EPC models: Functions have exactly one incoming and exactly one outgoing arc. Events have at most one outgoing and at most one incoming arc. Events without an incoming arc are called start events (like “customer order arrived” in Fig. 1), and events without an outgoing arc are called end events (like “order processed” in Fig. 1). Connectors can be either splits or joins as described above. Events and functions alternate along the flow of control, i.e. an event has to be followed by a function and vice versa. Finally, an event must not be followed by an XOR- or OR-split. The intention of this last rule is that the activity that lead to the (X)OR-decision should be modelled explicitly as a function.

3 Automatic Checks in bflow

For the experiment described in this paper, we used a localized German version of the open source modelling tool *bflow Toolbox 0.0.4*¹. Currently, this Eclipse-based tool supports

¹www.bflow.org

three graphical modelling languages, including EPCs. The version number 0.0.4 indicates that the development status of the tool is that of an alpha version. In many points, the usability of the tools still needs to be improved.

The *bflow Toolbox* comes with a feature called continuous validation. Checks for correct syntax, control-flow correctness (for example absence of deadlocks) and adherence to modelling conventions can be run in background. The user gets a feedback about errors and warnings. This feedback is shown by placing a marker at a modelling element which is related to the problem and by adding a textual information to the “Problems” view of the Eclipse IDE. By placing the cursor on the marker, the user can get additional information as shown in Fig. 2. This way, a problem is shown immediately without forcing the users’ attention away from the primary modelling task, as recommended in [McF02].

A detailed description of the continuous validation feature can be found in [KKGL08].

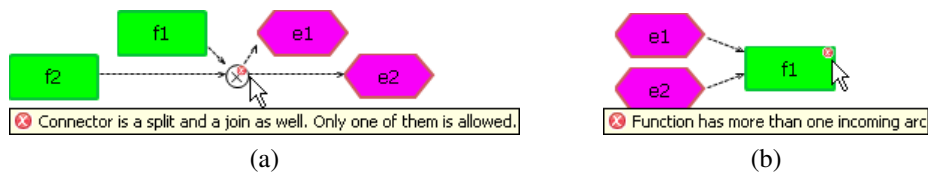


Figure 2: Feedback on two syntax errors

4 Experimental Design

The experiment was run at the the University of Applied Sciences Bonn-Rhein-Sieg, Germany. The participants were 15 business administration students. The students have selected business information systems as one of two fields of specialisation. So it can be assumed that they have an high affinity in busines process modeling like practitioners in real business process management projects. The students were familiar with modelling using Event Driven Process Chains from lectures they followed in the previous academic years. During those lectures, they had to draw models using pen and paper, but they did not have to model using an editor.

At the beginning of the experiment, we introduced the *bflow* editor and shortly recapitulated syntax and semantics of Event Driven Process Chains. We told the students that we will give them a case study for which they will be asked to create a business process model. We told them that the case study is similar to the ones they will be given at the written examination. Also, we told the students that the purpose of the experiment was to get feedback on the usability of the *bflow* editor and to collect suggestions for possible improvements (which was indeed the case, but is out of the scope of this paper).

We asked the students to fill a questionnaire, where we asked the following questions:

1. What is your age?
2. What is your sex?

3. What is your field of study, and how long have you been studying so far?
4. Is German your native language?
5. How familiar do you feel with modelling using Event Driven Process Chains? (allowing a choice between five answers from “very good” to “unfamiliar”)
6. How many business process models you have modelled so far using a graphical editor? (allowing the choice between “none”, “less than 5” and “more than 5”)
7. How familiar you are with Eclipse, regardless whether used as a programming environment or an editor for modelling? (allowing a choice between five answers from “very good” to “unfamiliar”)

Each of the 15 students filled the questionnaire. We exploited the answers to the questions 2, 4, 5, 6 and 7 for grouping the students into two groups: Group A had to create a model without continuous feedback on errors, group B could make use of this feature.

Among the 15 students, there were 13 male students whose native language was German and two female students whose native language was not German. As the ability to model correctly depended on understanding the case study (which was given in German and included domain-specific phrases from the field of mechanical engineering), the two latter students were placed in different groups.

We made the group assignments such that the groups were as similar as possible with regard to the other questions.

The students was given a case study - a hypothetical mechanical engineering business process. The case study has been constructed for the purpose of the experiment.

The students had to work individually. If there was a general question about the handling of the editor, the question has been answered by us. We did, however, not answer any questions about the case study.

While group A used the *bflow Toolbox* without the continuous validation feature, group B was provided with the same tool where continuous validation was running in the background.

The allotted time for solving the task was 45 minutes for each group. Students, who completed the model earlier, were told to write down the time needed.

We saved the models created during the experiment for being analysed later. Unfortunately, the result of one student of each group could not been saved because of a technical problem, i.e. we were able to collect 13 models only. However, this did not affect the considerations to make both groups as similar as possible with regard to the answers given on the questionnaire: In each group of the 13 students whose models have been collected was one student who has used Eclipse “a little bit” before, the others have never done so. Also, each group contained two students who rated their knowledge of the EPC modelling method as “moderate”, two students who rated it as “imperfect” and two students who rated it as “rather poor”. With one exception, the students have never used a tool for creating a business process model. The student who has already used a tool was also the only one who rated his knowledge of the EPC method as “good”. With the exception of this student, who was randomly assigned to group B, the groups were absolutely equal with respect to the answers to the questions on the questionnaire. The arithmetic mean of the

ages in group A was 25.2 years, and the arithmetic mean of the ages in group B was 26.9 years.

5 Quantitative Results

All students spent the given time of 45 minutes on modelling, no one finished the task earlier.

Because of the simplicity of the case study, no model contained one of the semantic errors (deadlocks etc.) that can be detected using the continuous validation feature. The syntactic errors that were made are shown in Tab. 1. Altogether, there were 24 such errors in 6 models in group A and 6 such errors in 7 models in group B.

model	without validation						with validation						
	1	2	3	4	5	6	7	8	9	10	11	12	13
event or function has more than one incoming or outgoing arcs	0	0	3	1	1	1	0	0	0	0	0	0	2
model starts with a function instead of an event	0	0	1	0	0	1	0	0	0	0	0	0	0
model ends with a function instead of an event	1	0	2	1	5	2	0	0	0	0	0	0	1
connector is both split and join	0	0	0	0	0	0	0	1	0	0	0	0	0
isolated element	0	0	2	0	0	0	1	0	0	0	0	0	0
event is followed by an (X)OR split	2	0	0	0	0	1	0	0	0	0	0	0	1
Sum	3	0	8	2	6	5	1	1	0	0	0	0	4

Table 1: Syntactic Errors Found in the Models

For the purpose of our comparison, we did not count two kinds of syntactic errors: One error we disregarded for the purpose of this experiment was the use of the wrong type of arc. *bflow* offers a dotted arc for modelling the control flow and a continuous arc for modelling the information flow. It is a weakness of the current version of *bflow* that selecting the wrong type of arc can happen very easily. The second error we did not count was when a function was followed by a function instead of an event. Strictly, this would be against the EPC syntax. However, the students have been told in the course before (in accordance with most textbooks on the subject) that in some cases such a construct is allowed. Without excluding both types of errors, there would have been 53 errors in 6 models in group A and 12 errors in the 7 models in group B.

We also evaluated the overall quality of the models under terms of regular exams. This was done by asking five university lecturers who are very familiar with business process modelling to give subjective marks for each model. They could choose from a range

model		without validation						with validation						
		1	2	3	4	5	6	7	8	9	10	11	12	13
manual	quality	3.6	1.6	4.0	2.4	3.6	3.6	4.2	3.2	4.6	2.0	2.5	2.3	3.0
rating														

Table 2: Marks given to the models

between mark 1 (“very good”) to mark 5 (“fail”). The averages of the marks given are shown in Tab. 2.

The mean grade for group A was 3.13; the mean grade for group B was 2.97.

6 Statistical Analysis

For comparing the number of errors made in each of the two groups, we performed the one-tailed Mann Whitney U test. The null hypothesis was that the results of both groups were drawn from the same population, i.e. the presence of the validation feature does not have an effect on the number of syntactic errors.

The group sizes were 6 and 7; and the value of U was 6. From this follows that the probability for the results from both groups belonging to the same population is $p=0.02275$. Using a significance level of 0.025 for the one-tailed test, we can conclude that the results from both groups are significantly different, i.e. the continuous validation feature has had a positive effect on the number of errors.

7 Discussion and Further Research Directions

An obvious weakness of our experiment was the small number of students taking part. Nevertheless, the results shown in Tab. 1 give raise to the assumption that the continuous validation feature indeed has a positive effect on the presence of errors. It has to be noted that the errors investigated in our experiment contributed only to a very small part of the model quality. The marks given to students in group B were only slightly better than the marks given to students in group A; the difference is in no way statistically significant.

The fact that only one student has used a business process modelling tool before limits the scope of the result. Most likely, more experienced modellers would not profit from finding rather trivial syntactic errors as much. Our result indicates that continuous validation for such syntactic errors is valuable at least for novice modellers.

Because of the time limitation, we could not test the effect of continuous validation on errors in larger and more complicated models. As such models are more likely to contain control-flow related errors (like deadlocks), we would expect that the ability of the validation in *bflow* to detect such errors as well would make it more helpful even for more experienced users.

Further experiments are needed with more participants and if possible with more complex models. It is our intention to repeat the experiment with a larger group of students. However, before doing so, we will make some improvements to the tool. For example, it seems to be less useful to create an alert when one draws two incoming arcs into an event or function. A much more straightforward action to prevent this kind of errors would be to forbid such constructs by restricting the meta-model.

References

- [BI06] Fredrik Bäckström and Anders Ivarsson. Verification and Correction of UML Models. 2006. Online; zuletzt abgerufen am 3. Juli 2009.
- [Dra07] Dimitris Dranidis. Evaluation of StudentUML: an Educational Tool for Consistent Modelling with UML. In *Proceedings of the Informatics Education Europe II Conference IEEII 2007*, 2007.
- [Egy06] Alexander Egyed. Instant consistency checking for the UML. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 381–390, New York, USA, 2006. ACM.
- [KKGL08] Stefan Kühne, Heiko Kern, Volker Gruhn, and Ralf Laue. Business Process Modelling with Continuous Validation. In *Business Process Management Workshops, BPM 2008 International Workshops, Milano, Italy, September 2008, Revised Papers*, volume 17 of *Lecture Notes in Business Information Processing*, pages 212–223. Springer, 2008.
- [LBCD06] Christian F. J. Lange, Bart Du Bois, Michel R. V. Chaudron, and Serge Demeyer. An Experimental Investigation of UML Modeling Conventions. In *MoDELS*, volume 4199 of *Lecture Notes in Computer Science*, pages 27–41. Springer, 2006.
- [McF02] Daniel C. McFarlane. Comparison of Four Primary Methods for Coordinating the Interruption of People in Human-Computer Interaction. *Human-Computer Interaction*, 17:63–139, 2002.
- [NR02] Markus Nüttgens and Frank J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen*, pages 64–77, 2002.
- [RR00] Jason E. Robbins and David F. Redmiles. Cognitive support, UML adherence, and XMI interchange in Argo/UML. *Information & Software Technology*, 42(2):79–89, 2000.
- [Saf04] David Saff. *Automatic continuous testing to speed software development*. PhD thesis, Massachusetts Institute of Technology, 2004.
- [SE04] David Saff and Michael D. Ernst. An experimental evaluation of continuous testing during development. In *ISSTA 2004, Proceedings of the 2004 International Symposium on Software Testing and Analysis*, pages 76–85, Boston, USA, July 12–14, 2004.
- [van99] Wil M. P. van der Aalst. Formalization and verification of event-driven process chains. *Information & Software Technology*, 41(10):639–650, 1999.

Überführung von EPK-Modellen in ausführbare Grid- und Cloud-Prozesse

Andreas Hoheisel¹, Thorsten Dollmann², Michael Fellmann³

¹ Fraunhofer-Institut für Rechnerarchitektur und Softwaretechnik (FIRST)
Kekuléstraße 7, 12489 Berlin
andreas.hoheisel@first.fraunhofer.de

² Institut für Wirtschaftsinformatik (IWi) im DFKI
Stuhlsatzenhausweg 3, 66123 Saarbrücken
thorsten.dollmann@iwi.dfki.de

³ Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung (IMU)
Lehrstuhl für Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49069 Osnabrück
michael.fellmann@uos.de

Zusammenfassung: Die Überführung von fachlichen Geschäftsprozessen in technische, ausführbare IT-Prozesse bleibt aufgrund unterschiedlicher Modellierungsansätze, Zielstellungen und Abstraktionsniveaus eine Herausforderung. Dieser Artikel beschreibt ein Vorgehen, mit dem fachliche Prozesse, die als Ereignisgesteuerte Prozessketten modelliert wurden, automatisch in Petrinetz-basierte, ausführbare Prozesse übersetzt werden können. Der Ansatz bietet darüber hinaus eine automatische Abbildung der Prozesse auf verteilte IT-Ressourcen im Rahmen von Service-Orientierten Architekturen, Cloud-Plattformen und Computing-Grids unter Berücksichtigung von Aspekten wie Lastausgleich und Skalierbarkeit.

1 Einleitung

Die Überführung von Geschäftsprozessen in technische Prozesse, welche IT-unterstützt in verteilten Systemen ausgeführt werden sollen, ist derzeit meist umständlich und nur zum geringen Teil automatisiert. Der Grund hierfür liegt insbesondere in der Kluft zwischen der oftmals informalen und abstrakten Modellierung von Geschäftsprozessen einerseits, und der formalen, mit konkreten technischen Details versehenen Modellierung von IT-Prozessen andererseits. Neben dem unterschiedlichen Abstraktionsniveau sind beide Arten der Prozessbeschreibung auch in der Regel organisatorisch unterschiedlich im Unternehmen verankert und unterscheiden sich in Bezug auf den Zeitpunkt ihrer Durchführung. So wird in den entsprechenden Softwareentwicklungsprojekten zum einen sehr früh und zum anderen sehr spät modelliert. Die frühe Modellierung umfasst die Spezifikation eines Systems aus fachlicher Sicht (z. B. Lastenheft). Die späte Modellierung definiert die Komponenten und deren Zusammenspiel aus einer technischen

Sicht und ist oft eng verbunden mit den zur Implementierung verwendeten Entwicklungsumgebungen. Durch die bestehende Lücke zwischen fachlichem und technischem Modell ist eine konsistente Überführung fachlicher Anforderungen in unterstützende IT-Systeme nicht gewährleistet. Die mangelnde Durchgängigkeit der Modellierung von der fachlichen zur technischen Ebene birgt – neben dem Risiko einer Mehrfachimplementierung von identischen Funktionalitäten, die aus fachlicher Sicht in verschiedenen Modellen unterschiedlich beschrieben werden – auch eine Synchronisationsproblematik zwischen fachlichem und technischem Prozessmodell. So spiegeln sich beispielsweise aufgrund betriebswirtschaftlicher Notwendigkeiten vorgenommene Ad-hoc-Änderungen an den Geschäftsprozessen zwar in den Implementierungsmodellen wieder, nicht aber in den fachlichen Modellen.

Weitere Unterschiede bestehen hinsichtlich des Gegenstandsbereichs. Während fachliche Prozessbeschreibungen beispielsweise Ziele und organisatorische Zuständigkeiten aufzeigen, sind für die technischen Prozesse die Datenflüsse und die Aufteilung von Ressourcen wie etwa Rechenleistung von zentraler Bedeutung. In der Summe führen die genannten Unterschiede und Probleme dazu, dass für die Modellierung von Geschäftsprozessen und technischen (ausführbaren) Prozessen in der Regel unterschiedliche und oftmals inkompatible Ansätze verwendet werden, die nur schwer zu überbrücken sind.

Unser Beitrag besteht darin, ein möglichst einfaches Vorgehen aufzuzeigen, um Geschäftsprozessmodelle in IT-Prozesse zu überführen, die in unterschiedlichen verteilten Systemen, wie zum Beispiel Service-Orientierten Architekturen (SOA), Cloud-Plattformen oder Computing-Grids zur Ausführung kommen können. Der vorgestellte Ansatz zeichnet sich *erstens* durch ein generatives Verfahren aus, bei dem ausgehend von einem fachlichen Prozessmodell vollautomatisch anhand der zuvor in der Workflow-Engine hinterlegten Konfigurationsparameter ein ausführbares Modell erzeugt wird. Zur fachlichen Modellierung wird hierbei die betriebswirtschaftlich orientierte, semiformale Modellierungssprache EPK (Ereignisgesteuerten Prozesskette) [KS92] verwendet, welche in Wissenschaft und Praxis weite Verbreitung erlangt hat. *Zweitens* zeichnet sich der Ansatz dadurch aus, dass die im Grid- und Cloud-Computing relevanten Aspekte wie Lastausgleich und Skalierung der Ressourcen berücksichtigt werden.

Der Artikel ist wie folgt aufgebaut. Zunächst werden im Abschnitt 2 Alternativen zu dem in diesem Beitrag vorgestellten Technologieansatz erläutert. Anschließend erfolgt eine Einführung in das Speicher- und Austauschformat EPML, das zur Repräsentation von auszuführenden EPK-Modellen genutzt wird. Auf die grundsätzlichen Besonderheiten hinsichtlich der Ausführung von Prozessen in verteilten Systemen geht anschließend Abschnitt 4 ein. Der beschrittene Weg der Transformation von EPK-Modellen in ausführbare Prozesse wird in Abschnitt 5 detailliert beschrieben und in Abschnitt 6 anhand einer Fallstudie beispielhaft veranschaulicht.

2 Verwandte Ansätze

Um dem im vorhergehenden Abschnitt beschriebenen Problem einer mangelnden Durchgängigkeit der Modelle von der fachlichen zur technischen Ebene zu begegnen, ist ein in Wissenschaft und Praxis vielfach reflektierter Weg die Verwendung von BPMN (Business Process Modeling Notation) [Ob09] zur Modellierung der Geschäftsprozesse und die anschließende Überführung in WS-BPEL (Web Services Business Process Execution Language) [Oa07] anhand der im Anhang A des BPMN-Standards enthaltenen Transformationsregeln. Eine Implementierung für eine Untermenge der durch die BPMN-Spezifikation definierten Sprachkonstrukte steht zum Beispiel durch das Projekt Babel zur Verfügung (<http://www.bpm.fit.qut.edu.au/research/projects/babel/>). Gleichwohl hiermit prinzipiell eine Durchgängigkeit im Sinne eines über alle Phasen des Entwicklungszyklus existierenden Prozessmodells erreicht wird, zeigt dieses Vorgehen jedoch für einige Anwendungsfälle gewichtige Nachteile:

- Sowohl WS-BPEL als auch BPMN haben eine relativ komplexe Semantik und sehr umfangreiche Syntax, was zu einer komplexen und fehlerträchtigen Transformation führt, die auch eine Überprüfung der Modelle hinsichtlich formaler Kriterien (Model Checking) erschwert. So besitzt WS-BPEL zum Beispiel drei Möglichkeiten um eine Schleife zu definieren (while, repeatUntil, forEach), obwohl aus technischer Sicht ein Schleifenkonstrukt ausreichen würde. Dies führt dazu, dass eine Überführung von BPMN-Modellen in ausführbare Modelle typischerweise manuelle Arbeitsschritte erfordert, die eine enge Kopplung zwischen Modellierung und Ausführung verhindern und folglich den Aufwand zur Unterstützung dynamischer Prozesse erhöhen.
- WS-BPEL ist ohne spezifische Erweiterungen wie BPEL-J auf die Orchestrierung von Web Services im Sinne von SOAP-Diensten spezialisiert und lässt sich nicht einfach auf andere Komponentenmodelle anwenden. Die Kapselung aller Komponenten als Web Services ist zwar für viele Anwendungsfälle ein gangbarer Weg, birgt jedoch zusätzliche Komplexität und Geschwindigkeitseinbußen. IT-Prozesse, die zum Beispiel überwiegend aus Datenbankaufrufen bestehen, lassen sich effektiver direkt über ODBC-Schnittstellen realisieren. Ein weiteres Beispiel sind rechen- oder datenintensive Prozesse, die auf der Ausführung von Kommandozeilenprogrammen in Multicore-, Cluster- oder Grid-Computing-Umgebungen basieren und schneller z.B. über POSIX-Schnittstellen oder dazwischen geschaltete Batch-Systeme (LSF, PBS etc.) ausgeführt werden können. Darüber hinaus muss eine nebenläufige Ausführung von Prozessen explizit in den BPEL-Prozessen vorgegeben werden, sie kann nicht automatisiert erfolgen.
- WS-BPEL bietet nur eingeschränkte Möglichkeiten, Prozesse unabhängig von der ausführenden Infrastruktur zu definieren. Diese Abstraktionsmöglichkeiten sind insbesondere in dynamischen verteilten Umgebungen notwendig (Cloud, Grid), da dort Teile der Infrastruktur während der Prozessausführung wegfallen, neu hinzukommen, oder fehlschlagen können. Zudem lässt sich durch abstrakte Prozesse eine bessere Wiederverwendbarkeit für unterschiedliche Infrastrukturen erzielen. So sollte

z. B. der gleiche Prozess für Tests zunächst lokal auf einem Laptop und später in Produktion auf einem großen Cluster ausgeführt werden können.

Im Unterschied zu einer Transformation von BPMN zu BPEL besitzt der von uns vorgeschlagene Ansatz eine vergleichsweise einfache Syntax und Semantik, die auf Ereignisgesteuerten Prozessketten (EPK) sowie Petrinetzen aufbaut, die beide (im Gegensatz etwa zu BPMN und WS-BPEL) theoretisch bereits umfangreich erforscht worden sind. Die EPK wird hierbei zur fachlichen Prozessbeschreibung aufgrund ihrer weiten Verbreitung in Wissenschaft und Praxis und ihrer strukturellen Einfachheit und Ähnlichkeit zu Petrinetzen ausgewählt. Für die Beschreibung ausführbarer Prozesse wird die auf Petrinetzen basierende Sprache GWorkflowDL gewählt, die durch eine Abstraktionsschicht für IT-Prozesse in verteilten Rechnerarchitekturen eine wichtige Voraussetzung zur Unterstützung von Grid- und Cloud-Prozessen erfüllt. Somit ist der vorgeschlagene Ansatz nicht auf Web Services als Ausführungs-Technologie beschränkt und gestattet es, Prozesse unabhängig von der ausführenden Infrastruktur zu definieren.

Weitere Ansätze, die ebenfalls eine Verbesserung der Überführung von fachlichen Prozessbeschreibungen in ausführbare Prozesse intendieren, finden sich im Umfeld der *Semantic Web Services* (für eine Übersicht vgl. [CS05] sowie [Ca04]). Durch eine maschinenverarbeitbare, semantische Beschreibung von Web Services soll hierbei vor allem eine leichteres Auffinden (Discovery) und Auswählen (Selection) ermöglicht werden sowie eine Überbrückung semantischer Differenzen im Rahmen von Service-Kompositionen (Orchestration) erreicht werden. Hierzu werden allerdings umfangreiche ontologiebasierte Beschreibungen benötigt, wozu in der Vergangenheit eigene Ontologiesprachen und -Vokabulare wie WSMO (Web Service Modeling Ontology) [Ar05] und OWL-S (Web Ontology Language for Web Services) [Ma04] entwickelt worden sind.

Semantic Web Services können als eine komplementäre Technologie zu dem in diesem Beitrag beschriebenen Ansatz aufgefasst werden, da sie einerseits eine Überbrückung semantischer Differenzen zwischen verschiedenen Diensten fokussieren, die nicht das primäre Ziel des von uns vorgestellten Ansatzes ist. Andererseits werden von den genannten Technologien Aspekte wie Lastausgleich und Skalierung der Ressourcen nicht per se berücksichtigt, die jedoch ein grundlegender Bestandteil der hier vorgestellten Technologie sind.

3 EPML als Austauschformat für EPK-Modelle

Zum Austausch von Prozessmodellen, die mit Hilfe einer Ereignisgesteuerten Prozesskette beschrieben werden, hat sich die *Event-driven Process Markup Language (EPML)* durchgesetzt [MN04]. Wesentliche Merkmale der EPML ist die Graph-orientierte Repräsentation, die Trennung zwischen Definition und Ausprägung, die Erweiterbarkeit sowie die XML-basierte Notation, welche spezifisch als Austauschformat für EPK-Modelle entwickelt wurde. Eine graph-orientierte Repräsentation liegt vor, da die mit EPML beschriebenen EPK-Modelle als gerichteter Graph aufgefasst werden, dessen Knoten durch die Modellelemente eines EPK-Modells konstituiert werden und dessen Kanten

dem Kontrollfluss eines EPK-Modells entsprechen. Eine Trennung zwischen Definition und Ausprägung erlaubt in Analogie zu datenbankgestützten Modellierungswerkzeugen eine redundanzfreie Speicherung der Beschreibung von Modellinhalten wie etwa auszuführender Funktionen unabhängig von deren konkretem Auftreten (Ausprägung) in einem prozessualen Kontext. Eine Erweiterbarkeit von EPML ist über Attribute gegeben, die beliebige Werte aufnehmen können und im Kopfbereich eines EPML-Dokumentes deklariert werden. Die XML-basierte Notation gestattet eine Plattform- und sprachunabhängige Repräsentation von EPK-Modellen. Als Werkzeug zur Erzeugung von EPML wurde im Rahmen dieser Arbeit das am Institut für Wirtschaftsinformatik in Saarbrücken entwickelte Werkzeug CoMoMod eingesetzt.

4 IT-Prozesse in verteilten Systemen

Bei der IT-gestützten Automatisierung von Prozessen kommen zunehmend verteilte Systeme zum Einsatz, bei denen die unterschiedlichen Prozessaktivitäten nicht auf einem zentralen Server, sondern auf räumlich zum Teil weltweit verteilten Rechnern ausgeführt werden. Die Kommunikation zwischen diesen Rechnern erfolgt in der Regel über das Internet oder über lokale Netzwerke. Eine typische Realisierungsform eines solchen verteilten Systems ist eine *Service-orientierte Architektur (SOA)*, bei der die Dienste „top-down“ an den einzelnen Funktionalitäten der Geschäftsprozesse ausgerichtet werden und oftmals in Form von Web-Services auf räumlich verteilten Servern zur Verfügung gestellt werden.

Eine weitere Methode zur Realisierung von besonders skalierbaren Rechnernetzen zur Ausführung von IT-Prozessen sind sogenannte *Clouds*. Der Begriff *Cloud* wird im Rahmen dieses Artikels entsprechend der Definition von Forrester Research verwendet, wonach eine Cloud ein Pool aus abstrahierter, hochskalierbarer und verwalteter IT-Infrastruktur ist, die Kundenanwendungen vorhält und nach Verbrauch abgerechnet wird. Der Schwerpunkt von Cloud-Computing liegt im Bereich der ungekoppelten oder lose gekoppelten Massendienste. Spezielle Dienste, wie zum Beispiel parallele Simulationsrechnungen, welche eng gekoppelte Parallelrechner oder Computer-Cluster mit schnellen internen Netzwerkverbindungen benötigen, lassen sich derzeit hingegen besser auf eine Grid-Infrastruktur abbilden, wie sie in Deutschland zum Beispiel durch D-Grid (<http://www.d-grid.de/>) gegeben ist. Zudem werden beim Grid-Computing in der Regel komplexe virtuelle Organisationen besser unterstützt, welche einen Zusammenschluss von vielen Ressourcenbetreibern und Anwendern für die Lösung einer Aufgabe bezeichnen. Im Unterschied zu Clouds zielt Grid-Computing auf die transparente, gemeinsame Nutzung von Ressourcen in einem Netzwerk innerhalb einer Organisation, in Gruppen oder auch in weltweiten Verbänden ab. Zu den Ressourcen eines Grids zählen hierbei neben Rechen- oder Speicherressourcen z.B. auch Anwendungen, Software, Web-Services, Lizenzen, oder Sensoren zur Datenerfassung.

4.1 Grid Workflow Description Language (GWorkflowDL)

Für die Beschreibung von IT-gestützten Prozessen in SOAs, Clouds oder Grids entwickelt Fraunhofer FIRST die XML-basierte Workflow-Beschreibungssprache *GWorkflowDL*, welche auf High-Level-Petrinetzen basiert, um die Kontroll- und Datenflüsse von verteilten IT-Prozessen zu modellieren [Al06, Al06b, HA06]. Der Begriff *Workflow* wird hierbei als „Automatisierung von IT-Prozessen mittels Graphen“ verstanden. Ziel der GWorkflowDL ist neben der Modellierung und Analyse von Workflows insbesondere deren effektive Ausführung und Überwachung. Dieser Artikel bezieht sich auf die aktuelle GWorkflowDL Version 2.0, welche erstmalig die für die Abbildung von Hinterlegungen wichtigen Unternetze direkt unterstützt und zudem besondere Kantentypen für Verbesserung des Datenflusses in verteilten Systemen vorsieht. Da diese Neuerungen noch nicht ausführlich in anderen Veröffentlichung dokumentiert wurden, wird die GWorkflowDL 2.0 im Folgenden zumindest informal eingeführt. Für eine formale Spezifikation der GWorkflowDL sei auf das XML-Schema der GWorkflowDL [Ho09], sowie auf Kapitel 3 von [Vo08] verwiesen.

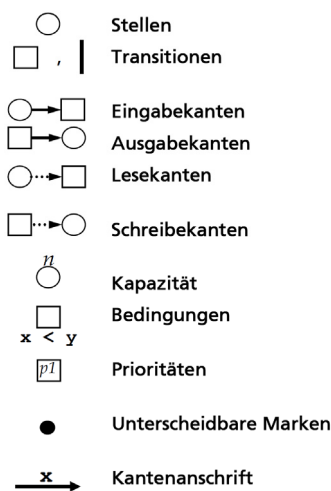


Abbildung 1: Komponenten zur grafischen Notation der GWorkflowDL.

Abbildung 1 zeigt die grafische Notation der einzelnen Bestandteile der GWorkflowDL. Entsprechend des Standards ISO/IEC 15909-1 [Is04] werden *Stellen* durch Kreise, *Transitionen* durch Quadrate und *Eingabe-* sowie *Ausgabekanten* durch durchgezogene Pfeile dargestellt. Im IT-Prozess repräsentieren die *Stellen* Platzhalter für Daten oder Kontrollmarken. Die *Kapazität* gibt die maximale Anzahl von Marken an, die auf einer Stelle liegen können. Falls keine Kapazität angegeben ist, haben die Stellen standardmäßig eine unendliche Kapazität. Die *Transitionen* dienen zur Modellierung abstrakter oder konkreter Aktivitäten, die zum Beispiel auf Web-Service-Methodenaufrufe in einer SOA, Programmausführungen in einem Grid oder auf Unterprozesse – die selber wieder als Petrinetz dargestellt werden können – abgebildet werden. Die *Eingabekanten* sind Pfeile von Stellen nach Transitionen, *Ausgabekanten* sind Pfeile von Transitionen nach Stellen. Die Stellen, die über Eingabekanten mit einer Transition verbunden sind, nennt man *Eingabestellen* dieser Transition. Analog hierzu nennt man die Stellen, die über Ausgabekanten mit einer Transition verbunden sind, *Ausgabestellen*. Jede Stelle kann mehrere unterscheidbare *Marken* enthalten, die als ausgefüllte Kreise dargestellt werden und im XML-Format der GWorkflowDL entweder einen Verweis auf Daten (z.B. als URL) oder die Daten selber enthalten. Eine spezielle Form der Marken sind die Kontrollmarken, welche lediglich die booleschen Werte „wahr“ oder „falsch“ annehmen können. Transitionen können *Bedingungen* enthalten, die in Form von XPath-Ausdrücken [CD99] als boolesche Funktionen der Eingabemarken formuliert werden.

Eine Transition nennt man *aktiviert*, wenn auf allen Eingabestellen der Transition mindestens eine Marke vorhanden ist und keine der Ausgabestellen ihre Kapazität erreicht hat. Eine aktivierte Transition kann *eintreten* (*ausgeführt werden, schalten*), wenn alle ihre *Bedingungen* erfüllt sind. Beim Eintreten einer Transition wird von jeder Eingabestelle eine Marke entfernt und anschließend auf jede Ausgabestelle eine neue Marke gelegt. Sind mehrere Transitionen zugleich aktiviert, tritt die Transition ein, welche die höchste *Priorität* besitzt. Durch die Einführung von Prioritäten wird die GWorkflowDL Turing-Vollständig und erhält die gleiche Beschreibungskraft wie Inhibitor-Netze. Dadurch werden zwar die Analysemöglichkeiten eingeschränkt (zum Beispiel der Beweis der Beschränktheit), da aber die Workflow-Ausführung und nicht deren Analyse im primären Fokus steht, wird dies in Abwägung der besseren Ausdrucksmöglichkeiten in Kauf genommen. Zudem bietet das Konzept der Prioritäten gute Möglichkeiten, die Ausführung von Workflows zu beschleunigen, z. B. indem bei nebenläufigen Transitionen alle Transitionen des kritischen Pfades mit hohen Prioritäten versehen werden.

In der GWorkflowDL kann über das XML-Element `<operation>` eine Transition mit einer bestimmten Workflow-Aktivität verknüpft werden, sodass jedes Eintreten der Transition eine Ausführung der entsprechenden Aktivität zur Folge hat. Jede Aktivität konsumiert dabei einen Satz Eingabedaten – also die Daten jeweils einer Marke aller Eingabestellen – und erzeugt einen Satz Ausgabedaten, der anschließend in Form von Marken auf alle Ausgabestellen gelegt wird (jeweils eine Marke pro Ausgabestelle). Anders als in der Theorie erfolgt das Eintreten einer Transition nicht instantan, sondern benötigt eine unbestimmte Zeitdauer, da das Schalten der Ausführung einer realen Workflow-Aktivität entspricht. Um die nebenläufige Verarbeitung eines GWorkflowDL-Prozesses zu ermöglichen, müssen daher während der Ausführung die Eingabemarken gesperrt und bei beschränkter Kapazität Platz für die Ausgabemarken reserviert werden.

Zur besseren Modellierung von Workflow-Aktivitäten, die nicht nur Daten konsumieren (= lesen + löschen) und neue Daten erzeugen, sondern auch Daten lesen bzw. vorhandene Daten (ggf. teilweise) überschreiben, wurden zusätzlich *Lesekanten* bzw. *Schreibekanten* eingeführt, die durch gestrichelte Pfeile zwischen Stellen und Transitionen bzw. Transitionen und Stellen dargestellt werden. In der Theorie sind Lese- und Schreibekanten gleichbedeutend mit einer Schleife. Eine *Schleife* bezeichnet hierbei einen Bereich im Petrinetz, in dem eine Transition und eine Stelle sowohl durch eine Eingabekante, als auch durch eine Ausgabekante miteinander verbunden sind, die Stelle somit zugleich Eingabestelle und Ausgabestelle einer Transition ist. Um festzustellen, ob eine Transition aktiviert ist, kann man alle Lese- und Schreibekanten jeweils durch eine Schleife von Eingabe- und Ausgabekante ersetzen und oben definierte Regel zum Begriff „aktiviert“ anwenden.

In der Praxis erlauben *Lesekanten* die effektive Modellierung von nebenläufigem Zugriff von Workflow-Aktivitäten auf gemeinsame Daten. Anstatt also beim Eintreten der Transition eine Marke von einer Stelle zu entfernen und sie anschließend unverändert wieder auf die selbe Stelle zu legen, können die Marke und die damit verbundenen Daten an Ort und Stelle verbleiben. Damit die Transition aktiviert ist, muss die Marke jedoch auf der mit der Lesekante verbundenen Stelle bereits vorhanden sein. Bei der Ausführung der Workflow-Aktivität werden die Daten gelesen, aber nicht verändert. Im Gegen-

satz hierzu werden bei *Schreibekanten* Daten überschrieben ohne sie zuvor zu lesen. Beim Eintreten der Transition wird jeweils eine Marke, die bereits auf einer mit einer Schreibekante verbundenen Stelle liegt, durch eine neue Marke ersetzt. Eine Schleife, die sowohl eine Lese- als auch eine Schreibekante darstellen soll, kann alternativ als *Aktualisierungskante* durch eine gestrichelte Linie mit Pfeilspitzen an beiden Enden modelliert werden. Eine Aktualisierungskante beschreibt das Lesen und anschließende (ggf. teilweise) Überschreiben von Daten. Analog hierzu beschreibt eine *Auswechsellkante* eine Schleife aus Eingabe- und Ausgabekante, bei der die entsprechende Aktivität Daten nimmt (also liest und löscht) und anschließend auf derselben Stelle neu erzeugt. Eine Auswechsellkante kann alternativ als durchgezogener Strich mit Pfeilspitzen an beiden Enden dargestellt werden. Das Ersetzen von Schleifen durch Lese-, Schreibe-, Aktualisierung- oder Auswechsellkanten ändert jedoch nichts an der prinzipiellen Abfolge eines Petrinetzes. Sie dienen lediglich zur Optimierung der Datenflüsse bei der Ausführung der mit den Transitionen verbundenen Aktivitäten.

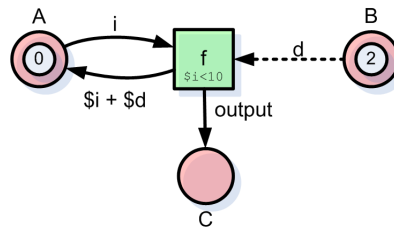


Abbildung 2: GWorkflowDL-Beispiel eines iterativen Aufrufs einer Aktivität f.

Die Bedeutung der *Kantenanschriften* weicht in der GWorkflowDL etwas von der sonst gebräuchlichen Notation in High-Level-Petrinetzen ab. Üblicherweise werden die Kantenanschriften dazu verwendet, um die Typen von Marken zu bezeichnen, die beim Eintreten einer Transition von der Transition konsumiert bzw. erzeugt werden [Is04]. In der GWorkflowDL dienen die Kantenanschriften hingegen dazu, die Marken den einzelnen Parametern einer Workflow-Aktivität zuzuordnen. Bei Eingabe- oder Lesekanten gibt die Kantenanschrift eine Variable vor, über die im Kontext der Transition auf die entsprechenden Marken zugegriffen werden kann. Befindet sich zum Beispiel auf einer Stelle, welche über eine Eingabekante mit der Kantenanschrift i mit einer Transition verbunden ist, eine Marke mit dem Inhalt 5, so wird beim Eintreten der Transition der Wert 5 der Variable i zugewiesen: $i=5$. Im Kontext der Transition kann auf den Inhalt der Variablen mit einem vorangesetzten „ $\$$ “ zugegriffen werden, z.B. in der Bedingung $\$x > 4$. Die Kantenanschriften an Ausgabe- bzw. Schreibekanten geben hingegen entweder einen Ausgabeparameter einer Aktivität an, oder eine Verarbeitungsanweisung in Form eines XPath-Ausdruckes, der eine Funktion der Ausgabedaten der Workflow-Aktivität sowie der durch die eingehenden Kantenanschriften definierten Variablen ist, z.B. $\$i+6$.

Die Notation eines ausführbaren GWorkflowDL-Prozesses ist exemplarisch in Abbildung 2 dargestellt. Die Transition f ist aktiviert da die Stellen A und B eine Marke enthalten und die Stelle C eine unendliche Kapazität besitzt. Die Transition f kann eintreten, da die Bedingung $\$i < 10$ mit $i=0$ erfüllt ist. Wenn die Transition eintritt wird die Aktivität $f(i, d)$ ausgeführt mit $i=0$ und $d=2$. Anschließend wird die Marke auf Stelle A mit dem Inhalt „0“ entfernt und eine neue Marke mit dem Inhalt $\$i+\$d=2$ auf Stelle A gelegt. Zudem wird eine neue Marke mit dem Wert des Ausgabeparameters

output auf die Stelle C gelegt. Insgesamt tritt die Transition f fünfmal ein und zwar mit den Eingabedaten $i=\{0, 2, 4, 6, 8\}$ und $d=\{2, 2, 2, 2, 2\}$.

Die XML-Syntax der GWorkflowDL ist eine direkte Umsetzung eines High-Level-Petrinetzes und orientiert sich an der Petri net Markup Language (PNML) [WK02]. Im Gegensatz zur PNML, welche überwiegend auf die Visualisierung und Analyse von Petrinetzen ausgerichtet ist, bietet die GWorkflowDL jedoch bessere Möglichkeiten, Transitionen mit realen Workflow-Aktivitäten zu verknüpfen. Außerdem werden Kanten nicht als eigenständige Komponenten, sondern als Unterelement von Transitionen definiert, da dies eher dem intuitiven Vorgehen bei der Modellierung von Methoden- oder Funktionsaufrufen in IT-Prozessen entspricht. Die Erweiterbarkeit der GWorkflowDL ist über generische property-Elemente gegeben, mit denen die einzelnen Komponenten des Petrinetzes annotiert werden können. Weitere GWorkflowDL-Beispiele, darunter auch die GWorkflowDL-Repräsentation einiger Workflow-Muster aus [AH02] (*Workflow Patterns*) finden Sie in der Workflow-Registry des myExperiment-Projektes [My09].

Um komplexe Prozesse in verschiedenen Detailabstufungen zu modellieren, können – ähnlich wie die Hinterlegungen in EPKs – einzelne Transitionen mit Unternetzen verknüpft werden. Ein Beispiel ist in Abbildung 3 dargestellt. Die Anschlussbedingungen sind über die Kantenanschriften gegeben. Bei jedem Eintreten der Transition wird eine neue Instanz des Unterprozesses gebildet und jeweils eine Marke von jeder Eingabestelle des übergeordneten Netzes an die Stelle des Unternetzes kopiert, die durch die entsprechende Kantenanschrift gegeben ist. Nachdem das Unternetz vollständig ausgeführt wurde, also keine Transition mehr aktiviert ist, werden jeweils eine Marke von der durch die ausgehenden Kantenanschriften bezeichneten Stellen vom Unternetz wieder in das übergeordnete Netz kopiert.

Durch die Marken des Petrinetzes wird nicht nur ein Prozessmuster, sondern auch der Zustand einer jeden Prozessinstanz beschrieben. Dies ermöglicht die direkte Überwachung der Prozesse sowie eine einfache Realisierung von Fehlertoleranzmechanismen.

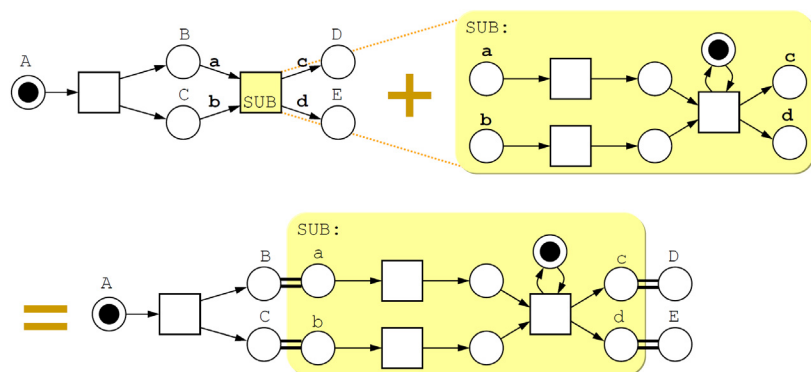


Abbildung 3: Beispiel eines hierarchischen GWorkflowDL-Netzes bei dem die Transition „SUB“ auf ein Unternetz verweist, welches bei jedem Eintreten der Transition ausgeführt wird.

4.2 Grid Workflow Execution Service (GWES)

Der von Fraunhofer FIRST entwickelte *Grid Workflow Execution Service (GWES)* ermöglicht die Automatisierung und das interaktive Management von komplexen und dynamischen Prozessabläufen in Serviceorientierten Architekturen oder Grid-Umgebungen [Ho08, Ho06, GH07, Vo08]. Ein Alleinstellungsmerkmal ist die vollständige Ressourcenvirtualisierung. Die abstrakte Modellierung der Prozessabläufe und Ressourcenbeschreibungen erlaubt die vollständig automatische Abbildung auf jeweils geeignete Hard- und Software-Ressourcen bzw. Dienste. Derzeit eingesetzt wird der GWES unter anderem in den D-Grid-Projekten MediGRID, MedInfoGrid, Services@MediGRID, BauVOGrid, PneumoGRID und TextGrid.

Der GWES wird in der Regel als Web-Service bereitgestellt, welcher GWorkflowDL-Netze automatisch, persistent und fehlertolerant auf verteilten IT-Ressourcen ausführt. Neben dem GWES besteht das Grid-Workflow-Management-System aus einer Ressourcen- und Workflow-Datenbank, einem *ResourceMatcher* zur Abbildung von abstrakten Job-Anfragen auf verfügbare Hardware- und Software-Ressourcen, einem Monitoring-System, welches über verteilt installierte *ResourceUpdater* alle 10-20s Informationen über die aktuelle Auslastung der Zielsysteme erhält, sowie einem Scheduler, welcher die Auswahl der Zielsysteme optimiert. Das Grid-Workflow-Management-System ermöglicht somit neben einer automatischen Ausführung von Workflows ein Meta-Scheduling von Anwendungen, welches sich im Produktivbetrieb bereits vielfach bewährt hat und bei dem die Anwender keine besonderen Kenntnisse der verteilten Infrastruktur benötigen.

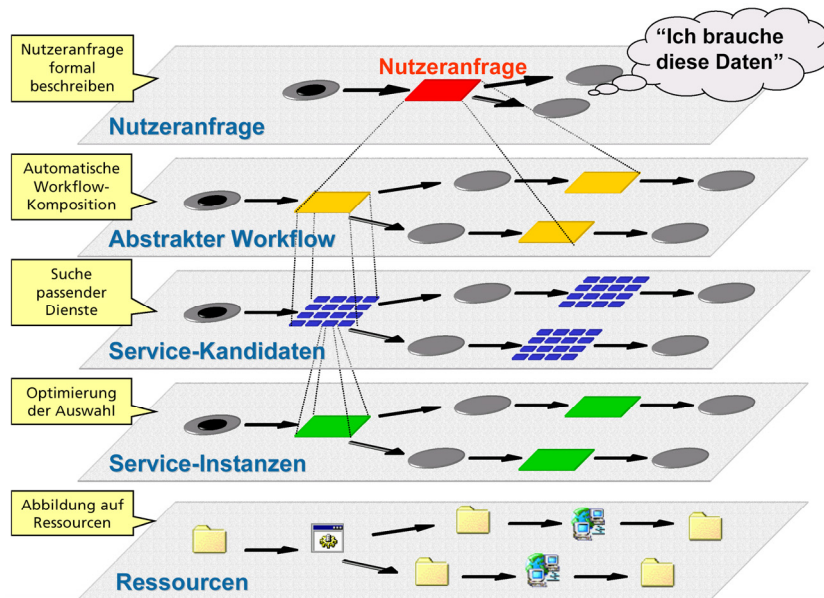


Abbildung 4: Automatische Abbildung von dynamischen Prozessen auf Ressourcen.

Abbildung 4 zeigt den gewählten Lösungsansatz für die automatische Abbildung von dynamischen Workflows auf die jeweils geeigneten und verfügbaren Ressourcen. Nutzeranfragen werden zunächst auf abstrakte Workflows (gelb) abgebildet, in diesem Fall durch die automatische Abbildung von EPKs auf GWorkflowDL-Dokumente (siehe nächstes Kapitel). Jeder Aktivität werden dann zunächst mit Hilfe des ResourceMatcher Service-Kandidaten (blau) zugeordnet, welche die entsprechende Funktionalität bereitstellen. Ein Scheduler wählt einen der Kandidaten aus (grün) und führt die Aktivität auf den entsprechenden Ressourcen aus.

Unter *Scheduling* wird die Abbildung von Workflow-Aktivitäten auf Ressourcen verstanden. Hierbei kann es sich sowohl um IT-Ressourcen (Dienste/Software/Hardware) als auch um Personalressourcen handeln. Abbildung 5 verdeutlicht diesen Prozess. Hierzu werden zunächst alle im aktuellen Prozessschritt nebenläufig ausführbaren Aktivitäten nach ihrer Priorität sortiert. Anschließend werden zu jeder Aktivität prinzipiell geeignete Ressourcen gesucht. Für jede Ressource wird eine Qualitätskennzahl berechnet, die z. B. von der Auslastung der Ressource abhängt. Passende Ressourcen, deren Qualität einen bestimmten Schwellwert (im Beispiel 0,8) überschreiten, werden den Aktivitäten zugeordnet.

5 Überführung von EPKs in ausführbare IT-Prozesse

Um EPKs in ausführbare IT-Prozesse zu überführen, haben wir einen Konverter entwickelt, welcher EPML-Prozessbeschreibungen unter Verwendung einer XSLT-Transformation [C199] in GWorkflowDL-Workflowbeschreibungen übersetzt. Zugrunde

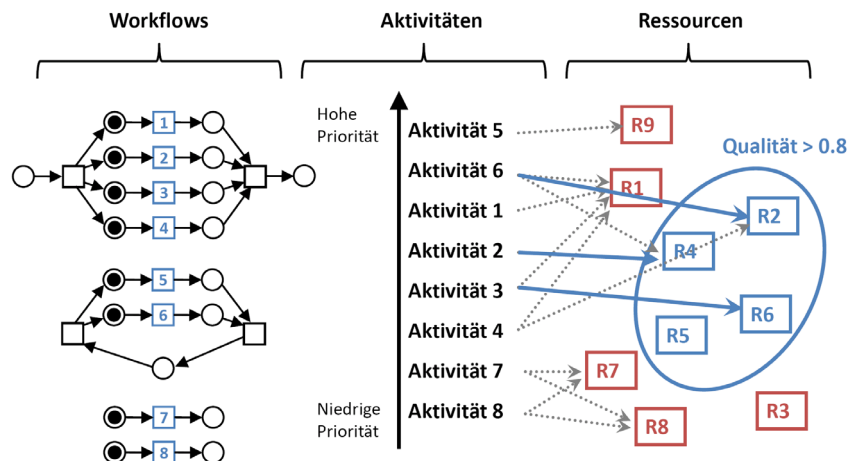


Abbildung 5: Abbildung der Workflow-Aktivitäten von aktivierten Transitionen auf verfügbare Ressourcen.

liegt hierbei eine Abbildung von EPKs auf Petrinetzen ähnlich zu der in [Aa99] beschrieben. Während van der Aalst jedoch jeweils einzelne EPK-Komponenten auf eine Kombination von Stellen, Transitionen und Kanten abbildet, haben wir eine Abbildungsvorschrift verwendet, welche Ereignisse, Funktionen und Konnektoren einer EPK in genau eine Transition oder Stelle eines Petrinetzes mit den ggf. notwendigen Eingabe- und Ausgabekanten übersetzt. Dadurch wird der Transformationsprozess einfacher und benötigt zudem keine Erweiterungen der EPK, falls zwei Konnektoren (z. B. XOR und AND) direkt miteinander verbunden sind. Analog zu [Aa99] ignorieren auch wir den OR-Operator, zumal er in der Praxis wenig eingesetzt wird und zudem, je nach Definition des OR-Operators, durch eine Kombination aus AND und XOR ersetzt werden kann.

Die verwendeten Abbildungsvorschriften zur Transformation der einzelnen Komponenten von EPKs in Petrinetze sind in Abbildung 6 dargestellt. Während Ereignisse und XOR-Konnektoren von EPKs auf jeweils eine Stelle des Petrinetzes abgebildet werden, werden Funktionen und AND-Konnektoren auf jeweils eine Transition abgebildet. Bei den Kanten gibt es zwei Sonderfälle: Kanten in einer EPK, die eine Funktion mit einem AND-Konnektor verbinden, werden im Petrinetz durch eine zusätzliche Stelle ergänzt, da sowohl Funktionen als auch AND-Konnektoren durch Transitionen abgebildet werden und eine direkte Kante zwischen zwei Transitionen in einem Petrinetz nicht erlaubt ist. Analog hierzu werden Kanten zwischen Ereignissen und XOR-Konnektoren in EPKs durch eine zusätzliche Transition im Petrinetz ergänzt.

Direkte Kanten zwischen zwei Funktionen sind in EPKs zwar oftmals nicht erwünscht, aber in der Praxis durchaus anzutreffen. Ein EPK-Modell kann somit für den Nutzer kompakter dargestellt werden, indem sogenannte Trivialereignisse (z. B. „Brief ist an Post übergeben“ als Ergebnis nach einer Funktion „Brief an Post übergeben“) weggelassen werden. Zudem gestatten auch die meisten Modellierungswerkzeuge derartig vereinfachte Modelle. Um auch diese „unsauberen“ EPKs in korrekte Petrinetze zu überführen, wird eine Kante zwischen zwei Ereignissen im Petrinetz durch die Kombination einer Eingabekante, einer Transition und einer Ausgabekante dargestellt. Analog wird eine

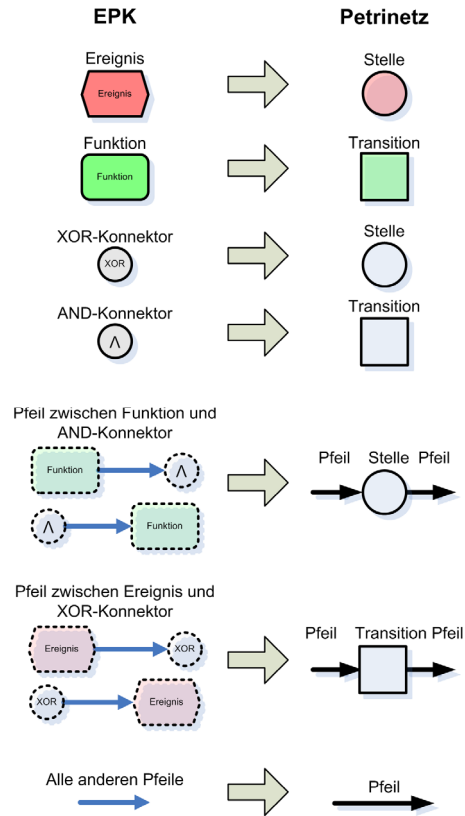


Abbildung 6: Abbildungsvorschriften zur Transformation von EPKs in Petrinetze.

Kante zwischen zwei Funktionen im Petrinetz durch eine Ausgabekante, eine Stelle und eine Eingabekante dargestellt (nicht in Abbildung 6 gezeigt).

Ein einfaches Beispiel einer Transformation einer EPK in ein Petrinetz zeigt Abbildung 7. Die Ereignisse A, B und D werden zu den Stellen A, B und C; die Funktionen D und E zu den Transitionen D und E. Die beiden XOR-Konnektoren werden jeweils durch eine Stelle abgebildet, der AND-Konnektor durch eine Transition. Zwischen die Stellen A und XOR, sowie zwischen B und XOR wird jeweils eine Transition eingefügt, alle anderen Kanten werden 1:1 abgebildet. Mit den Transformationsvorschriften aus Abbildung 6 lässt sich somit die Übersetzung einer EPK in ein Petrinetz in einem Schritt durchführen, ohne wie bei [Aa99] zunächst die EPK um zusätzliche Komponenten erweitern zu müssen. Dieses Vorgehen produziert zwar in einigen Fällen einige redundante Stellen/Transitionen im Netz, welche sich aber leicht bei Bedarf wieder reduzieren lassen.

Ein gewichtiger Unterschied zwischen EPKs und GWorkflowDL-Netzen ist der, dass EPKs in der Regel den Kontrollfluss von fachlichen Prozessmustern beschreiben, während GWorkflowDL-Netze konkrete Instanzen von ausführbaren (technischen) Prozessen inklusive der darin zu verarbeitenden Daten sind. Um eine EPK auf einen ausführbaren Prozess abzubilden, benötigt man neben den Funktionen, Ereignissen und deren Verknüpfungen daher folgende zusätzliche Informationen:

Anfangsmarkierung (Kontrollmarken): Die Anfangsmarkierung gibt die Verteilung von Marken auf den Stellen des Petrinetzes zu Beginn der Ausführung an. Dabei wird in

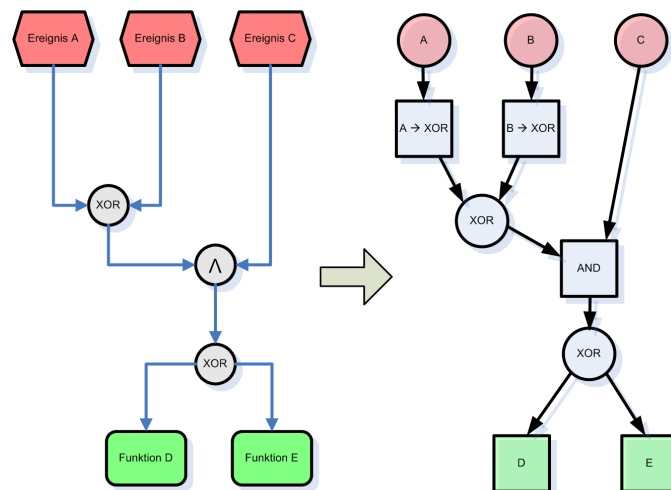


Abbildung 7: Transformation einer EPK in ein Petrinetz analog zum Beispiel aus [Aa99]. Die Anschriften „XOR“ und „AND“ im Petrinetz dienen lediglich der Information, aus welchen Teilen der EPK die Komponenten herrühren und sind für die Funktion des Petrinetzes nicht notwendig.

der GWorkflowDL zwischen Kontrollmarken und Datenmarken unterschieden. Die Kontrollmarken der Anfangsmarkierung können dadurch generiert werden, dass die Startereignisse der EPK durch ein Attribut am Ereignis annotiert werden (z.B.: `<attribute typeRef="workflowStart"/>`).

Anfangsmarkierung (Datenmarken): Komplexer stellt sich die Generierung der notwendigen Datenmarken auf den Stellen dar. Eine Möglichkeit stellt die Verwendung von Transitionen dar, die mit einer Workflow-Aktivität verknüpft sind, die aktiv im Rahmen des Prozesses aus den Kontrollmarken der Anfangsmarkierung Datenmarken erzeugen („hole-Daten-Aktivität“). Bei diesem Vorgehen besteht die Anfangsmarkierung also nur aus Kontrollmarken. Eine zweite Möglichkeit ist die explizite Modellierung von Daten unter Verwendung von *Informationsobjekten* der EPKs. Diese werden üblicherweise mit Funktionen verbunden, um deren Relation zu Dokumenten oder anderen Daten in einer EPK darzustellen. Diese Informationsobjekte können in Datenmarken übersetzt werden, die auf ggf. zusätzlich zu erzeugende Eingabe- bzw. Ausgabestellen gelegt werden. Eine dritte Möglichkeit ist die Verwendung von speziellen Attributen an Ereignissen, analog zu den Kontrollmarken. So kann zum Beispiel in der EPML über ein Attribut `<attribute typeRef="data" value="http://server/data.dat"/>` ein Ereignis mit dem Vorhandensein von bestimmten Daten in Verbindung gebracht werden.

Ausführbare Aktivitäten (Operation): Die GWorkflowDL zielt auf die konkrete Ausführung von Workflows in einem verteilten System ab. Als Mindestinformation benötigt es hierfür das `<oc:operationClass>`-Element, welches die Klasse von Aktivitäten festlegt, die beim Eintreten einer Transition ausgeführt werden soll. Die technisch einfachste Art, diese Information in einer EPK vorrätig zu halten, ist über eine Namenskonvention, bei der der Name einer Funktion in einer EPK identisch ist mit dem Namen des entsprechenden `<oc:operationClass>`-Elements. Der *ResourceMatcher*, der für die Suche passender Dienste zuständig ist, muss dies dann in seiner Konfiguration berücksichtigen. Des Weiteren können auch Ontologiedienste verwendet werden, um Funktionsnamen oder zusätzliche Kennungen auf passende Aktivitätsklassen abzubilden. Die verschiedenen Möglichkeiten der Annotierung sind in Abbildung 8 exemplarisch dargestellt.

Kantenanschriften: Die Kantenanschriften der GWorkflowDL stellen eine Beziehung zwischen den konkreten Daten (Marken auf Stellen) und bestimmten Funktionsparametern einer Workflow-Aktivität her. Eine EPK beschreibt jedoch einen abstrakten Prozessablauf, der in der Regel unabhängig von konkreten Daten ist und daher diese Beziehung nicht enthält. Um trotzdem die Kantenanschriften für den Datenfluss in einem GWorkflowDL-Netz generieren zu können, benötigt man somit eine externe Wissensba-

```
<function id="4">
  <name>f</name>
  <attribute typeRef="ontologyClassId" value="fService"/>
  <attribute typeRef="operationClass" value="urn:dgrdl:service:f"/>
  <attribute typeRef="wsOperation" value="http://server:8080/£?wsdl f"/>
</function>
```

Abbildung 8: EPML-Beispiel einer Annotierung einer EPK-Funktion durch Attribute, die für die Zuordnung einer Funktion zu ausführbaren Workflow-Aktivität verwendet werden können.

sis, die zum Beispiel Profile der Workflow-Aktivitäten mit deren Eingabe- und Ausgabeparametern vorhält. Wenn bei der Benennung der Daten gewisse Namenskonventionen eingehalten werden, können die dazu gehörenden Kantenanschriften z. B. auch direkt aus der WSDL-Schnittstellenbeschreibung des entsprechenden Web-Services extrahiert werden.

Hinterlegungen: Hinterlegungen einer EPK können wie in Abbildung 3 gezeigt hierarchisch durch Unter-Workflows abgebildet werden, auf die innerhalb einer Transition verwiesen werden kann.

Abbildung 9 zeigt exemplarisch den Ausschnitt aus dem XSLT-Stylesheet, welcher für die Transformation eines Ereignisses der EPK (*event*) in eine Stelle des Petrinetzes (*place*) zuständig ist. Die EPK-Annotation *control.token*, *workflowStart* oder *startEvent* wird dabei in eine Marke (*token*) des Petrinetzes übersetzt, welches Bestandteil der Anfangsmarkierung ist. Der EPK-Gegenpart der GWorkflowDL-Komponente wird über ein `<property>`-Element im Petrinetz annotiert, sodass das Petrinetz sehr einfach wieder auf die ursprüngliche EPK abgebildet werden kann – z.B. zur Darstellung des Fortschritts des Prozessablaufes in der EPK und zur Synchronisation bei Ad-hoc-Änderungen.

```

<!-- convert event to wf:place -->
<xsl:template match="event">
  <xsl:element name="place"
    namespace="http://www.gridworkflow.org/gworkflowdl">
    <xsl:attribute name="ID">p<xsl:value-of select="@id"/></xsl:attribute>
    <xsl:apply-templates select="name"/>
    <xsl:apply-templates select="graphics/position"/>
    <xsl:apply-templates select="attribute"/>
    <!-- convert token attribute to token -->
    <xsl:for-each select="(attribute[@typeRef='control.token']
      | attribute[@typeRef='workflowStart']
      | syntaxInfo[@implicitType='startEvent'])[1]">
      <xsl:element name="token"
        namespace="http://www.gridworkflow.org/gworkflowdl">
        <xsl:element name="control" name-
space="http://www.gridworkflow.org/gworkflowdl">true</xsl:element>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:template>

```

Abbildung 9: Ausschnitt aus dem XSLT-Stylesheet zur Konvertierung von EPML nach GWorkflowDL.

6 Fallstudie: Mängelmanagement im Bauwesen

Die effektive Zusammenarbeit und Kooperation von Unternehmen sowie die effiziente Nutzung von geeigneten Informationstechnologien sind Grundvoraussetzungen für den Erfolg jedes Zusammenschlusses von Personen, Unternehmen und realen Organisationen. Die spezifische Situation im Bauwesen wird von folgenden Faktoren geprägt: der Einmaligkeit jedes Bauwerks (Produktunikat) und der Einmaligkeit jedes Projekts, das von einem Zusammenschluss vieler, überwiegend kleiner Unternehmen durchgeführt wird (Projektunikat). Weiterhin werden die Anforderungen an die Abwicklung von Bau-

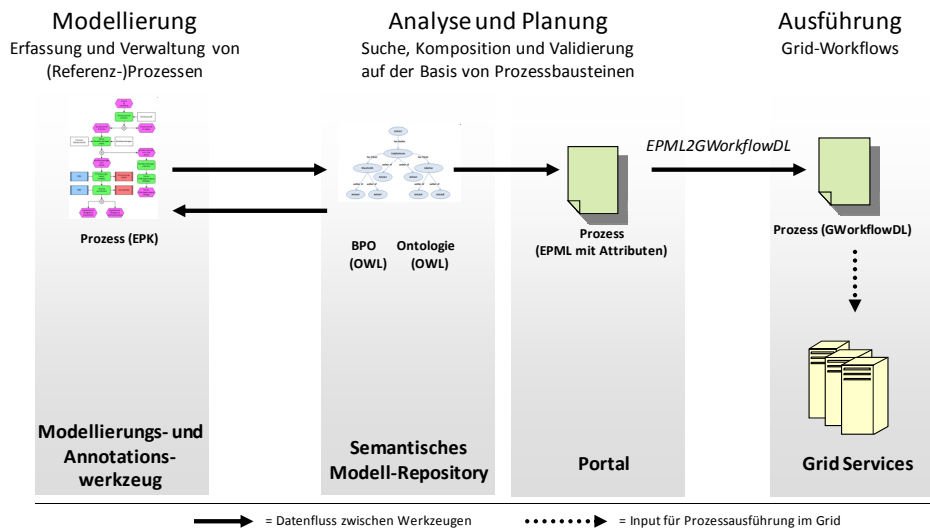


Abbildung 10: BauVOGrid Methodenbaukasten

projekten einerseits und das Bestreben nach Wirtschaftlichkeit und Kostenreduktion andererseits immer höher.

Im Rahmen von BauVOGrid¹ werden verschiedene Werkzeuge zur Prozessmodellierung und Automatisierung eingesetzt, welche die gesamte Kette von der fachlichen Modellierung von Geschäftsprozessen bis hin zur automatischen Ausführung der darunterliegenden IT-Prozesse auf der Grid-Infrastruktur von BauVOGrid abbilden. Da die unterschiedlichen Werkzeuge jeweils eine eigene Sicht auf die Prozesse haben, liegen diesen oftmals auch unterschiedliche Formalismen und Beschreibungssprachen zugrunde, die ineinander überführt werden müssen. Die verschiedenen Werkzeuge und Methoden bilden einen integrierten Werkzeug- und Methodenkasten zur Prozessmodellierung. Bei der Abbildung dynamischer Prozesse werden Zusatzinformationen bei der Modellierung mitgegeben, welche die Integration bzw. Kopplung von Prozessen in Grid-Umgebungen (teil-)automatisiert ermöglichen. Abbildung 10 zeigt die verschiedenen Bestandteile des hierzu entwickelten Methodenbaukastens.

In der Phase der *Modellierung* erfolgt die Erhebung der Referenzprozessmodelle mit einem Standard-Modellierungswerkzeug.

In der Phase der *Analyse und Planung* werden auf der Basis dieser Prozessbausteine konkrete, in einem Bauprojekt ablaufende Prozesse neu erstellt oder geändert. Hierbei wird dem Grundgedanken gefolgt, dass ein Modell nicht gänzlich neu erstellt werden muss, sondern dass Änderungen am Prozess vielmehr durch das Neuarrangieren von

¹ BauVOGrid steht für „Grid-basierte Plattform für virtuelle Organisationen im Bauwesen“. Das Projekt wird vom BMBF innerhalb der D-Grid Initiative gefördert (Förderkennzeichen: 01IG07001A).

Prozessbausteinen leicht und im Vergleich zur Neu-Modellierung zeitsparend umgesetzt werden können. Auch die Ausführung von Prozessen zum Zeitpunkt des Prozessablaufes (Runtime) wird so unterstützt und beispielsweise Ad-hoc-Änderungen von real auf einer Baustelle ablaufenden Prozessen ermöglicht.

In der Phase der *Ausführung* werden schließlich die zuvor erstellten Prozessmodelle in auf dem Grid ausführbare Prozessmodelle überführt. Durch diese Modelle wird das Workflow-Management-System GWES konfiguriert, das eine Prozessausführung von Kooperationsverbänden in verteilten Systemen unterstützt. Zur Ausführung von Prozessen werden die aus den fachlichen Prozessmodellen gewonnenen ausführbaren Prozessbeschreibungen herangezogen, welche die Ablauflogik sowie die prinzipiell beteiligten Informationsobjekte und erforderlichen Ressourcen zur Ausführung eines Prozesses angeben.

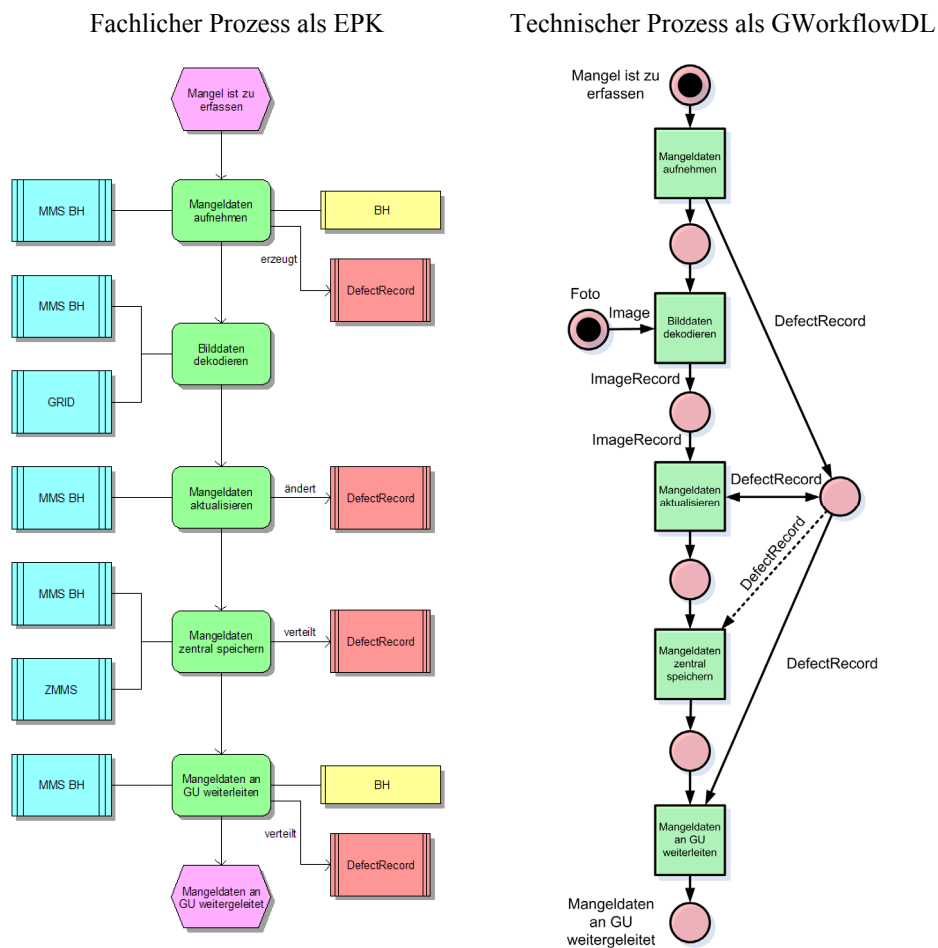


Abbildung 11: Beispiel einer Überführung eines einfachen Geschäftsprozesses aus dem Bereich Mängelmanagement im Bauwesen in einen ausführbaren Grid-Workflow.

Im Rahmen von BauVOGrid ist insbesondere die Transformation von EPML nach GWorkflowDL wichtig, da dies die Schnittstelle zwischen den Geschäftsprozessen und den ausführbaren Grid-Prozessen darstellt. Die Instanziierung eines Prozesses im Sinne der Erzeugung eines konkreten, ablauffähigen Vorgangs entspricht technisch gesehen einem Export eines EPML-Dokuments aus dem Modellierungswerkzeug, einer anschließenden Konvertierung nach GWorkflowDL (vgl. Abschnitt 5) und eines Importes in den GWES.

In dem Projekt wurde exemplarisch die Überführung von Geschäftsprozessen aus dem Bereich Mängelmanagement im Bauwesen auf Dienste der D-Grid-Infrastruktur durchgeführt. Abbildung 11 zeigt einen einfachen Beispielprozess, der die Schritte von der Aufnahme eines Baumangels durch den Bauherrn bis hin zur Übergabe der Mängeldaten an den Generalunternehmer zum Inhalt hat. Zunächst werden bei einer Baustellenbegehung verschiedene Baumängel durch den Bauherrn (BH) erfasst und mit einem mobilen Gerät (z. B. PDA) in das System eingegeben. Anschließend werden vorhandene Fotos, welche zusätzlich mit einem anderen Gerät von den Mängeln zur Dokumentation erstellt wurden, den Mängeln zugeordnet. Die Zuordnung erfolgt über die Dekodierung von im Foto enthaltenen Datamatrix-Codes (2D-Barcodes), die beim Fotografieren mit ins Bild gehalten wurden. Da die Datamatrix-Codes nur einen sehr kleinen Bereich der Fotografien ausmachen und erst nach mehreren Bildbearbeitungsschritten deutlich zu erkennen sind, ist dieser Prozessschritt sehr rechenintensiv und wird in der Regel auf Rechen-Clustern im Grid ausgeführt. Nachdem die Fotos den Mängeln zugeordnet wurden, müssen die Mängeldaten aktualisiert, zentral gespeichert und an den Generalunternehmer (GU) weitergeleitet werden, welcher für die Beseitigung der Mängel zuständig ist.

Die Notwendigkeit einer Automatisierung eines solchen Prozesses wird deutlich, wenn man beachtet, dass bei größeren Baustellen – wie zum Beispiel den Bau eines Flughafens – durchaus etwa 200GB an Fotos und 40.000 Mängel anfallen können, deren Beseitigung bis zu 10% der gesamten Bausumme machen kann. Für jeden Mangel muss dabei eine neue Instanz des entsprechenden technischen Prozesses ausgeführt werden. Durch den hier vorgeschlagenen Ansatz können die ausführbaren Prozesse direkt aus den entsprechenden EPK-Modellen generiert und auf leistungsfähigen Grid-Ressourcen ausgeführt werden, ohne dass sich die Nutzer mit den technischen Details der Infrastruktur auseinandersetzen müssen. Durch die strukturell sehr einfache und direkte Abbildung der Geschäftsprozessmodelle auf ausführbare IT-Prozesse können zudem dynamische Änderungen in den Geschäftsprozessabläufen, welche z. B. durch eine Havarie auf der Baustelle oder durch den Wechsel von Unterauftragnehmern (Insolvenz) ausgelöst werden können, schnell auf modifizierte IT-Prozesse abgebildet werden. Im Gegenzug wird eine direkte Überwachung der IT-Prozesse in der Geschäftsprozesssicht ermöglicht, sodass zum Beispiel auf Verzögerungen im Prozessablauf aufgrund von Engpässen in der IT (Datenspeicher voll) direkt auf Ebene der Geschäftsprozesse reagiert werden kann.

7 Schlussfolgerungen und Ausblick

Im vorliegenden Beitrag wurde aufgezeigt, wie EPK-Modelle durch einen Transformationsansatz in Grid- und Cloud-Computing-Umgebungen zur Ausführung gebracht werden können. Besondere Merkmale dieses Ansatzes sind zum Einen die vergleichsweise einfachen Transformationsregeln, die eine direkte und robuste Überführung von EPML-basierten Prozessbeschreibungen in GWorkflowDL-basierte Prozessbeschreibungen gestatten. Zum Anderen werden die Aspekte Skalierung, Ressourcenmanagement und Technologieunabhängigkeit explizit berücksichtigt, womit wesentliche Herausforderungen im Bereich des Grid- und Cloud-Computing adressiert werden.

Durch den vorliegenden Ansatz sollen zukünftig dynamische Prozesse ermöglicht werden, die eine enge bidirektionale Kopplung zwischen der Geschäftsprozessmodellierung einerseits und der ausführenden Plattform andererseits voraussetzen. Ad-hoc Änderungen der Geschäftsprozesse resultieren dadurch direkt in geänderten IT-Prozessen. Im Gegenzug kann auf Unterschiede in der Verfügbarkeit oder Zuverlässigkeit von IT-Ressourcen direkt durch Modifikationen im Geschäftsprozess reagiert werden.

Zukünftiger Forschungsbedarf besteht hinsichtlich der expliziten Abbildung der Informations- beziehungsweise Datenflüsse. Ein mögliches Vorgehen zur Generierung der hierfür notwendigen Datenmarken und Kantenanschriften im GWorkflowDL-Netz aus EPK-Attributen und EPK-Informationsobjekten wurde in Abschnitt 5 nur sehr grob skizziert und muss weiter konkretisiert werden.

8 Literaturverzeichnis

- [Aa99] van der Aalst, W.M.P.: Formalization and verification of event-driven process chains. In: Information and Software Technology, Volume 41, Issue 10, Elsevier, 1999; S. 639-650
- [AH02] van der Aalst, W.M.P.; ter Hofstede, A.H.M.: Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. Department of Technology Management, Eindhoven University of Technology P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands. 2002.
- [Ar05] Arroyo S, Cimpian E, Domingue J, Feier C, Fensel D, König-Ries B, Lausen H, Polleres A, Stollberg M (2005) Web Service Modeling Ontology Primer : W3C Member Submission 3 June 2005. Innsbruck, DERI. – URL <http://www.w3.org/Submission/WSMO-primer/> [Zugriffsdatum 20.07.2006]
- [AI06] Alt M., Gorlatch S., Hoheisel A., Pohl H.-W.: Using High-Level Petri Nets for Hierarchical Grid Workflows. 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006), Amsterdam, Netherlands, IEEE, 2006.
- [AI06b] Alt M., Hoheisel A., Pohl H.-W., Gorlatch S.: A Grid Workflow Language Using High-Level Petri Nets. PPAM05, LNCS 3911, Springer, 2006; S. 715-722
- [Ca04] Cabral L, Domingue J, Motta E, Payne TR, Hakimpour F (2004) Approaches to Semantic Web Services: an Overview and Comparisons. In: Bussler C, Davies J, Fensel D, Studer R (Hrsg) The Semantic Web: Research and Applications: Proceedings of the First European Semantic Web Symposium, ESWS 2004 Heraklion, Crete, Greece, May 10-12, 2004. Berlin, Springer, S. 225-239
- [Cl99] Clark, J.: XSL Transformations (XSLT) Version 1.0. W3C Recommendation, W3C, 1999; <http://www.w3.org/TR/xslt> [Zugriffsdatum 20.07.2009]

- [CD99] Clark, J.; DeRose, S.: XML Path Language (XPath) Version 1.0. W3C Recommendation, W3C, 1999; <http://www.w3.org/TR/xpath> [Zugriffsdatum 20.09.2009]
- [CS05] Cardoso J, Sheth AP (2005) Introduction to Semantic Web Services and Web Process Composition. In: Semantic Web Services and Web Process Composition: First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004.
- [Do09] Dollmann, T.; Fellmann, M.; Thomas, O.; Loos, P.; Hoheisel, A.; Katranuschkov, P.; Scherer, R.: Process-Oriented Collaboration in Grid-Environments: A Case Study in the Construction Industry. In Proceedings of the 15th Americas Conference on Information Systems: August 6-9, San Francisco, California, USA, 2009.
- [GH07] Gubala T.; Hoheisel A.: Highly Dynamic Workflow Orchestration for Scientific Applications. CoreGRID Technical Report Number TR-0101, CoreGRID, 2007.
- [HA06] Hoheisel A.; Alt M.: Petri Nets. In (Taylor I.J., Gannon D., Deelman E., Shields M.S. Hrsg.): Workflows for e-Science – Scientific Workflows for Grids, Springer, 2006.
- [Ho06] Hoheisel, A.: User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2006.
- [Ho08] Hoheisel, A.: Grid-Workflow-Management. In (Weisbecker, A.; Pfreundt, F.-J.; Linden, J.; Unger, S. Hrsg.): Fraunhofer Enterprise Grids – Software. Fraunhofer IRB Verlag, Stuttgart, 2008. ISBN: 978-3-8167-7804-2
- [Ho09] Hoheisel, A.: XML-Schema der Grid Workflow Description Language (GWorkflowDL) – Version 2.0. Fraunhofer FIRST, 2009; http://www.gridworkflow.org/kwfguid/src/xsd/-gworkflowdl_2_0.xsd [Zugriffsdatum 20.09.2009]
- [Is04] ISO/IEC 15909-1: High-level Petri nets – Part 1: Concepts, definitions and graphical notation. 2004.
- [GKS08] Gehre, A.; Katranuschkov, P.; Scherer, R.J.: Semantic Support for Construction Process Management in Virtual Organisation Environments. In: (Zarli, A.; Scherer, R. Hrsg.): ECPPM 2008 – eWork and eBusiness in Architecture, Engineering and Construction, Taylor & Francis, London, 2008, S. 85–92
- [KS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: Scheer A-W (Hrsg): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 89, Saarbrücken : Universität des Saarlandes. 1992; <http://www.iwi.uni-sb.de/Download/iwihefte/heft89.pdf>
- [Ma04] Martin, D.; Burstein, M.; Lassila, O.; Paolucci, M.; Payne, T.; McIlraith, S.A.: Describing Web Services using OWL-S and WSDL. Arlington, VA, BBN Rosslyn office. 2004; <http://www.daml.org/services/owl-s/1.1/owl-s-wsdl.html> [Zugriffsdatum 21.11.2005]
- [MN04] Mendling, J.; Nüttgens, M.: Exchanging EPC Business Process Models with EPML. In (Nüttgens, M.; Mendling, J. Hrgs.) Proc. of the 1st GI Workshop XML4BPM - XML Interchange Formats for Business Process Management; Modellierung 2004, Marburg Germany, 2004, S. 61-79.
- [My09] GWorkflowDL-Registry des myExperiment-Projektes: <http://www.myexperiment.org/workflows/search?query=GWorkflowDL> [Zugriffsdatum 20.09.2009]
- [Oa07] OASIS: Web Services Business Process Execution Language Version 2.0. OASIS, 2007; <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html> [Zugriffsdatum 20.09.2009]
- [Ob09] Object Management Group: Business Process Modeling Notation (BPMN) – Version 1.2. Object Management Group, 2009; <http://www.omg.org/spec/BPMN/1.2/> [Zugriffsdatum 20.09.2009]
- [Vo08] Vossberg, M.; Hoheisel, A.; Tolxdorff, T.; Krefting, D.: A Workflow-based Approach for Fault-tolerant Medical Image Transfer in Health Grids. In: Future Generation Computer Systems, Elsevier, 2008.
- [WK02] Weber, M.; Kindler, E.: The Petri Net Markup Language. In (Ehrig, H.; Reisig, W.; Rozenberg, G.; Weber, H. Hrgs.) : Petri Net Technology for Communication Based Systems. Berlin, Heidelberg; Springer, 2002.

Towards a Usability Assessment of Process Modeling Languages

Kathrin Figl[†], Jan Mendling[‡], Mark Strembeck[†]

[†]Vienna University of Economics and Business (WU Vienna)

Augasse 2-6, A-1090 Wien, Austria

`firstname.lastname@wu.ac.at`

[‡]Humboldt-Universität zu Berlin

Unter den Linden 6, 10099 Berlin, Germany

`jan.mendling@wiwi.hu-berlin.de`

Abstract: Visual process models provide support for analyzing and improving complex organizational processes. In this paper, we discuss differences of process modeling languages according to their ease of use and quality of user experience. These aspects are of importance for learning a modeling language, creating models, and understanding models. Modeling languages can be compared according to criteria as representational clarity, perceptual discriminability, perceptual immediacy, visual expressiveness, and graphic parsimony which are likely to influence cognitive effectiveness. In particular, we analyze the symbols sets of UML Activity Diagrams, YAWL, BPMN, and EPCs according to these criteria and discuss their strengths and weaknesses. Based on our findings, we propose a preliminary evaluation of the cognitive effectiveness of these modeling languages, which provides a basis for empirical studies.

1 Introduction

Business process models play an important role for the documentation of organizational processes and for the specification of information systems requirements. Although process modeling can be traced back at least to office automation research in the 1970s [Eil79], there is surprisingly little consensus on which elements should be covered by a process modeling language. Since the invention of EPCs in the early 1990s [KNS92, Sch00], there have been several other languages proposed including Petri nets [Mur89], UML Activity Diagrams [OMG09b], YAWL [AH05], and BPMN [OMG09a]. Yet, the preference for a particular language or the intervention of yet another language is often (not always) based on claims about usability (see [FLM⁺09]). However, usability considerations provide a good angle for checking the superiority of a language from an empirical perspective.

An important reference discipline to discuss usability issues of process modeling languages is cognitive research. This area deals with insights on how humans process information, create knowledge, and solve problems. A central property of the human brain is that it works with specialized areas. Sensory information from eyes and ears is transmit-

ted to the *sensory memory*, which holds an after-image of that information for a moment. This after-image is analyzed and key concepts are held in *short-term memory* for some minutes. *Visual perception* enables understanding of sensory information. Those aspects that can be linked to other prior knowledge can become part of *long-term memory*. All these three different memory parts have different characteristics, which are the foundation of various theories of comprehension, learning, and problem solving. Several of these theories have been applied to information systems modeling, among others cognitive load theory [Swe88], cognitive fit theory [Ves91], the cognitive dimensions framework for notational systems [GP96], and the theory of multimedia learning [May01]. All of them identify perspectives that can also be considered for discussing advantages and drawbacks of process modeling languages.

There are many orthogonal characteristics of process modeling languages that can be subject to a cognitive investigation. In this paper we focus on a recent framework proposed by Moody and Van Hillegersberg [MH08] who discuss desirable properties of notation elements from a perceptual perspective. Such a perspective is particularly appealing for process modeling languages since many of the prominent languages share a rather large common subset. Different workflow pattern analyses have shown (see, e.g., [Men08]) that UML Activity Diagrams, YAWL, BPMN, and EPCs share actions and different routing elements including different types of splits and joins. Our contribution is a discussion of how the principles of Moody and Van Hillegersberg can be used to identify strengths and weaknesses of process modeling languages. Furthermore, we synthesize them in terms of propositions, which speculate on the cognitive effects of these strength and weaknesses. In this way, we pave the way towards an empirical investigation of their arguments in an experimental setting.

The remainder of this paper is structured as follows. In Section 2, we summarize the major arguments on usability of modeling languages. Section 3 discusses cognitive theories which are relevant for understanding process modeling languages. Section 4 then turns to the work by Moody and Van Hillegersberg, which we utilize in Section 5 to discuss strengths and weaknesses of process modeling languages. Finally, Section 6 concludes the paper and gives an outlook on future research.

2 Theoretical Background

Usability in general can be described as the measure of the ease of use and quality of the user experience when interacting with a device the user can operate in some way or another [Nie09]. Similarly, according to the ISONORM 9241/11, usability is defined by the “extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (see [Int98]). In the context of modeling, *effectiveness* means how well the modeling technique assists in accomplishing modeling goals and *efficiency* refers to the resources needed to use a particular modeling technique (see [GW04]). Since the benefit of using a specific language for communication between analysts, developers and users largely depends on the cost of production, learning curve and speed of decay [AEHE07], usability aspects of modeling

languages are of practical relevance. Moody [Moo05] notes that in the area of researching the quality of modeling languages there is lack of recognition of the relevant ISO standards for quality management and suggests to build on these foundations. ISONORM 9241 is a multi-part standard entitled ‘Ergonomics of Human System Interaction’ and in the sub part 9241/110 general usability heuristics for the design, the description and evaluation of interactive systems are proposed [Int06]. Although these principles were designed for the interaction between a user and a software system, at least the criteria: *suitability for the task*, *self-descriptiveness*, *conformity with user expectations*, *suitability for individualization* and *suitability for learning* also fit to the evaluation of modeling languages (*controllability* and *error tolerance* fit less). Nielsen [Nie09] refers to similar quality characteristics for usability: *learnability*, *efficiency* (time needed to perform a task), *memorability* (performance after a period without using it), *amount of errors made by users* and *user satisfaction*. When considering the usability of a modeling language, *ease-of-use* refers to the effort of learning the language, using the language (active modeling) and understanding models defined in the respective language [RR07, p. 2022]. Although there are standards for measuring quality and usability of software, there is no equivalent for models and modeling languages [Moo05]. Therefore, the next section will detail a number of factors that have to be taken into account for an usability evaluation of modeling languages.

3 Cognitive Aspects of Modeling Languages

Basically, two main aspects of modeling can be distinguished (see [GW04]):

- **Creating models:** Process aspects of creating models include the effort (e.g. time) required to construct them, subjective ease-of-use, and ease-of-learning. The result of the modeling process is a model, which can be assessed concerning different quality aspects (such as correctness, accuracy, detail, completeness, quality, type of errors).
- **Understanding/Interpreting models:** In the process of model understanding, much cognitive activity is involved. The outcome of understanding is cognitive per se, i.e. it is created in the viewer’s cognition and not directly observable. It can only be measured indirectly, e.g. via ‘problem-solving’ tasks in which newly developed mental models have to be used or via comprehension tests (see, e.g., [GW04]). Additionally, recall tests or correctly identified problems can be used to capture ‘understanding’. Models from different modeling languages are likely to differ according to the effort required to interpret them and develop understanding, as well as in the perceived difficulty of obtaining information through their visual representation.

Both of these aspects show the complex interplay between human cognitive models and visual models. Therefore, the ability of a language to support appropriate translations between cognitive and visual models is probably the most essential criterion for determining its usability. Norman’s theory of action [ND86] can be used to explain differences between cognitive models of a model creator/viewer and a visually expressed model [GW04,

p. 256].

- A ‘gulf of execution’ in modeling is the gap between a user’s goals and the means that are applied to execute that goal. Discrepancies may, for instance, result from a modeling language that is not expressive enough and lead to extra effort, i.e. requiring additional thinking and distract from achieving the original modeling goal.
- A ‘gulf of evaluation’ refers to the gap between a stimulus (a model) and the resulting comprehension. Comprehension problems causing deviations between cognitive models and the actual visual representation/the intended meaning can for example be provoked by ambiguity of a modeling language.

Cognitive theories are central to the usability evaluation of different modeling languages. Mental processes as visual perception, information processing, reasoning and problem solving, attention, as well as short and long term memory are affected in learning how to use specific modeling languages, creating models, and understanding models. Aside from the fact that e.g. learning a modeling language may demand different mental processes than understanding a model, we hypothesize that there are specific aspects inherent a language that are likely to have an impact on all of these cognitive processes.

Up to now a variety of underlying cognitive theories were adopted to the context of modeling - often to explore potential benefits of the visual expression - as the dual coding theory [Pai91], the theory of working memory [BH74], the cognitive load theory [Swe88], the theory of multimedia learning [May01] and the external cognition theory [SR96].

Since process models usually represent complex relationships and humans have limited information processing capabilities (see [Ves91, p. 220]), a main goal in the design of process modeling languages is to reduce cognitive load for users to enable more effective problem solving. Low cognitive effort is positively related to a variety of quality aspects of models (see [ST05, p. 352]). Maes et al. [MP07] even define the perceived ease of understanding of a model as “the degree to which a person believes that using ... would be free of mental effort”. Cognitive load is determined by the amount of elements needed to be paid attention to at a point of time [Kir02]. There is a natural limit of the capacity of short-term memory of humans of approximately 7 ± 2 elements [Mil56]. Although the amount of elements is limited, their size and complexity is not. Chunking expands the capacity of short-term memory, because information units belonging together are chunked into one unit [GLC⁺01]. Short-term memory, refined by Baddely and Hitch [BH74] as working memory, includes besides a central executive component a phonological loop and a visuo-spatial sketchpad. Performance on simultaneous tasks requiring separate perceptual domains (as visual and verbal) is much higher than for example working with two different texts. Similarly, the dual coding theory postulates that visual information (e.g. pictures) and verbal information (e.g. texts) are stored and processed differently via separate mental channels that do not compete with each other (see [Pai91]). In particular, research found evidence that learning material combining text and visual images support learning better than text only. The cognitive multimedia learning theory [May01] proposes different principles how to combine text and pictures. According to the *contiguity principle*, learning is better, when text and pictures are presented spatially near each other. In

consequence, text and icons belonging together should be placed near each other in visual models. However, the combination of text and visual information can also lead to cognitive overload (*split-attention principle*).

The cognitive load theory [Swe88] details how the working memory load influences learning and knowledge acquisition. In experiments, Sweller [Swe88] found out that heavy cognitive load during problem solving exercises impairs learning, especially schema construction. The theory differs between three types of cognitive load: *intrinsic*, *extraneous* and *germane cognitive load*. In contrast to *intrinsic cognitive load* (determined by the complexity of information, i.e. the amount of elements, and their relations and interactions), the *extraneous cognitive load* is influenced by the way the information is represented (see [Kir02, p.4]). While the cognitive load devoted to learning and understanding (*germane cognitive load*) should be promoted, *extraneous cognitive load* should be held low. This can, for example, be achieved by reducing additional, irrelevant information. By mapping the cognitive load theory to the context of modeling, it becomes clearer, why modeling languages might vary in their cognitive effectiveness. If the same information is modeled in different modeling languages, the resulting models should, to a large extend, have a similar *intrinsic cognitive load*, but they differ in their *extraneous cognitive load*. The amount of *extraneous cognitive load* caused by the modeling languages leads to differences in learning and understanding (see, e.g., [CS96]).

Due to the complex control flows, the creation and understanding of process models is likely to demand high cognitive reasoning and effort for logical thinking for human users. Additionally, users may, for example, want to deduce correct assumptions on how business processes are carried out in an organization based on a process model, or determine what will happen if a process cannot be carried out. Visual models not only demand, but also support users in their reasoning processes because “cues to the next logical step in the problem may be present at an adjacent location” [LS87, p.65]. Most research on deductive reasoning focuses on experiments with textual material and little is known about reasoning involved in visual modeling. Research, has shown, for instance, that there exist systematic fallacies (so called ‘illusory inferences’) when individuals internally construct mental models on premises including modeling-level connectives (like conjunctions, inclusive, or exclusive disjunctions) and draw conclusions from these mental models (see, e.g., [KJL09, NJL06]). In correspondence to mental models this may also hold true for externalized visual process models. In addition, the vast body of literature on error analysis of process models seems to give a hint on the occurrence of systematic reasoning fallacies concerning connectors (see, e.g., [MNvdA07, MvdA07, Men07]). Since correct interpretation of connectors in process models is inevitable for overall understanding, we hypothesize that the visualisation of connectors may especially affect whether different process model notations support understandability.

Besides creating and understanding models, learnability is a main factor for the usability of a modeling language. Learnability of the modeling notation largely depends on the depth of knowledge someone wants to acquire (e.g. learning the full set of language constructs, vs. learning a subset that is sufficient for conducting a specific modeling task). Green and Petre [GP96] point out that ‘consistency’ of visual languages is relevant. This means that users who have learned a certain part of the language can infer the rest. In more complex

languages, as for example BPMN (39 language constructs), it will take longer to acquire expertise, as in more simple languages, such as EPC (9 language constructs) [RD07, p. 358]. Similar to notations in physics or geometry some knowledge on the conventions of the notation will be inevitable for understanding the intended meaning of a process model. Other general aspects regarding learnability include the prior knowledge and experience or the instruction, which may have more impact on learning than the language per se. Recker and Dreiling [RD07, p. 351] showed that transfer concerning the ability to understand and read a model from a specific process modeling notation to another is surprisingly easy. Students who had solely be trained to EPC models scored equally well in comprehension tests of BPMN and EPC models.

4 Evaluating Cognitive Quality of Modeling Languages

The form of visual information representation can have a significant impact on the efficiency of information search, explicitness of information, and problem solving (see [LS87]). Compared to the semantic development, little research has been undertaken in order to improve the visual syntax of modeling languages (see, e.g., [MH08]). One of the key goals for the visual design of a model is that viewers draw attention to those components crucial for understanding and cognitive inferencing (see [SR96, p.207]). According to Moody and Hillersberg [MH08] there are 5 principles for cognitive effective visual design of modeling notations: semiotic clarity, perceptual discriminability, perceptual immediacy, visual expressiveness and graphic parsimony.

- **Semiotic/Representational Clarity:** This principle points out the importance of a good fit between the graphical symbols used in a modeling notation and the semantic concepts they refer to. Anomalies like symbol redundancy (more than one symbol represents the same concept), overload (one symbol represents more than one concept), symbol excess and deficit (there are graphical symbols without a correspondence to a semantic construct or vice versa) should be avoided, since they lead to ambiguity and additional unnecessary cognitive load for the user (see [MH08, p. 20]). For cognitive efficiency, graphical constraining will be a goal¹. Consequently, notations will also highlight specific aspects of information at the expense of other aspects (see [GP96]). For evaluating the representational potential of a process modeling language the Bunge-Wand-Weber model [YR95] can be used. In a recent comparative analysis (see [RRIG09]), the modeling techniques EPC and BPMN were compared concerning their ontological completeness and clarity. The results of this comparison revealed interesting differences that are likely to have an effect on the cognitive efficiency of these languages.
- **Perceptual Discriminability:** Perceptual discriminability of symbols determines how easy it is for a user to distinguish between different symbols and visually rec-

¹Graphical constraining is defined by Scaife and Rogers [SR96, p.189] as “the way graphical elements in a graphical representation are able to constrain the kinds of inferences that can be made about the underlying represented world”

ognize differences between them. It is highly influenced by the amount of visual variables in which symbols differ (also referred to as visual distance). If visual symbols are highly unique on a visual variable, they are likely to ‘pop out’ and are easy to locate in a model [MH08, p. 21]. Low perceptual discriminability can lead to misunderstandings. Research showed that for instance rectangles and diamonds in ER diagrams are easily confused (see [NC99]). On the other hand, if different symbols in a notation have similar attributes as color or shape, they are likely to be recognized as belonging together. According to the ‘Gestalt law of similarity’, elements will be grouped together if they have similar attributes (see [Wer38]). In consequence, symbols in a modeling language should differ sufficiently by visual variables to be perceptual discriminable. However, sometimes it is intended that symbols share visual variables if they should be recognised as related to each other.

- **Perceptual Immediacy:** Perceptual immediacy supports the user’s understanding of the meaning of graphical symbols and representations, and describes whether symbols and their corresponding concepts are easily associated. Icons, for example, are easily associated with their referent real-world concepts. Iconic representations for classes of activities could improve the understandability of process models as suggested by [MR08], but are not yet commonly used. Additionally, spatial relationships of symbols can help to induce specific desired interpretations by a user (e.g. left-to-right implies sequence). Even a small layout change in the same graph may transport a different meaning (e.g. centrality vs. hierarchy) (see [AEHE07]). Both principles, iconic and spatial representation, seem to be related to the concept of cognitive metaphors [Lak92], which refers to the mapping of a conceptual domain in terms of another target domain. Modeling languages are likely to differ according to the intuitiveness of the visual metaphors they use for expressing real-world concepts. Theories of long term memory as the spreading activation theory [And83] assume that internal mental representation consist of associative networks in the form of a graph. According to Rockwell and Bajaj [BR05], modeling effectiveness and efficiency will be higher if the symbols used in a modeling notation are more similar to “the concept of node-relationship arc depiction of information”. Therefore, it can be hypothesised that process modeling notations with nodes and edges are likely to be intuitively understandable (because of their compatibility with internal mental representations).
- **Visual Expressiveness:** Modeling notations which fully exploit the range of visual variables (spatial dimensions like horizontal and vertical, as well as shape, size, colour, brightness, orientation, and texture) have higher visual expressiveness. In comparison to a textual representation (words), which are encoded verbally in their reading direction, visual symbols are internally encoded in their spatial arrangement (see [San77]). Therefore, it is of importance to assess spatial dimensions of modeling notations. Using swimlanes in activity diagrams, for example, includes both planar variables for depicting information on who is working on an activity (see [MH08, p. 29]).
- **Graphic Parsimony:** Model complexity is determined by “the use of embedded symbols and distinct graphic symbols” [NC99, p. 141]. High graphic complexity of

a modeling notation may impair understanding and therefore should be avoided. On the other hand there will be a tradeoff between complexity and expressiveness.

5 Language Differences

5.1 Basic Elements of Process Modeling Languages

In general, process modeling languages include a number of basic, generic, and consensual elements that allow to define process flows. Below we briefly introduce these basic elements, before the subsequent sections discuss how different process modeling languages define these generic elements.

Start A start node defines the initial node of a process. A start node may have one or more outgoing edges.

End An end node defines the final node of a process or sub-process. An end node does only have incoming edges.

Tasks Different variants of task nodes exist. In general, a task node models a clearly defined action or process step. Tasks can be conducted by human users or software agents.

Decision A decision node models choices in a process flow. Decision nodes thereby define alternative routing options in a certain process.

Merge A merge node consolidates different (optional) paths that result from a choice made by a decision node.

Split A split node models parallel branches in a process. Thus, sub-processes started by the same split node are performed simultaneously.

Join A join node synchronizes parallel sub-processes. Typically, a join node synchronizes sub-processes resulting from a split node.

Event An event defines the occurrence of a certain type of ‘incident’. In process models events are typically used to trigger tasks. In addition, a task may also produce events which may trigger other tasks.

In Figures 1 and 2, we see five variants of a simplified credit application process modeled via an UML activity model, BPMN, YAWL, EPC, and as classical Petri net, respectively.

5.2 UML Activity Models

Figure 3 depicts the basic elements of UML2 activity models (see [OMG09b]). The main element of an activity model is an Activity. The Activity represents a process that consists of Actions and different types of control nodes. Actions thereby define the tasks

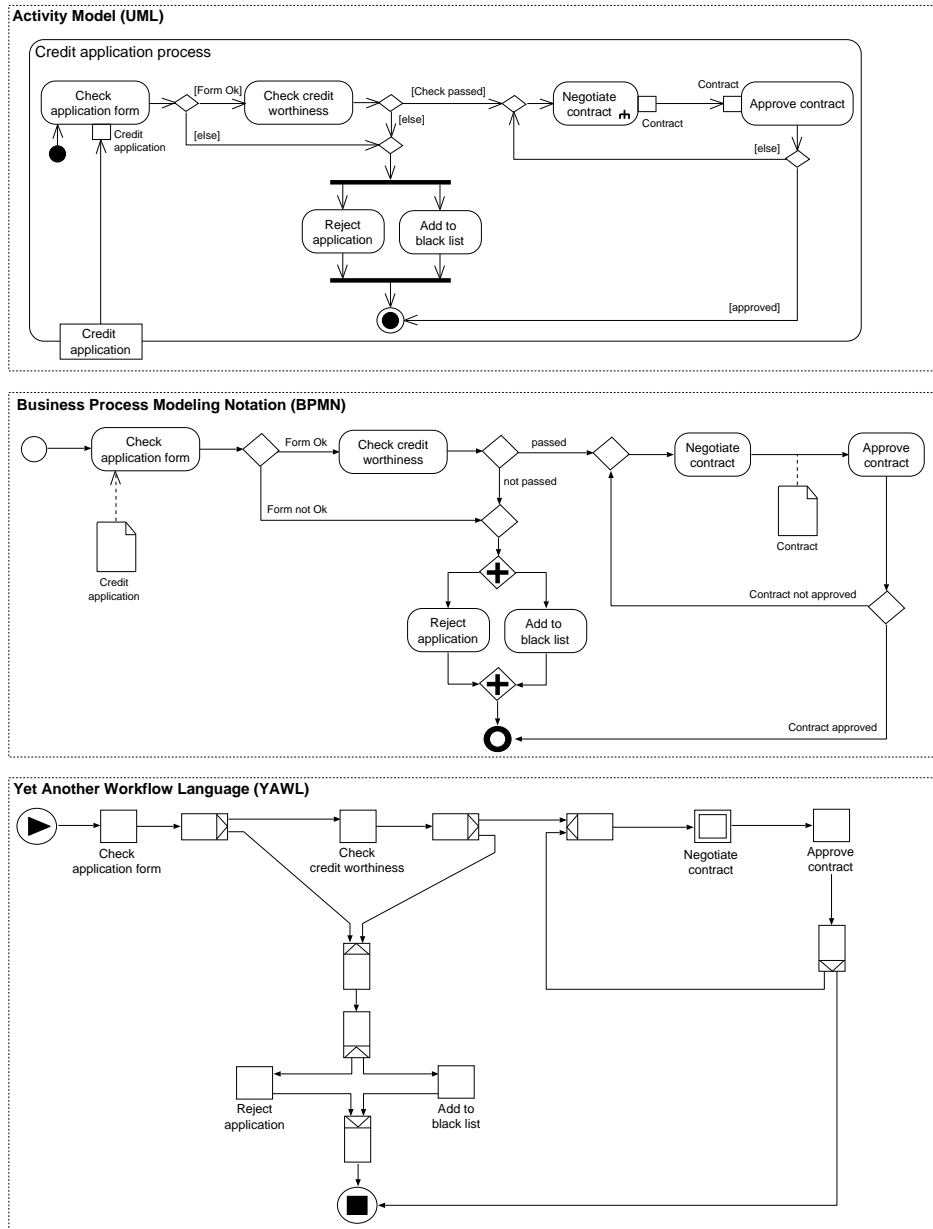


Figure 1: An example process modeled via UML, BPMN, and YAWL

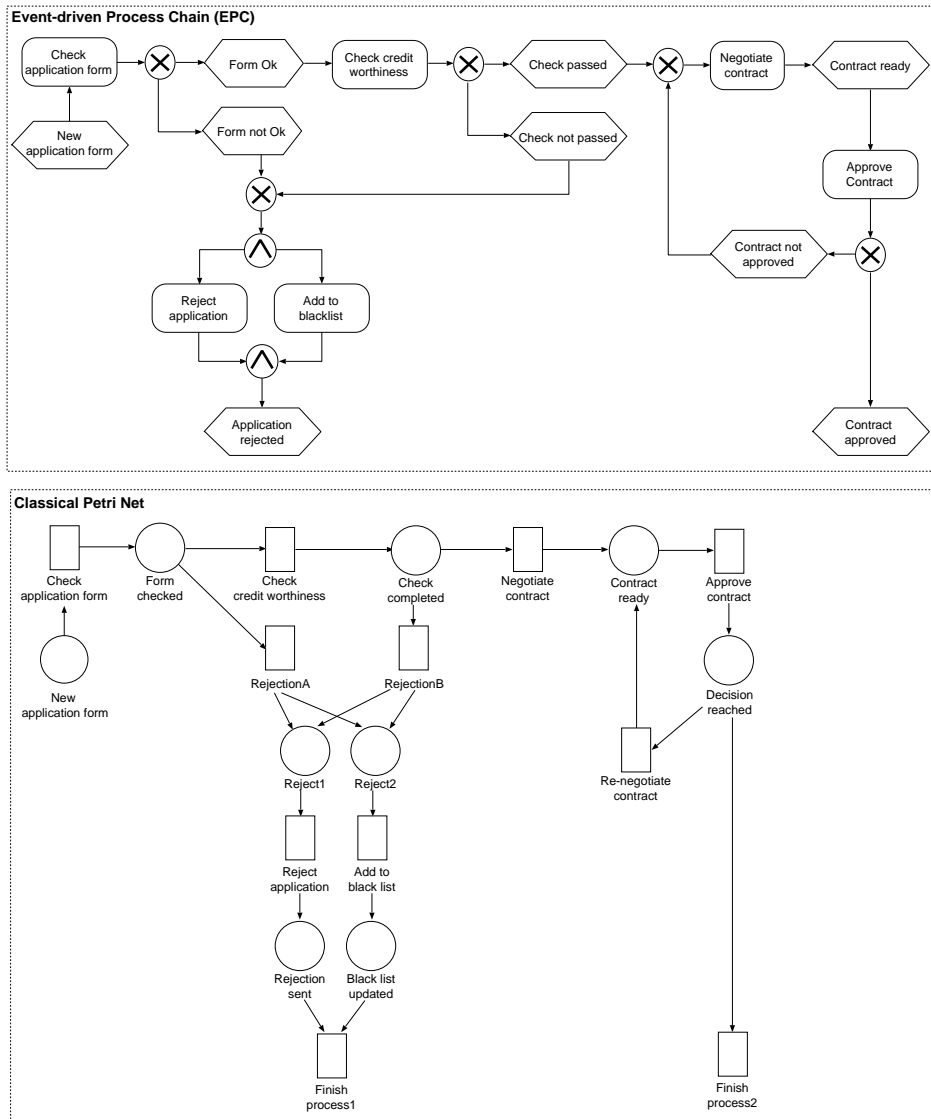


Figure 2: An example process modeled as EPC and classical Petri net

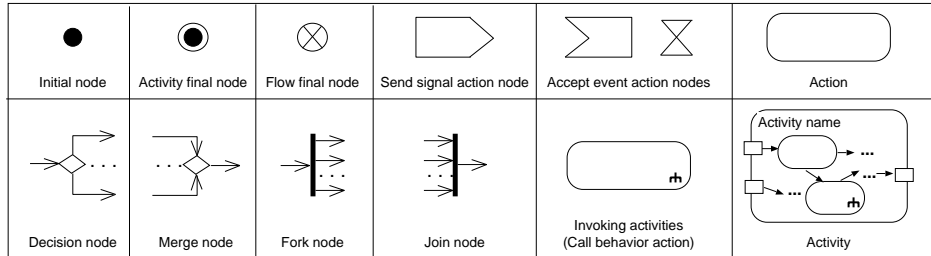


Figure 3: Basic elements of UML activity models

(steps) that are performed when executing the corresponding Activity. To model the (synchronous or asynchronous) invocation of other processes from an Activity, one may use an action symbol (round-cornered rectangle) including a small rake symbol (see Figure 3). In essence, this modeling element defines that the corresponding step in a process is itself a process that is modeled via an own Activity model.

Activity models have a token semantics, similar (but not equal) to Petri nets. Each Activity may have one or more Initial nodes and one or more Final nodes. In case the process flow reaches an Activity Final node, it immediately stops the execution of all actions and terminates the respective Activity. In contrast, a Flow Final node only destroys the tokens flowing into it but does not terminate the Activity. A Decision node is represented by a diamond-shaped symbol and has one incoming and multiple outgoing edges. A Merge node is represented by a diamond-shaped symbol and has multiple incoming and one outgoing edge. A Fork node is represented via a thick line and has one incoming and multiple outgoing edges. A Join node is represented via a thick line and has multiple incoming and one outgoing edge. A Send Signal node is represented by a convex pentagon. Two different types of Accept Event Action nodes exist. An ordinary Accept Event Action node is represented via a concave pentagon and waits for the occurrence of a particular type of event. The corresponding event type may result from a signal produced by a Send Signal node. An Accept Time Event Action node is represented by an hour-glass shaped symbol.

The UML notation partially violates some of the criteria identified by Moody. There is some gap of representational clarity with different end symbols (activity final and flow final). On the other hand, the different end symbols have clearly defined and distinct semantics. Furthermore, start and end nodes look quite similar. A strong point of UML is the clear discrimination between decisions and concurrency. Both routing elements are significantly different. A certain degree of immediacy is provided by Send Signal and Accept Event nodes. The rest of the notation remains rather abstract. Visual expressiveness is used to distinguish concurrent from alternative branching. Altogether, the notation set is quite parsimonious although it is quite expressive.

5.3 Business Process Modeling Notation (BPMN)

Figure 4 depicts the core process model elements of the BPMN notation [OMG09a]. Note that there are further symbols, which are more specific, e.g. for throwing exception events. BPMN uses different types of events for defining the start and end of a process. The symbols shown in the figure can be extended with additional symbols to describe, for instance, a message start event or a timer. BPMN also distinguishes different types of end events. The terminate event explicitly kills all parts of the process that still might be active. Sending and receiving intermediate events are defined by filled and unfilled symbols within a circle, respectively. The different steps in a process are captured as tasks. Different types of tasks exist, e.g. embedded and collapsed sub-processes. Finally, there are gateways for specifying routing constraints. Splits and joins both use a diamond shape. XOR gateways can be drawn without a symbol or with an X inside. AND gateways contain a plus sign.

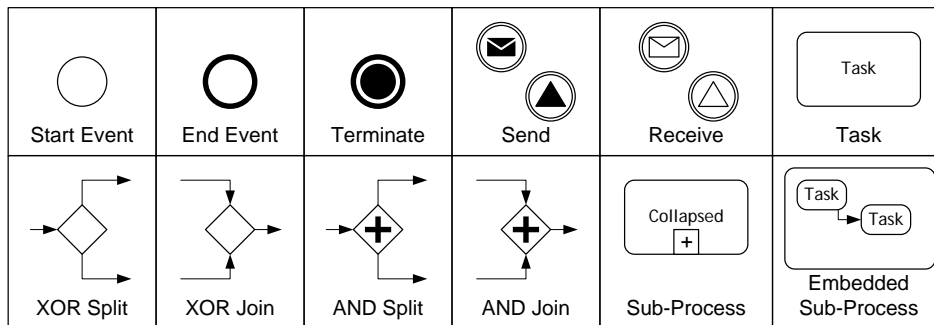


Figure 4: Basic elements of BPMN diagrams

The BPMN notation exhibits different weaknesses. Representational clarity is partially violated. XOR gateways can be described with different symbols (blank diamond and X diamond). This may be confusing to model readers. Furthermore, there is little perceptual discrimination between different pairs of elements. Start, end, and intermediate events are only distinguished by their border (thin, bold, double-lined). Also XOR gateways as X diamonds are difficult to distinguish from AND gateways with a plus. This may lead to confusion, too. Intermediacy is rather low with the blank set of symbols. On the other hand, many event subtype symbols (e.g. envelop for message event) are very intuitive. Visual expressiveness is somewhat limited for the standard set of symbols: tasks and gateways are both quadrangles and gateways are circles. However, a rich set of shapes is used as symbols for event subtypes along with varying brightness for throwing (filled black) and catching of events (not filled). Texture and color is not used in the standard symbol set, but is often introduced by modeling tools for highlighting tasks. Finally, there is hardly any graphic parsimony. BPMN uses an extensive set of symbols and subtype symbols. This might make it difficult to learn the language.

5.4 Yet Another Workflow Language (YAWL)

The core elements of YAWL are shown in Figure 5. YAWL uses a unique start and end condition to mark the beginning and termination of a process. Conditions are used to separate tasks (similar to places in Petri nets), yet they do not need to be explicitly drawn. Tasks are depicted as rectangles. There are variants of tasks, e.g. multiple instance tasks and sub-process tasks. Routing behavior is defined by thin blocks on the left-hand and right-hand side of tasks. Both XOR and AND splits and joins use triangle symbols.

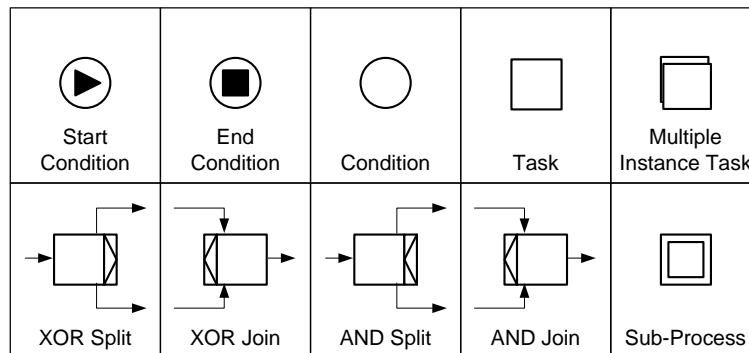


Figure 5: Basic elements of YAWL diagrams

The criteria by Moody point to potential problems of the YAWL notation. Representational clarity is violated, because a decision and a simple merge can be described either by conditions with multiple incoming and outgoing arcs, or with XOR joins and splits. There is also no direct support for events in YAWL. Thus, catching and throwing of events have to be defined using tasks. There is also a serious problem with discrimination of AND and XOR splits and joins. They share the same symbol. The semantics have to be identified also based on the position on the split/join block along with the number of incoming and outgoing arcs to the respective task. YAWL is also a quite abstract notation. However, start and end conditions provide a good intuition of their semantics based on an audio player metaphor. There is little visual expressiveness due to little variation in shape and size. Filled shapes are only used for starts and ends. On the other hand, the notation is quite parsimonious, at least when compared to BPMN.

5.5 Event-driven Process Chains (EPCs)

The EPC notation contains three types of symbols: circles for connectors, rounded rectangles of functions, and hexagons for events. There is no distinction symbol-wise between different types of events. Different connectors can be distinguished by their symbol, \times for XOR, \wedge for AND, and \vee for OR. Beyond that, there are two ways of defining sub-processes, either by a sub-process symbol on the right bottom corner of the function symbol or as a so-called process interface to start a subsequent process.

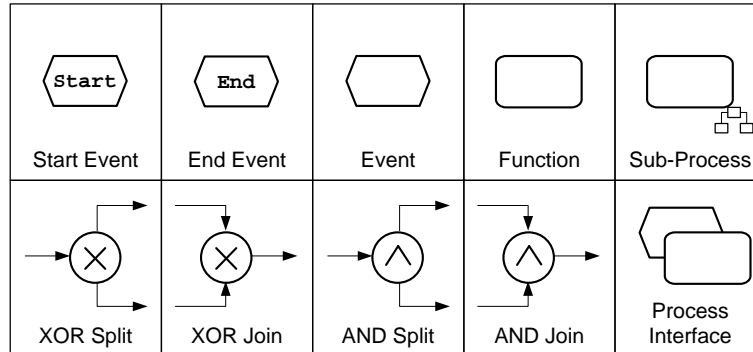


Figure 6: Basic elements of EPCs

EPCs also have problems with different usability criteria. The event symbol is heavily overloaded, which causes problems in terms of clarity. Start, end, intermediate events, and all kinds of events types map to the same symbol. There is also a problem to easily discriminate AND and OR connectors. They share the same symbol, only mirrored vertically. Altogether, the symbol set of EPCs is very abstract, such that there is little immediacy and intuition of the symbols. Visual expressiveness is also limited. Some tools depict events and functions in red and green color, which helps to distinguish them. The notation is also parsimonious by using only a small symbol set.

5.6 Petri Nets

Petri nets have the smallest symbol set of the languages discussed here. Transitions capture tasks, as well as synchronization (multiple in-arcs) and introduction of concurrency (multiple out-arcs). Places define a partial state, that can also be used to define decision points and simple merges.

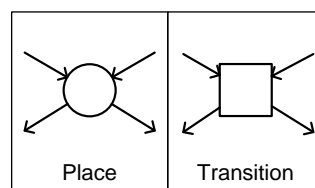


Figure 7: Basic elements of Petri nets

Due to the small set of symbols, Petri nets have problems with clarity. Their elements capture quite different behavior depending on the number of incoming and outgoing arcs. The two symbols, rectangles for transitions and circles for places, can be easily discriminated, though. Yet, they remain abstract such that there is no intuition or immediacy. Visual expressiveness only builds on the parameter shape, which is OK for only two sym-

bols. Altogether, Petri nets are a language of extreme graphic parsimony using only two different symbols.

5.7 Evaluation and Discussion

Based on the number and types of issues identified above we can hypothesize about understanding and learning performance depending on the process modeling language.

Representational Clarity A lack of clarity is likely to result in two problems when creating a model based on redundant symbols. First, modelers might need more time to decide which among several alternative elements to use for representing a certain real-world fact. Second, models created by different modelers may vary significantly in terms of the symbols they use for the same facts. These problems are likely to show up with BPMN and YAWL. On the other hand, symbol overload may result in corresponding ambiguity for somebody who is reading the model. This issue seems to be more relevant to EPCs and Petri nets. UML activity diagrams seem to have the highest degree of clarity from the languages being discussed.

Perceptual Discriminability Good discrimination should support fast visual perception and good recall in short term memory. All languages have some weaknesses in discriminating start and end nodes except YAWL. On the other hand, YAWL is very weak in discriminating routing elements. This is also partially the case for EPCs (OR versus AND) and BPMN (X versus +). UML very clearly distinguishes concurrency and alternative branching with significantly different symbols. Finally, EPCs have some weaknesses in discriminating functions and events.

Perceptual Immediacy Many language elements are very abstract such that no intuition is provided. Some counter examples are start and end conditions in YAWL and certain event types in BPMN. These elements should be easy to identify and to remember.

Visual Expressiveness The visual expressiveness of process modeling languages is limited. Most elements only vary in shape and size. Counter examples are again events in BPMN (throwing versus catching), YAWL start and end, as well as UML concurrency and alternative routing.

Graphic Parsimony BPMN is by far the richest language in terms of its symbol set. Therefore, it should also be the most time consuming to learn. YAWL and UML also provide a rather large set of symbols. EPCs and Petri nets are lean in terms of symbols, such that they should be easy to learn notation-wise. On the other hand, they are the most abstract notations such that they should be the most difficult ones to be applied correctly.

The different observations on weaknesses of process modeling languages offer the chance for a comparative evaluation. A particular aspect that is strongly supported by one language and weakly by another should directly materialize in a difference in understanding

or learning performance. A respective empirical experiment is missing so far, and on our agenda of future research.

6 Conclusion

Despite increasing consciousness about the need to consider the (non-technical) user's point of view in the area of information systems engineering, little research has been undertaken in order to improve and understand the usability of modeling languages. Up to now, few studies have investigated differences of process modeling languages concerning human factors such as ease-of-use, understandability, or learnability. On that account we build on cognitive theories to identify different perspectives for discussing usability of modeling languages. The main contribution of this paper is an analysis of the symbol sets of the five most prominent process modeling languages according to characteristics which are likely to have an impact on cognitive effectiveness and usability. The evaluation results show remarkable differences of languages according to representational clarity, perceptual discriminability, and graphic parsimony. Consequently, to meaningfully examine and better support the choice and development of process modeling languages, future research should investigate whether these differences of the visual notations in fact influence understandability or learnability. Further empirical studies including human errors analysis as well as problem-solving, comprehension, and recall tasks have the potential to provide new insights concerning the usability of different process modeling languages and to test our preliminary hypotheses.

References

- [AEHE07] J. Aranda, N. Ernst, J. Horkoff and S. M Easterbrook. A Framework for Empirical Evaluation of Model Comprehensibility. In *29th International Conference on Software Engineering (ICSE'07). International Workshop on Modeling in Software Engineering (MiSE-07)*, Minneapolis, USA, 2007.
- [AH05] W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4), June 2005.
- [And83] John R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 1983.
- [BH74] A.D. Baddeley and G. Hitch. *The psychology of learning and motivation: Advances in research and theory*, Jgg. 8, Kapitel Working memory., pp. 47–89. Academic Press, New York, 1974.
- [BR05] Akhilesh Bajaj and Stephen Rockwell. *Advanced Topics in Database Research*, Kapitel COGEVAL: A Propositional Framework Based on Cognitive Theories To Evaluate Conceptual Models, pp. 255–282. Idea Group Publishing, 2005.
- [CS96] Paul Chandler and John Sweller. Cognitive Load While Learning to Use a Computer Program. *Applied Cognitive Psychology*, 10(2):151–170, 1996. 10.1002/(SICI)1099-0720(199604)10:2<151::AID-ACP380>3.0.CO;2-U.

- [Ell79] C.A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems*, pp. 225–240, Boulder, Colorado, 1979. ACM Press.
- [FLM⁺09] D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich and S. Zugal. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In Terry Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Pnina Soffer and Roland Ukor, Hrsg., *Enterprise, Business-Process and Information Systems Modeling. Proceedings of the 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009*, Jgg. 29 of *Lecture Notes in Business Information Processing*, 2009.
- [GLC⁺01] Fernand Gobet, Peter C.R. Lane, Steve Croker, Peter C-H. Cheng, Gary Jones, Iain Oliver and Julian M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5(6):236 – 243, 2001.
- [GP96] T.R.G. Green and M. Petre. Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework. *J. Vis. Lang. Comput.*, 7(2):131174, 1996.
- [GW04] Andrew Gemino and Yair Wand. A Framework for Empirical Evaluation of Conceptual Modeling Techniques. *Requirements Engineering*, 9(4):248–260, 2004.
- [Int98] International Organization for Standardization. ISO 9241-11:1998 - Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability, 1998.
- [Int06] International Organization for Standardization. ISO 9241-110:2006 Ergonomics of human-system interaction – Part 110: Dialogue principles., 2006.
- [Kir02] Paul A. Kirschner. Cognitive load theory: implications of cognitive load theory on the design of learning. *Learning and Instruction*, 12(1):1–10, February 2002.
- [KJL09] Sangeet Khemlani and P. N. Johnson-Laird. Disjunctive illusory inferences and how to eliminate them. *Memory & Cognition*, 37(5):615–623, 2009.
- [KNS92] G. Keller, M. Nüttgens and A.-W. Scheer. Semantische Prozessmodellierung auf der Grundlage “Ereignisgesteuerter Prozessketten (EPK)”. Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1992.
- [Lak92] George Lakoff. *Metaphor and Thought*, Kapitel The Contemporary Theory of Metaphor. Cambridge University Press, 1992.
- [LS87] Jill H. Larkin and Herbert A. Simon. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11(1):65 – 100, 1987.
- [May01] Richard E. Mayer. *Multimedia Learning*. Cambridge University Press, Cambridge, Massachusetts, 2001.
- [Men07] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. Phd thesis, Vienna University of Economics and Business Administration, 2007.
- [Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, Jgg. 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.
- [MH08] Daniel Moody and Jos Hillegersberg. *Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams*. Springer-Verlag, Berlin, Heidelberg, 2008.

- [Mil56] G.A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.
- [MNvdA07] Jan Mendling, Gustaf Neumann and Wil M. P. van der Aalst. Understanding the Occurrence of Errors in Process Models based on Metrics. In Robert Meersman and Zahir Tari, Hrsg., *On the Move to Meaningful Internet Systems 2007*, Jgg. 4803 of *Lecture Notes in Computer Science*, pp. 113–130. Springer, Vilamoura, Portugal, 2007.
- [Moo05] Daniel Laurence Moody. Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. *Data & Knowledge Engineering*, 15(3):243–276, 2005.
- [MP07] Ann Maes and Geert Poels. Evaluating Quality of Conceptual Modelling Scripts Based on User Perceptions. *Data & Knowledge Engineering*, 63(3):701–724, 2007.
- [MR08] Jan Mendling and Jan Recker. Towards Systematic Usage of Labels and Icons in Business Process Models. In Terry Halpin, Hendrik Alexander Proper and John Krogstie, Hrsg., *12th International Workshop on Exploring Modeling Methods in Systems Analysis and Design*, CEUR Workshop Proceedings Series, pp. 1–13. Montpellier, France, 2008. CEUR.
- [Mur89] Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- [MvdA07] Jan Mendling and Wil M. P. van der Aalst. Formalization and Verification of EPCs with OR-Joins Based on State and Context. In John Krogstie, Andreas L. Opdahl and Guttorm Sindre, Hrsg., *Advanced Information Systems Engineering - CAiSE 2007*, Jgg. 4495 of *Lecture Notes in Computer Science*, pp. 439–453. Springer, Trondheim, Norway, 2007.
- [NC99] J. C. Nordbotten and M. E. Crosby. The effect of graphic style on data model interpretation. *Information Systems Journal*, 9(2):139–155, 1999. 10.1046/j.1365-2575.1999.00052.x.
- [ND86] Donald A. Norman and Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.
- [Nie09] J. Nielsen. Usability 101: Introduction to Usability, 2009.
- [NJL06] Mary R. Newsome and P. N. Johnson-Laird. How falsity dispels fallacies. *Thinking & Reasoning*, 12(2):214–234, 2006.
- [OMG09a] OMG. Business Process Modeling Notation (BPMN). available at: <http://www.omg.org/spec/BPMN/1.2/>, January 2009. Version 1.2, formal/2009-01-03, The Object Management Group.
- [OMG09b] OMG Unified Modeling Language (OMG UML): Superstructure. available at: <http://www.omg.org/technology/documents/formal/uml.htm>, February 2009. Version 2.2, formal/2009-02-02, The Object Management Group.
- [Pai91] Allan Paivio. Dual Coding Theory: Retrospect and Current Status. *Canadian Journal of Psychology*, 45(3):255–287, 1991.
- [RD07] Jan Recker and Alexander Dreiling. Does It Matter Which Process Modelling Language We Teach or Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education. In Mark Toleman, Aileen Cater-Steel and Dave Roberts, Hrsg., *18th Australasian Conference on Information Systems*, pp. 356–366, Toowoomba, Australia, 2007. The University of Southern Queensland.

- [RR07] Jan Recker and Michael Rosemann. Understanding the Process of Constructing Scales Inventories in the Process Modelling Domain. In Hubert Österle, Joachim Schelp and Robert Winter, Hrsg., *15th European Conference on Information Systems*, pp. 2014–2025, St. Gallen, Switzerland, 2007. University of St. Gallen.
- [RRIG09] Jan Recker, Michael Rosemann, Marta Indulska and Peter Green. Business Process Modeling- A Comparative Analysis. *Journal of the Association for Information Systems*, 10(4), 2009.
- [San77] J. L. Santa. Spatial transformations of words and pictures. *Journal of Experimental Psychology: Human Learning & Memory.*, 3:418–427, 1977.
- [Sch00] A.W. Scheer. *ARIS - Business Process Modeling, 3rd Edition*. Springer Verlag, 2000.
- [SR96] Mike Scaife and Yvonne Rogers. External cognition: how do graphical representations work? *Int. J. Hum.-Comput. Stud.*, 45(2):185–213, 1996.
- [ST05] Keng Siau and Xin Tan. Improving the quality of conceptual modeling using cognitive mapping techniques. *Data Knowl. Eng.*, 55(3):343–365, 2005.
- [Swe88] J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science: A Multidisciplinary Journal*, 12(2):257–285, 1988.
- [Ves91] I. Vessey. Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*. *Decision Sciences*, 22(2):219–240, 1991.
- [Wer38] M. Wertheimer. *A sourcebook of Gestalt psychology*, Kapitel Laws of organization in perceptual forms, pp. 7188. Routledge and Kegan Paul, London, UK, 1938. Reprinted from *Psychologische Forschung*, 4, pp. 301350, 1923.
- [YR95] Y. Wand and R. Weber. On the deep structure of information systems. *Information Systems Journal*, 5(3):203–223, 1995. 10.1111/j.1365-2575.1995.tb00108.x.

Modellierung von Entscheidungen und Interpretation von Entscheidungsoperatoren in einem WfMS

Jens Brüning, Peter Forbrig

Fakultät für Informatik und Elektrotechnik
Universität Rostock
Albert-Einstein-Str. 21
18059 Rostock
{jens.brueening, peter.forbrig}@uni-rostock.de

Abstract: In diesem Artikel wird die Modellierung von Entscheidungen in Workflows aus diversen Blickwinkeln betrachtet. Es existieren verschiedene Arten von Entscheidungen, die bei der Modellierung oder während des Ablaufs der Workflows getroffen werden müssen. Für die Modellierung gibt es unterschiedliche Sprachmittel bzw. Modellierungsweisen in Prozesssprachen wie z.B. EPKs, UML Aktivitätsdiagrammen oder YAWL um Entscheidungen auszudrücken. Der Prozessmodellierer kann zum einen die Art der Entscheidung und zum anderen die Phase festlegen, in der die Entscheidung getroffen werden soll. Für EPKs wird das Konzept der Entscheidungsfunktion um Entscheidungsprozesse erweitert, die es ermöglichen komplexere Entscheidungen mit Hilfe von hierarchischer Verfeinerung von Prozessen und Ereignissen zu modellieren. Des Weiteren werden implementierte Konzepte zur Endnutzer-Darstellung von unterschiedlichen Entscheidungsoperatoren an einem Workflow Management System (WfMS) vorgestellt.

1 Einleitung

Prozessmodelle stellen Handlungsabläufe dar, die Geschäftsprozesse dokumentieren und Abläufe vorschreiben. Es existieren unterschiedliche Modellierungsweisen von Prozessen, die Prozessmodelle entweder flexibler oder restriktiver für die Ausführung machen. Bei der Modellierungsweise *flexible by design* [SRM07] werden zwar die Aktivitäten definiert, die im Prozess ausgeführt werden müssen, die Reihenfolge ihrer Abarbeitung wird während der Designtime aber offen gelassen. Der ausführenden Person werden damit alternative Ausführungsmöglichkeiten gegeben. Diese muss sich bei jedem Schritt neu entscheiden und selektieren, welche Aktivität als nächstes durchgeführt werden soll. Da weniger vorgegeben wird hat der Nutzer zwar mehr Freiheiten, allerdings auch weniger Unterstützung. Meist kosten die Entscheidungen während der Runtime, die den Ablauf des Prozesses betreffen, auch Zeit.

Bei der Modellierung der Prozesse muss also ein Kompromiss zwischen Flexibilität und Unterstützung des Nutzers durch Vorgabe der Prozessschritte gefunden werden.

Es gibt außerdem unterschiedliche Arten von Entscheidungen, die zu unterschiedlichen Zeiten des Prozess-Lebenszyklus bzw. deren Instanzen getroffen werden können. Zum einen gibt es Entscheidungen, die die Ablauflogik der Aktivitäten betreffen. Hier werden viele Entscheidungen schon während der Designtime für das Prozessmodell getroffen, um festzulegen welche Aktivitäten nacheinander auszuführen sind. Zum anderen können während der Designtime Referenzmodelle herangezogen werden, die Entscheidungsoperatoren über Prozessalternativen beinhalten. Diese sind aufzulösen oder in die Runtime zu verlagern, um konfigurierte, unternehmensspezifische Prozessmodelle zu erhalten [Sch98].

Entscheidungen über die Wahl eines Prozesspfades können sowohl explizit als spezielle Aktivität in Geschäftsprozessen als auch implizit ohne Angabe einer solchen modelliert werden. Während der Designtime kann dabei bereits die Art der zu treffenden Entscheidung gewählt werden. Entscheidungen können von Menschen oder Maschinen getroffen werden, wobei dieser Artikel den Fokus auf die Modellierung der menschlichen Entscheidungen in Geschäftsprozessen legt.

Der Artikel ist wie folgt gegliedert. In Abschnitt 2 werden die Grundlagen gelegt, indem die verwendeten Workflow Modellierungssprachen kurz vorgestellt werden. In Abschnitt 3 werden die unterschiedlichen Entscheidungsarten in den verschiedenen Workflowsprachen modelliert. Durch die Verwendung mehrerer Sprachen zusätzlich zu den EPKs werden die Konzepte der unterschiedlichen Entscheidungsmodellierung aus verschiedenen Blickwinkeln beleuchtet und die Sprachen zueinander in Kontrast gesetzt und Ausdrucksmöglichkeiten verglichen. Abschnitt 4 präsentiert eine Implementierung von den Entscheidungsoperatoren an einem WfMS. Schließlich wird im Abschnitt 5 die Arbeit zusammengefasst und ein Ausblick für weitere Arbeiten gegeben.

2 Grundlagen

Im weiteren Verlauf des Artikels werden unterschiedliche Prozessmodellierungssprachen verwendet. Es werden die EPK [KNS92], YAWL [HA05], UML2 Aktivitätsdiagramme (AD) [UML] und Aufgabenmodelle [Pa00, LPV01] verwendet. In diesem Abschnitt werden kurz die Grundlagen dafür gegeben.

Ereignisgesteuerte Prozessketten (EPKs) wurden am Institut für Wirtschaftsinformatik im Saarland entwickelt, um Geschäftsprozesse zu modellieren. Die Sprachelemente sind Funktionen und Ereignisse die über Kanten alternierend verbunden sind. Des Weiteren gibt es Konnektoren, die den Steuerfluss aufspalten und wieder zusammenführen können. Beim Aufspalten des Prozesspfades mit dem AND Konnektor werden Prozesspfade beschrieben, die voneinander unabhängig sind und auch parallel ausgeführt werden dürfen. Beim XOR-Split wird an der Funktion unmittelbar davor während der Runtime geprüft, welcher Pfad hinter dem Konnektor zu nehmen ist. Beim OR Split Konnektor können auf die gleiche Weise mehrere Pfade genommen werden.

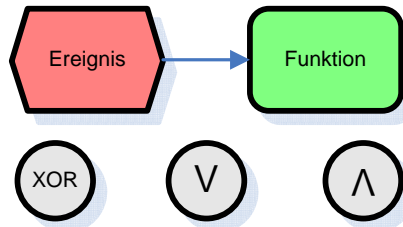


Abbildung 1: Grundelemente von EPKs

Yet another Workflow Language (YAWL) [HA05] ist eine Modellierungssprache für Workflows, die zusätzlich zum Modellierungswerkzeug (Editor) durch ein Workflow Management System (WfMS) implementiert wird. Die aus dem Editor entstandenen Modelle werden vom WfMS entgegengenommen, instanziiert und nutzergesteuert ausgeführt. Das WfMS berücksichtigt auch Daten- und Ressourcen-Aspekte, die modelliert und von der Workflow-Engine genutzt werden. Abbildung 2 zeigt die Sprachelemente. Elementare Bestandteile sind die Split- und Join-Operatoren, die direkt an die Aktivitäten notiert werden.



Abbildung 2: Grundelemente von YAWL

Die Unified Modeling Language [UML] stammt aus dem Bereich der Softwaretechnik. Mit UML lassen sich diverse Aspekte von Softwaresystemen mit unterschiedlichen Diagrammartentypen modellieren. Unter anderem gehören auch Aktivitätsdiagramme (AD) dazu, die auch zur Workflow-Modellierung geeignet sind. Die Grundelemente werden in Abbildung 3 veranschaulicht. Ebenfalls wie die bisher vorgestellten Sprachen sind auch ADs eine graphbasierte Sprache.

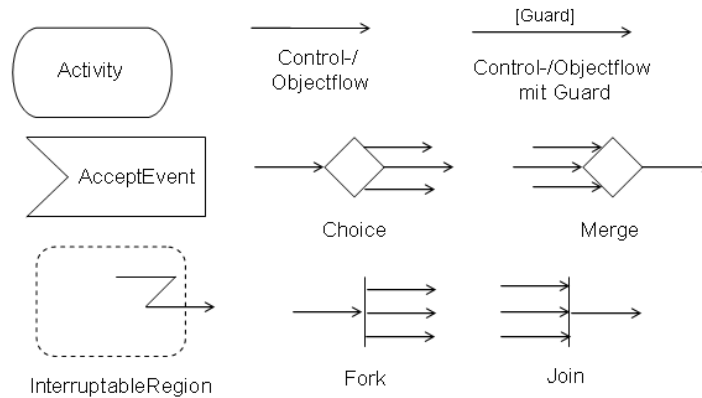


Abbildung 3: Grundelemente von UML Aktivitäts-Diagrammen

Aufgabenmodelle stammen aus dem Bereich der Human Computer Interaction (HCI) und beschreiben dort Handlungsabläufe, die ausdrücken wie der Mensch mit dem Computer interagiert. Sehr verbreitet sind hier die Concurrent Task Trees (CTT) [Pa00]. In [LPV01] werden Aufgabenmodell-Ansätze gegenübergestellt. Am Lehrstuhl Softwaretechnik der Universität Rostock wurde ein Workflow-Management System – das Task Tree Management System (TTMS) entwickelt, das auf hierarchischen Aufgabenmodellen ähnlich zu den CTTs basiert. Der Editor zum Erstellen der Workflow-Modelle stellt diese aber auch visuell ähnlich eines UML-Aktivitätsdiagramms dar. Zusammenhänge zwischen CTT und Aktivitätsdiagrammen sind in [BDF07] dargestellt. In Abbildung 4 ist der Editor zum Erstellen der Workflowmodelle abgebildet. Ein Workflow als Aufgabenbaum (ohne temporale Operatoren) ist links und die Aktivitätsdiagramm-ähnliche Darstellung ist dann rechts zu sehen.

In Abschnitt 3 wird nicht auf die Modellierung von Workflows in Form von Aufgabenbäumen eingegangen. Dort wird zunächst die Modellierung von Entscheidungen in den Sprachen EPK, UML ADs und YAWL betrachtet. In Abschnitt 4, wird wieder auf diese Modellierungsart bei der Betrachtung des TTMS eingegangen. Es wird gezeigt, wie die Interpretation der unterschiedlichen in Abschnitt 3 präsentierten Entscheidungsoperatoren dort umgesetzt ist.

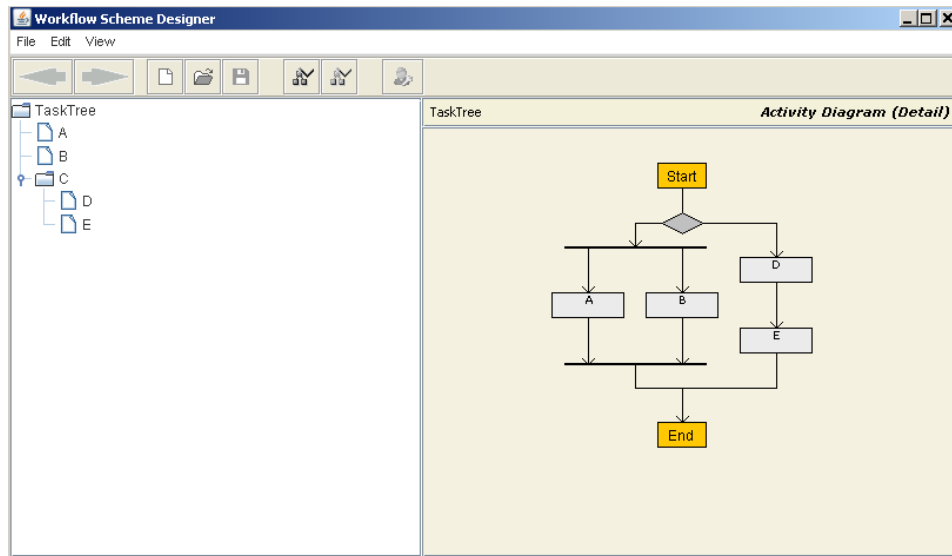


Abbildung 4: Aufgabenbaum links mit entsprechender AD ähnlicher Darstellung rechts

3 Modellierung von Entscheidungen

Es gibt verschiedene Arten von Entscheidungen, die während der Ausführung eines Workflows zur Runtime getroffen werden müssen. Wir unterscheiden die Arten in den folgenden Unterabschnitten: 1. können bzw. müssen Entscheidungen über den Ablauf der Aktivitäten im Geschäftsprozess getroffen werden, wenn die Reihenfolge während der Designtime noch nicht festgelegt wurde. 2. gibt es unterschiedliche Entscheidungsarten für alternativ auszuführende Aktivitäten im Prozess. Dafür gibt es unterschiedliche Entscheidungsoperatoren, die nachfolgend vorgestellt werden. 3. wird für EPKs das Konzept des Entscheidungsprozesses eingeführt, um auch komplexe Entscheidungen modellieren zu können. Schließlich sind 4. die unterschiedlichen Zeiten im Prozesslebenszyklus aufgelistet, in denen die entsprechenden Entscheidungen zu treffen sind.

3.1 Entscheidungen über den Prozessablauf

Es gibt Unterschiede bei Workflowmodellen bezogen auf die Ausführungsflexibilität der enthaltenen Aktivitäten. Es gibt Prozesse, die nach dem Prinzip *flexible by design* [SRM07] modelliert sind und dem Nutzer mehrere Ausführungsmöglichkeiten während der Runtime überlassen.

Nehmen wir an, dass eine Aktivitätsinstanz den Lebenszyklus aus dem UML-StateChart von Abbildung 5 hat, so dass die Aktivitäten initial in den Zustand *waiting* versetzt werden, dann gestartet und schließlich beendet werden können. Alternativ können Aktivitäten auch mit *skip* übersprungen werden.

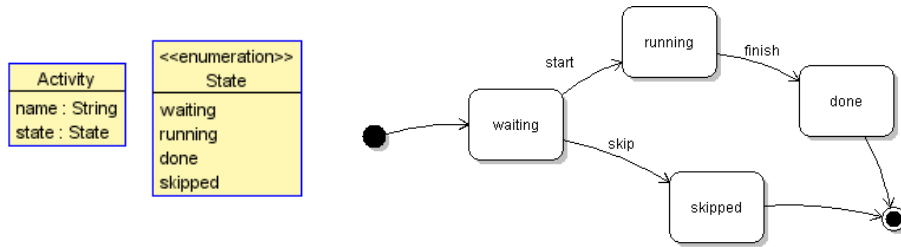


Abbildung 5: Lebenszyklus von Aktivitäten bzw. Aktivitätsinstanzen

Das EPK-Modell aus Abbildung 6 a) ist ein Beispiel für ein flexibles Ausführungsmodell. Hier können u.a. folgenden Ablaufreihenfolgen stattfinden ($start(A)$, $start(B)$, $finish(A)$, $finish(B)$), ($start(A)$, $finish(A)$, $start(B)$, $finish(B)$). Dagegen gibt es bei der restriktiven Prozessmodellierung aus Abbildung 6 b) nur eine Ausführungsreihenfolge ($start(A)$, $finish(A)$, $start(B)$, $finish(B)$). Das zweite Modell gibt dem Nutzer mehr Unterstützung, welche Aktivität als nächstes auszuführen ist, während das erste ihm mehr Flexibilität und damit mehr Entscheidungen abverlangt.

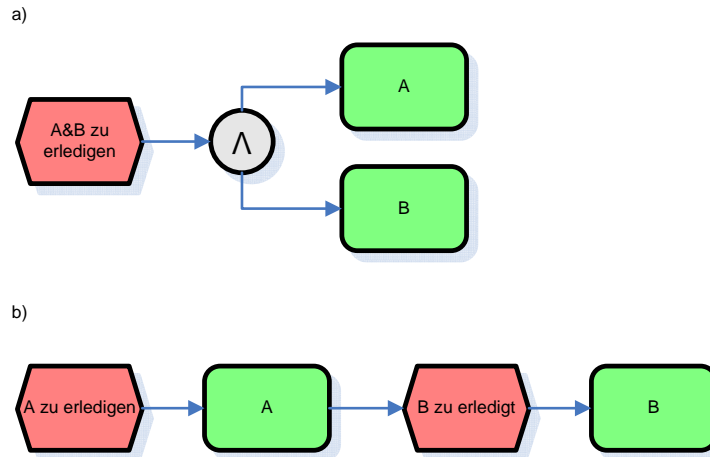


Abbildung 6: a) Flexibles Modell zur Erledigung von Aktivitäten A und B
b) restriktives Modell zur Erledigung von A und B

Es kann aber auch Situationen geben, bei denen man flexible Prozesse in der Form vom Modell aus Abbildung 6 a) haben möchte, die Flexibilität und damit den Entscheidungsraum des Nutzers für die Runtime aber etwas einschränken möchte. Der SEQ-Konnektor [ST05] ist ein Beispiel dafür. Er modelliert eine unabhängige Ausführung der nachfolgenden Aktivitäten, eine parallele Abarbeitung wird dabei jedoch untersagt.

In [ST05] wurde der SEQ-Konnektor im Zusammenhang mit einer explizit angegebenen Entscheidungsfunktion definiert, die unmittelbar vor Beginn der ersten Aktivität die komplette Ablaufreihenfolge der darauf folgenden Aktivitäten festlegt. Abbildung 7 a) zeigt ein Beispiel mit dem SEQ-Konnektor und Abbildung 7 b) den semantisch gleichen Prozessausschnitt mit XOR-Konnektoren.

Um den SEQ-Konnektor etwas flexibler zu definieren schlagen wir in Abbildung 7 c) deklarative Annotationen vor, die den Entscheidungsraum während der Runtime für flexible Prozessmodelle nicht so restriktiv einschränkt, wie die Modelle aus a) und b). In Abbildung 7 c) wird dabei mit Hilfe OCL-ähnlicher Formeln [OCL] gefordert, dass die Menge der laufenden Aktivitäten von A und B kleiner oder gleich 1 ist. Die OCL-Formel bezieht sich hierbei auf die Metaklasse *Activity* und die Enumeration *State* aus Abbildung 5. Mit dieser Notation wird die explizite Entscheidung „Über Abfolge entscheiden“ weggelassen und eine implizite Entscheidung des Nutzers zur Abfolge der Aktivitäten während der Runtime gefordert. Die Semantik ist somit gleich zum *Interleaved Parallel Routing Pattern* [DH01, S.9f]. Das Modell aus c) ist flexibler, weil hier vor der Abarbeitung der ersten Aktivität die Abfolge noch nicht feststeht, wohingegen bei a) und b) die Reihenfolge schon in der Funktion „Über Abfolge entscheiden“ festgelegt wird.

Es sind bei den deklarativen Annotationen auch andere temporale Beziehungen ausdrückbar. Beispielsweise kann man mit folgender OCL ähnlichen Formel fordern, dass wenn Funktion A läuft auch B laufen muss: *A.state=running implies B.state=running*.

Oder man fordert strikte oder auch logische Parallelität: *A.state=B.state*. Bei Workflows wird wohl häufiger die logische Parallelität vorkommen, so dass der Start und das Ende nicht exakt gleichzeitig geschehen müssen: Z.B. muss die Operation, die vom Chirurgen durchgeführt wird von der Schwester assistiert werden und umgekehrt. Strikte (realtime) Parallelität wird dabei wohl häufiger bei technischen Prozessen vorkommen. Weitergehende deklarative Ansätze zur Beschreibung von Workflows mit UML-Klassendiagrammen und OCL-Formeln sind in [BW09, Br09] zu finden.

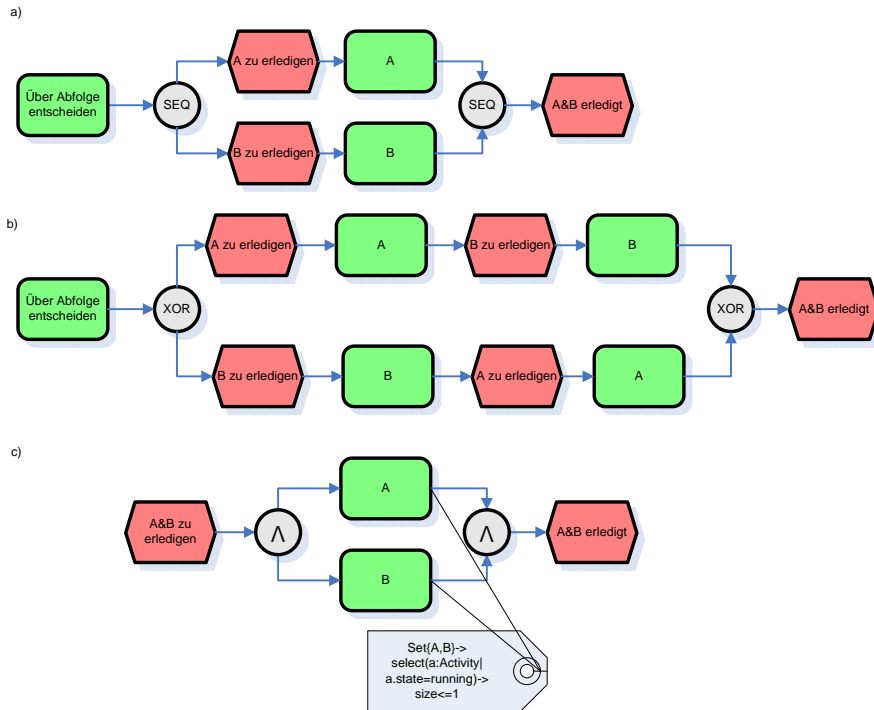


Abbildung 7: a) SEQ-Konnektor nach [ST05]
 b) SEQ-Semantik ausgedrückt mit XOR-Konnektoren
 c) SEQ-Semantik ausgedrückt mit deklarativen OCL-ähnlichen Annotation

3.2 Modellierung der ereignisbasierten und impliziten Entscheidungen

Bei den Prozess-Patterns [RH06] wird der *Deferred Choice* (Pattern 16) aufgeführt. Dieser unterscheidet sich von dem *Exclusive Choice* (Pattern 4) darin, dass die alternativ auszuführenden Aktivitäten der Umgebung angeboten werden. Die beiden Alternativen stehen in temporaler Konkurrenz zueinander und nachdem eine aktiviert wurde wird die andere aufgehoben bzw. deaktiviert. In UML-Aktivitätsdiagrammen kann dieser Sachverhalt wie in Abbildung 8 a) modelliert werden [WA04].

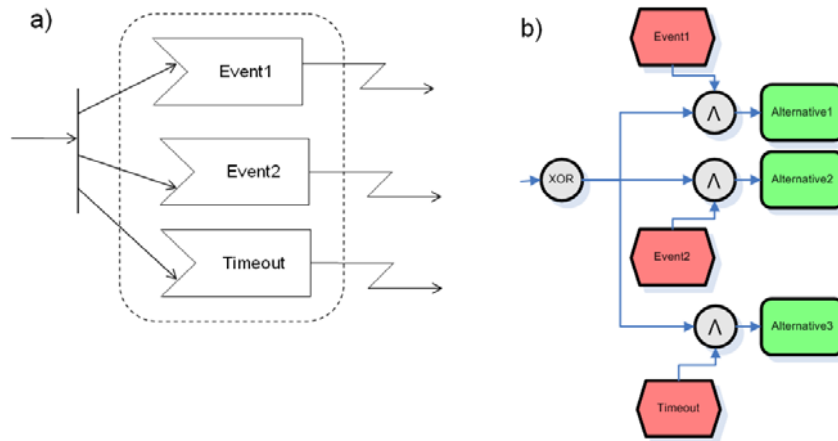


Abbildung 8: a) Modellierung des *Deferred Choice* mit UML ADs, b) Modellierung des *Deferred Choice* mit EPKs

Nach [RH06, S.133] ist der *Deferred Choice* in EPKs nicht modellierbar, weil Prozessabläufe an Konnektoren nicht verharren, sondern direkt weitergeführt werden. Außerdem bedarf es einer expliziten Entscheidungsfunktion vor XOR- bzw. OR-Split-Konnektoren [KNS92]. Lässt man diese zwei Aspekte außer acht, ist das *Deferred Choice* Pattern doch mit EPKs modellierbar. Es wird also keine explizite Entscheidungsfunktion mehr verlangt und Prozessabläufe können an Konnektoren verharren. Ein Beispiel ist in Abbildung 8 b) angegeben und eine ähnliche Modellierung wurde schon in [De02, Abb.2c)] verwendet. Die Ereignisse bei den EPKs werden dabei äquivalent zu den *AcceptEventActions* der UML Aktivitätsdiagramme verwendet. Der Ablauf verharret bei der Prozessarbeit also solange am XOR-Konnektor, bis das entsprechende Ereignis eingetroffen ist. Dann ist die Bedingung für den entsprechenden AND-Konnektor erfüllt und der Kontrollfluss kann auf dem zugehörigen Pfad weitergeleitet werden.

Nachdem die ereignisbasierte Entscheidung betrachtet wurde, kommen wir nun zur Modellierung der nutzerbasierten Entscheidung. Der *Deferred Choice* kann auch als eine nutzerbasierte Entscheidung interpretiert werden, bei der die Kriterien auf denen die Entscheidungen basieren im Modell nicht angegeben werden. Damit bleibt die Nutzerentscheidung unterspezifiziert und unterscheidet sich dadurch von der expliziten Entscheidung von Abschnitt 3.3. Das Verhalten des *Deferred Choice* mit der Nutzerauswahl ohne Angabe der Auswahlkriterien ist so auf der Website der Workflow-Patterns animiert [WfP].

Bei UML ADs kann für die Modellierung dieses Sachverhalts ein Choice-Operator ohne Angabe von Entscheidungskriterien in Form einer Entscheidungsfunktion und Guards für die Alternativen verwendet werden. Die Modellierung ist in Abbildung 9 a) angegeben.

In EPKs ist der *Deferred Choice* mit Nutzerauswahl ohne Angabe einer Entscheidungsfunktion ähnlich modellierbar, wie der Event-basierte *Deferred Choice* aus Abbildung 8 b). Es werden hierbei die Ereignisse weggelassen und die Kriterien zur Auswahl der Alternativen bleiben unterspezifiziert. Abbildung 9 b) zeigt diese Modellierung.

In YAWL wird für die Modellierung der Nutzerauswahl ohne Kriterien der *Condition Node* benutzt. Die alternativen Ausführungspfade werden dann mit den Aktivitäten, die mit den ausgehenden Kanten verbunden sind angegeben. Abbildung 9 c) gibt ein Beispiel dafür. Ein Event-Konzept ist in YAWL noch nicht realisiert, so dass der oben erwähnte Event-basierte *Deferred Choice* noch nicht ausdrückbar ist.

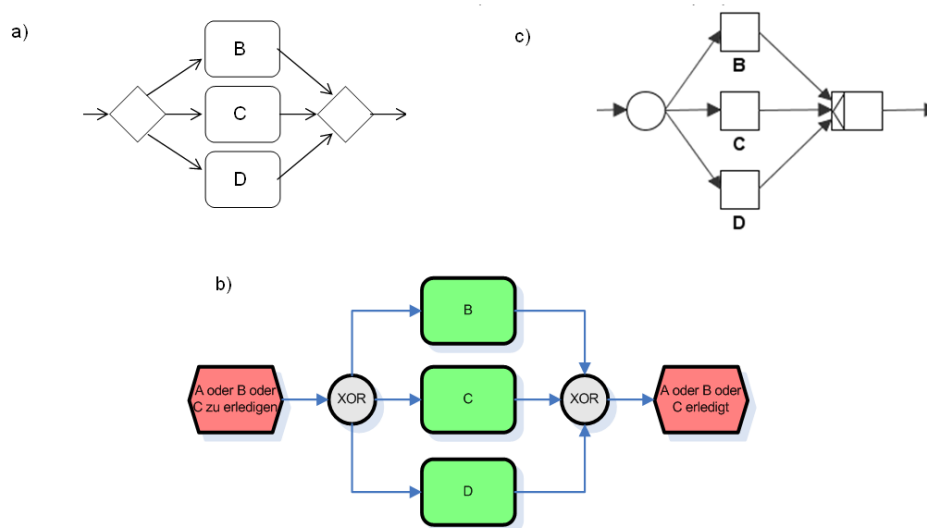


Abbildung 9: *Deferred choice* mit Nutzer Entscheidung a) UML AD, b) EPK, c) YAWL

3.3 Modellierung von expliziten Entscheidungen

Für die explizite Entscheidungsmodellierung benötigt man zwei Modellierungselemente, um zur Design-time die zur Runtime zu treffende Entscheidung zu modellieren. Zum einen muss ausgedrückt werden, was aktiv getan werden soll, um die Entscheidung zu treffen, z.B. Prüfen von Anträgen. Zum anderen müssen die Kriterien angegeben werden, die erfüllt sein müssen, um die entsprechenden alternativen Prozesspfade zu aktivieren. Aufgrund der angebotenen Alternativen kann der Nutzer dann die Entscheidung treffen und die dazugehörigen Pfade auswählen.

Bei UML ADs können explizite Entscheidungen mit Hilfe von Aktivitäten, dem Decision Node und Guards modelliert werden [UML, Abb.12.37] bzw. [UML, S. 359ff]. Die Aktivität, die zur Entscheidung beitragen soll (Entscheidungsfunktion) wird dabei direkt vor dem Decision Node modelliert. Hierbei ist zu beachten, dass die Bezeichnung der Aktivität zu einer Entscheidungsfunktion mit z.B. „prüfe“ oder „entscheide“ passt. An den ausgehenden Kanten hinter dem Decision Node werden Guards spezifiziert, die die Kriterien zur Aktivierung der Prozesspfade angeben. Ein Modell mit ADs ist in Abbildung 10 a) zu finden. Zusätzlich kann mit `<<DecisionInput>>` an den Decision Node eine Bedingung notiert werden, die ein wiederholtes Auswerten von Formeln in den Guards verhindert [UML, S.360].

Die explizite Entscheidung ist genau die Art der Entscheidungsmodellierung, die bei den EPKs nach [KNS92] bisher nur erlaubt war. Hierfür wurden Konnektoren XOR und OR in Verbindung mit einer Funktion, die unmittelbar vor dem Konnektor sein muss (Entscheidungsfunktion), verwendet. Des Weiteren sind die Ereignisse unmittelbar hinter den Konnektoren in das Entscheidungskonzept mit einzubeziehen. Sie stellen die Kriterien für die Auswahl dieser Pfadalternativen bereit, ähnlich zum Guard-Konzept bei den UML ADs. Abbildung 10 b) zeigt dafür ein Beispiel.

Bei YAWL gibt es die XOR- bzw. OR-Split-Operatoren, die in der Darstellung direkt an Aktivitäten geknüpft sind. Inhaltlich muss aber kein direkter Zusammenhang zwischen Aktivität und Entscheidung bestehen. Bei den Operatoren kann man über XPath-Formeln bzw. Prädikate angeben, welche Ausgangskante bzw. -kanten aktiviert werden sollen. Die Auswertung wird während der Runtime automatisch von der Workflow-Engine vorgenommen und basiert auf Daten, die vorher eingegeben wurden und über XPath abgefragt werden. In der Regel passiert die Angabe der relevanten Daten bei der Aktivität unmittelbar vor dem Operator, womit die visuelle Verknüpfung des Operators mit einer Aktivität in den meisten Fällen sinnvoll ist. Abbildung 10 c) stellt ein Beispiel dafür bereit, bei dem zu beachten ist, dass bei der Prüf-Aktivität die Ergebnisse abgefragt werden, auf der dann der darauffolgende Choice-Operator die Entscheidung automatisch vornimmt.

Bei YAWL Modellen sind auch UML-artige Guard Notationen zu finden (z.B. [LCH09]), welche aber lediglich als Kommentar dienen. Sie finden keine weitere Beachtung von der Workflow-Engine während der Runtime. Hier laufen alle expliziten Choices über das Datenmodell.

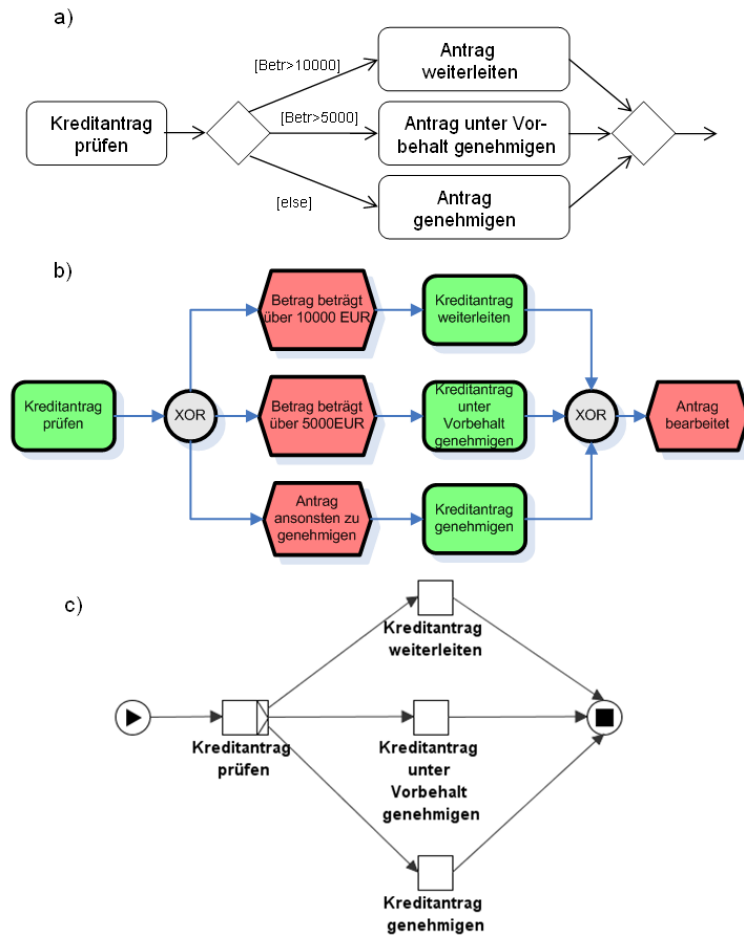


Abbildung 10: Explizite Entscheidung in folgenden Sprachen modelliert a) UML AD; b) EPK; c) YAWL

3.4 Modellierung von Entscheidungsprozessen

In [ST05] wurde in Abbildung 1 ein Entscheidungsprozess modelliert, der vier Entscheidungsaktivitäten umfasst. Die Abfolge der Aktivitätsabarbeitung ist in dem Modell nicht vorgegeben. Somit ist dieser Entscheidungsprozess nach dem Prinzip *flexible by design* (s. Abschnitt 3.1) spezifiziert.

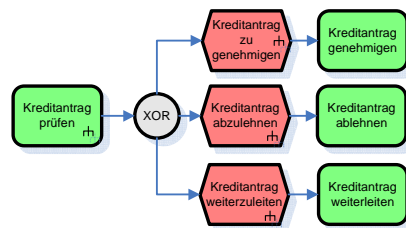
In diesem Unterabschnitt wird ein Konzept vorgestellt, um Entscheidungsprozesse zu modellieren, in denen die Ablaufreihenfolge der Entscheidungsfunktionen vorgegeben ist und die relevanten Prüfungen nicht flexibel in der Reihenfolge ausgeführt werden dürfen. Hierfür wird das Konzept der Entscheidungsfunktionen bei den EPKs, das in Unterabschnitt 3.3 vorgestellt wurde, um Entscheidungsprozesse erweitert.

Beim Konzept der Entscheidungsprozesse wird die Entscheidungsfunktion in einen Entscheidungsprozess mit Entscheidungs-Unterfunktionen untergliedert. Die Ereignisse dienen dem Nutzer als Wahlalternative während die Entscheidungsfunktion aktiv ist.

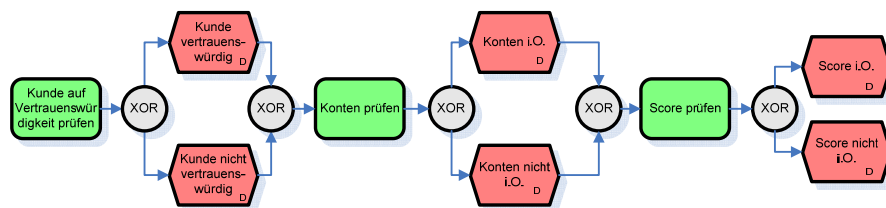
In Abbildung 11 a) wird ein Ausschnitt aus einem Prozess vorgestellt, in dem ein Kreditantrag geprüft und darüber zu entscheiden ist, ob dieser genehmigt, abgelehnt oder weitergeleitet werden muss. Sowohl die Funktion *Kreditantrag prüfen* als auch die darauffolgenden Ereignisse sind hierarchisch gegliedert, welches mit dem Hierarchie-symbol von UML ADs ausgedrückt wird.

Der Entscheidungsprozess ist in Abbildung 11 b) modelliert. Hier ist zu erkennen, dass die Prüf-Aktivitäten in der Abfolge nicht flexibel spezifiziert sind. In der Sequenz müssen hier drei Entscheidungsfunktionen abgearbeitet werden. Die Ereignisse hinter den XOR-Split-Konnektoren sind mit einem *D* gekennzeichnet, welches für *Decision* steht und bedeutet, dass diese Ereignisse für die Auswertung nach Beendigung des Entscheidungsprozesses herangezogen werden.

a)



b) Entscheidungsprozess: *Kreditantrag prüfen*



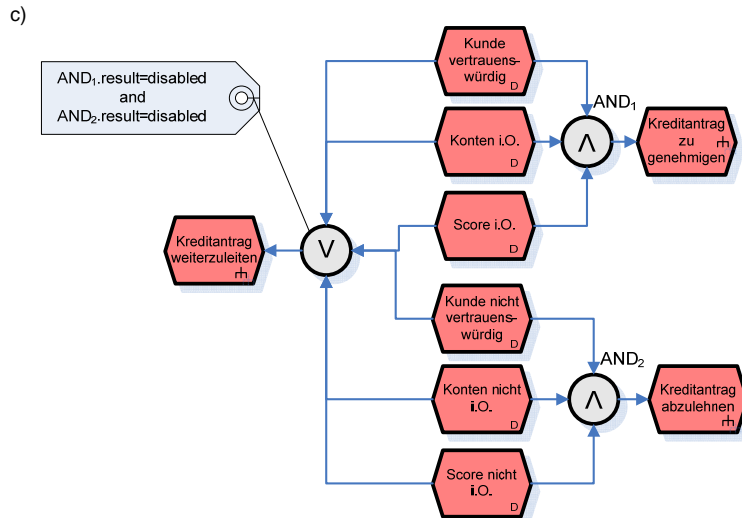


Abbildung 11: a) Übergeordneter Prozess; b) Entscheidungsprozess *Kreditantrag prüfen*;
c) Regeln zum Auswerten der Entscheidungsereignisse zu den übergeordneten Ereignissen

Die Auswertungsregeln sind dann in Abbildung 11 c) ausgedrückt. Hier sind alle *Decision-Ereignisse* aufgelistet und mit logischen Ausdrücken bzw. Konnektoren verbunden, die dann schließlich zu den strukturierten Events führen. Es wird hier 1. ausgesagt, dass alle drei Prüffunktionen positiv ausfallen müssen, damit ein Kreditantrag genehmigt wird. 2. wird spezifiziert, dass bei negativer Bewertung aller Prüffunktionen der Kreditantrag abgelehnt werden muss. 3. muss bei jedem anderen Fall der Antrag weitergeleitet werden. Hierfür wurde mit einer Bedingung an dem OR-Join ähnlich zu den C-EPC-Annotationen [RA07] vermerkt, dass wenn die AND₁ und AND₂-Konnektoren nicht schalten können, also als Resultat-Zustand *disabled* liefern, der Kreditantrag weiterzuleiten ist.

3.5 Phasen, in denen Entscheidungen zu treffen sind

Es existieren unterschiedliche Phasen bei Geschäftsprozessmodellen, in denen Entscheidungen getroffen werden müssen. Die Entscheidungsoperatoren geben an, in welcher Phase sie aufzulösen sind oder ob sie evtl. in die nächste Phase verschoben werden können. In den folgenden Punkten werden die Phasen aufgelistet, in denen sich Geschäftsprozessmodelle befinden können.

- **Buildtime/Designtime:** Es existieren für EPKs viele Referenzmodelle [Sch98], die u.a. für unterschiedliche Branchen entwickelt wurden und z.B. mit der C-EPC [RA07] modelliert sind. In den Referenzmodellen existieren diverse sogenannte konfigurierbare Konnektoren [RA07], die während der Buildtime für das konkrete Unternehmen aufgelöst werden können. Das Referenzmodell wird dadurch für das konkrete Unternehmen konfiguriert.

Während der Buildtime werden nur implizite Entscheidungen vom Prozessmodellierer bei Auflösung der konfigurierbaren Konnektoren gemacht. Diese haben damit den Charakter eines *Deferred Choice*, die während der Buildtime durch Designentscheidungen aufgelöst werden.

- **Instantiationtime:** In [BF08] und [APS09] wird die Instantiationtime als letzte Phase der Konfiguration der Workflow-Modelle vorgestellt, in der die Modelle für den konkreten Anwendungsfall z.B. für spezielle Kundenwünsche angepasst werden. Bei den EPKs können die konfigurierbaren Konnektoren nicht nur für Referenzmodelle angewendet werden, sondern auch durch die Instantiationtime-Operatoren bei der Instanziierung des Geschäftsprozesses [BF08].

Bei der Instantiationtime kann eine Unterscheidung von explizit und implizit zu treffenden Entscheidungen vorgenommen werden. Einerseits können vom WfMS bei der Instanziierung die Kriterien für die zu wählenden Prozesspfade abgefragt werden, was eine explizite Entscheidung bedeuten würde. Andererseits können die möglichen Pfadalternativen ohne Kriterien direkt zur Wahl gestellt werden, was eine Art *Deferred Choice* zur Instanziierung darstellt.

- **Runtime:** Nach der Instanziierung des Geschäftsprozesses befindet man sich in der Runtime. Die in den bisherigen Abschnitten betrachteten Entscheidungsoperatoren betreffen die Runtime. Es wird zwischen expliziten und impliziten Entscheidungen unterschieden.

Zusätzlich stellt sich während der Runtime auch noch die Entscheidung der Ablaufreihenfolge bei Prozessmodellen, die nach dem *flexible by design* Prinzip modelliert wurden. Sogar die Ablauflogik kann explizit mit einer Funktion zur Runtime entschieden werden [ST05, Abb. 4], wobei diese Modellierungsart in der Regel implizit bleibt.

4 Interpretation von Entscheidungsoperatoren im WfMS

In diesem Abschnitt wird das in Abschnitt 2 erwähnte TTMS (ein auf Aufgabenbäumen basierendes WfMS) näher vorgestellt. Für das WfMS wurden implizite und explizite Entscheidungsoperatoren implementiert. Im Unterschied zu YAWL laufen die Auswertungen der expliziten Entscheidungen aber nicht über das Datenmodell, sondern werden direkt über Conditions und Guards interpretiert. Es wird gezeigt, wie die Entscheidungsoperatoren interpretiert und dem Endnutzer präsentiert werden.

4.1 Interpretation von impliziten Entscheidungen zur Runtime

Implizite Entscheidungen werden z.B. bei *flexible by design* Modellen für die Abfolge der Aktivitäten getroffen. Hierbei sind die meisten Aktivitäten aktiviert und werden vom WfMS dem Nutzer als startbar angezeigt. Auch parallele Abarbeitung ist möglich, wenn der Nutzer neben einer schon laufenden Aktivität eine weitere startet. Interpretationen von deklarativen Annotationen, werden vom WfMS noch nicht unterstützt, so dass Modelle wie von Abbildung 7 c) noch nicht ausdrückbar sind.

Implizite Entscheidungen über Prozessalternativen sind über den *Deferred Choice* möglich und mit dem TTMS modellierbar. Beim Modell aus Abbildung 4 ist der erste Operator ein *Deferred Choice*. Der weitere Operator ist der Fork, der die Unabhängigkeit von Aktivität A und B ausdrückt. Nach Instanziierung sind drei Aktivitäten aktiviert und damit startbar: A, B und D.

UserApp - Workspace (Administrator)

The screenshot displays the 'UserApp - Workspace (Administrator)' interface. At the top, there are navigation tabs: 'Workspace' (selected), 'Project Center', 'User Center', 'Resource Center', and 'Logout'. Below these are sub-tabs: 'DETAILS', 'TOOLS', 'EXECUTORS', and 'DOCUMENTS'. The main content area is a form for configuring a task. The form fields are as follows:

Name	D
Description	
ID	T1.3.1
State	enabled
Requester	Administrator
Executor	Administrator
Creationtime	Today - 04:57:45 PM
ExecutionOrder:	
ExecutionType:	Sequence
Assigned Tools	
Is Explicit	false
Is explicit Selected	false
Guard	

At the bottom of the form, there are three buttons: 'Start', 'Save', and 'Complete'. To the left of the form is a 'TaskTree' view showing a hierarchy of tasks: 'Projects' > 'TaskTree' > 'A', 'B', 'C', 'D', 'E'. To the right of the form, there are tabs for 'DECISION' and 'PROCESS DEFINITIONS'.

Abbildung 12: Das User Interface vom Task-Tree-Management-System (TTMS)

In Abbildung 12 ist die GUI des WfMS zu sehen, in dem die startbaren Aktivitäten grün angegeben sind. Aus Sicht des Nutzers ist hier nicht ersichtlich in welchem Verhältnis A, B und D zueinander stehen. Es könnten alle Aktivitäten unabhängig zueinander stehen oder mit einem *Deferred Choice* miteinander verknüpft sein. Startet der Nutzer nun Aktivität B, wird D automatisch deaktiviert. Wird jedoch D gestartet, werden A und B deaktiviert. In Abbildung 12 ist Aktivität D ausgewählt und startbar, was auch an dem aktiven Start-Button erkennbar ist.

4.2 Interpretation von expliziten Entscheidungsfunktionen zur Runtime

Um die expliziten Entscheidungen im WfMS nutzen zu können, müssen diese mit dem Editierwerkzeug modelliert werden. Abbildung 13 zeigt ein solches Modell mit einem expliziten XOR-Entscheidungsoperator inkl. Guards. Es wird eine Kreditantragsabwicklung modelliert. Zunächst findet beim Entscheidungsoperator eine Prüfung statt. Die Bezeichnung der Entscheidungsaktivität ist dort hinterlegt. Im Anschluss werden die Kriterien zur Pfadauswahl in den Guards spezifiziert.

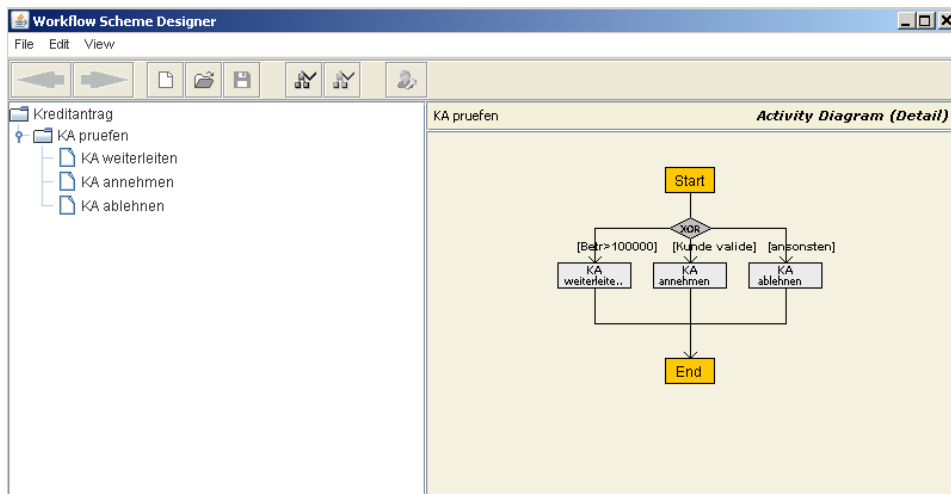


Abbildung 13: Spezifikation des expliziten Entscheidungsoperators im Editor

Abbildung 14 zeigt den Prozess während der Runtime im WfMS. Im rechten Frame ist unter den Buttons *Decision* und *Process Definitions* die Auswahl der alternativen Kriterien zu finden. Die Auswahl findet über Radio Buttons statt und garantiert dadurch eine exklusive Auswahl. Es wurde in diesem Fall das Kriterium *Kunde valide* ausgewählt, womit eine Aktivierung der dazugehörigen Aktivität erfolgt.

Beim expliziten OR-Operator werden dem Nutzer Checkboxes zur Auswahl bereitgestellt. Hier können dann mehrere Kriterien ausgewählt und Prozesspfade aktiviert werden.

UserApp - Workspace (Administrator)

The screenshot displays the 'UserApp - Workspace (Administrator)' interface. At the top, there are navigation tabs: 'Workspace', 'Project Center', 'User Center', 'Resource Center', and 'Logout'. The main area is divided into several sections:

- Left Panel (TaskTree):** A tree view showing a hierarchy of projects and tasks. The 'Kreditantrag' folder is expanded, showing sub-tasks: 'KA pruefen', 'KA weiterleiten', 'KA annehmen', and 'KA ablehnen'.
- DETAILS Panel:** A form for configuring the selected task 'KA pruefen'. Fields include:
 - Name: KA pruefen
 - Description: (empty)
 - ID: T2.1
 - State: waiting
 - Requester: Administrator
 - Executor: Administrator
 - Creationtime: Today - 07:42:37 PM
 - ExecutionOrder: 1x2x3
 - ExecutionType: (empty)
 - Assigned Tools: (empty dropdown)
 - Is Explicit: false
 - Is explicit Selected: false
 - Guard: (empty)
- DECISION PROCESS DEFINITIONS Panel:** A panel for defining decision logic. It includes:
 - Decision: KA pruefen
 - is made: false
 - Type: XOR
 - Three radio button options:
 - Betr>100000 (KA weiterleiten)
 - Kunde valide (KA annehmen)
 - ansonsten (KA ablehnen)
 - A 'Save Decisions' button.
- Bottom Panel:** Contains 'Start', 'Save', and 'Complete' buttons.

Abbildung 14: Auswertung des expliziten Entscheidungsoperators im TTMS

4.3 Interpretation von Instantiationtime Entscheidungen

In [BF08] wurde die Instantiationtime als Phase vorgestellt, in der die letzte Konfiguration vom WfMS vorgenommen wird. Der explizite XOR-Entscheidungsoperator aus Abbildung 13 wurde für diesen Unterabschnitt durch ein Instantiationtime XOR-Operator ersetzt. Bei der Erzeugung des Prozesses erfolgt die Auswertung dieses Operators und es erscheint der Auswertungsbildschirm von Abbildung 15. Hier kann das entsprechende Kriterium ausgewählt werden, womit der Operator aufgelöst wird. Zusätzlich kann die Entscheidung auch in die Runtime verlagert werden.

Choice: KA pruefen

	Guard	ID	Number	Name	Description
<input type="radio"/>	ansonsten	Thu Sep 10 18:06:41 CEST 2009	3	KA ablehnen	
<input checked="" type="radio"/>	Kunde valide	Thu Sep 10 18:06:06 CEST 2009	2	KA annehmen	
<input type="radio"/>	Betr>100000	Thu Sep 10 18:05:03 CEST 2009	1	KA weiterleiten	
<input type="radio"/>	Decision in Runtime				

Save Instantiation Decisions

Abbildung 15: Auswertung des Instantiationtime-XOR-Operators im TTMS

5 Zusammenfassung und Ausblick

In diesem Artikel wurde umfassend auf Entscheidungen eingegangen, die zum einen die Ablauflogik der Workflows und zum anderen Entscheidungen zur Auswahl von Prozessalternativen betreffen. Diese Betrachtungen wurden für die unterschiedlichen Prozesssprachen EPK, UML AD und YAWL durchgeführt, die damit auch verglichen wurden. Für EPKs wurden Vorschläge unterbreitet, um auch den *Deferred Choice* ausdrücken zu können. Außerdem wurde für komplexe Entscheidungen in EPKs das Konzept der Entscheidungsprozesse eingeführt, die mehrere Entscheidungsfunktionen umfassen und in einer festgelegten Reihenfolge ablaufen sollen.

Es wurden deklarative Annotationen in OCL-Syntax vorgeschlagen, um temporale Beziehungen zwischen Aktivitäten (z.B. das *Interleaved Parallel Routing Pattern*) besser modellieren zu können. Zusätzliche temporale Beziehungen wurden vorgestellt, wie z.B. die Forderung nach der (logischen) Parallelität, die in bisherigen imperativen Prozessmodellen nicht ausdrückbar waren.

Die in [BF08] diskutierten Instantiationtime-Operatoren wurden im TTMS implementiert. Des Weiteren wurden die expliziten Entscheidungsfunktionen umgesetzt, die in Form von Conditions und Guards zur Designtime die Entscheidungskriterien festlegen können. Die implizite Entscheidung (*Deferred Choice*) war dagegen bereits Bestandteil vom TTMS.

Für weitere Arbeiten ist die Anbindung des UML-/OCL-Modellierungswerzeugs *USE* (A UML-based Specification Environment) [USE, GBR07] an das Workflow Management System TTMS wünschenswert, um deklarative Aspekte an Workflow-Abläufen überprüfen zu können. USE kann die Daten aus dem WfMS abfragen und in sein eigenes Objektmodell überführen und fortlaufend aktualisieren. Dann kann USE als Monitoring Werkzeug für Workflows benutzt werden. Beziehungen zwischen Aktivitäten können mit OCL-Invarianten deklarativ ausgedrückt werden [Br09, BW09], die dann von USE während der Runtime überprüft werden. Mit dem *Class-Invariant-View* werden dann die verletzten Invarianten hervorgehoben und mit dem *Evaluation Browser* als OCL-Expression-Debugger kann der Grund der Verletzung gefunden werden.

Außerdem stellt USE schon diverse statistische Views bereit und OCL kann in USE als Anfragesprache benutzt werden, so dass Process-Mining Techniken in Verbindung mit diesem Werkzeug angewendet werden können.

Literaturverzeichnis

- [APS09] van der Aalst, W.; Pesic, M.; Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2):99–113, 2009.
- [BF08] Brüning, J.; Forbrig, P.: Methoden zur adaptiven Anpassung von EPKs an individuelle Anforderungen vor der Abarbeitung, EPK-Workshop2008, Saarbrücken, CEUR-WS 420
- [BFD07] Brüning, J.; Dittmar, A.; Forbrig, P., Reichart, D.: Taking SW Engineers on Board: Task Modelling with Activity Diagrams, EIS2007, Salamanca, LNCS 4940, 2008
- [BW09] Brüning, J.; Wolff, A.: Declarative Models for Business Processes and UI Generation using OCL, Models2009, Denver, OCL-Workshop, Electronic Communications of the EASST, <http://eceasst.cs.tu-berlin.de/index.php/eceasst/issue/archive>
- [Br09] Brüning, J.: Declarative Workflow Modeling with UML Class Diagrams and OCL. BPSC2009, Leipzig, LNI-P 147, 2009, S. 227-228
- [De02] Dehnert, J.: Making EPCs fit for Workflow Management, EPK-Workshop2002, Trier, http://www.wiso.uni-hamburg.de/fileadmin/WISO_FS_WI/EPK-Community/epk2002-proceedings.pdf, S.51-70 (besucht: 14.9.09)
- [DH01] Dumas, M.; ter Hofstede, A.: UML Activity Diagrams as a Workflow Specification Language. *Proceedings of the International Conference on the Unified Modeling Language (UML)*, Toronto, Canada, Springer, 2001.
- [GBR07] Gogolla, M.; Büttner, F.; Richters, M.: USE: A UML-Based Specification Environment for Validating UML and OCL. *Science of Computer Programming*, 69:27-34, 2007.
- [HA05] ter Hofstede, A.; van der Aalst, W.: YAWL: Yet Another Workflow Language. *Information Systems* 30(4), 2005, S. 245-275
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)" Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992.
- [LPV01] Limbourg, Q., Pribeanu, C., Vanderdonckt, J.: Towards Uniformed Task Models in a Model-Based Approach. DSV-IS2001, LNCS 2220, 2001.
- [LCH09] La Rosa, M.; Clemens, S.; ter Hofstede, A.: The Order Fulfilment Process Model (Appendix A1). In: ter Hofstede, H.; van der Aalst, W.; Russell, N.; Adams, M. (eds.) *Modern Business Process Automation: YAWL and its Support Environment*. Springer, 2009

- [OCL] OMG OCL Specification v.2.0; <http://www.omg.org/spec/OCL/2.0/PDF> (besucht: 14.9.09)
- [Pa00] Paterno, F.: Model-Based Design and Evaluation of Interactive Applications. Springer Verlag, 2000.
- [RA07] Rosemann, M.; van der Aalst, W.: A configurable reference modelling language. Information Systems 32(1), S. 1-23, 2007.
- [RH06] Russel, N.; ter Hofstede, H; van der Aalst, W.; Mulyar, N.: Workflow Patterns : A Revised View. BPM Center Report BPM-06-22, BPMcenter.org, 2006.
- [Sch98] Schütte, R.: Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Gabler, Wiesbaden, 1998, Diss. Univ. Münster
- [SRM07] Schonenberg, M.H.; Mans, R.S.; Russell, N.C.; Mulyar N.A.; v. d. Aalst W.M.P.: Towards a taxonomy of process flexibility (extended version). BPM Center Report BPM-07-11, BPMcenter.org, 2007
- [ST05] Scheer, A.-W.; Thomas, O: Geschäftsprozessmodellierung mit der Ereignisgesteuerten Prozesskette. Das Wirtschaftsstudium 34, Nr. 8-9, S. 1069-1078. 2005.
- [UML] OMG UML Superstructure Specification v.2.1.2 <http://www.omg.org/docs/formal/07-11-02.pdf>, 2007, S.295-418
- [USE] A UML-based Specification Environment, Universität Bremen, Lehrstuhl Datenbanken: <http://www.db.informatik.uni-bremen.de/projects/use/> (besucht: 14.9.09)
- [WfP] Workflow Patterns, WCP16: Deferred Choice Pattern http://www.workflowpatterns.com/patterns/control/state/wcp16_animation.php (besucht: 14.9.09)
- [WA04] Wohead, P.; van der Aalst, W.; Dumas, M.; ter Hofstede, A.; N. Russell: Pattern-based Analysis of UML Activity Diagrams. BETA Working Paper Series, WP 129, Eindhoven University of Technology, Eindhoven, 2004.

Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten – Linguistische Analyse und Vorschlag eines Lösungsansatzes

Patrick Delfmann, Sebastian Herwig, Łukasz Lis

Westfälische Wilhelms-Universität Münster
European Research Center for Information Systems (ERCIS)
Leonardo-Campus 3, 48149 Münster
{delfmann | herwig | lis}@ercis.uni-muenster.de

Abstract: Eine kritische Voraussetzung für den erfolgreichen Einsatz von fachkonzeptionellen Modellen ist ihre Verständlichkeit und Vergleichbarkeit. Modelladressaten müssen in die Lage versetzt werden, den mit den Modellen vermittelten Inhalt eindeutig zu erkennen. Dies impliziert das Vorhandensein eines gemeinsamen Begriffsverständnisses unter den Modellierern. Die Herstellung eines solchen gemeinsamen Begriffsverständnisses stellt für Prozessmodelle im Allgemeinen und Ereignisgesteuerte Prozessketten (EPK) im Speziellen eine besondere Herausforderung dar. Ursache sind Benennungspraktiken in Prozessmodellen, die sich nicht in einfachen Ein-Wort-Bezeichnungen, sondern ganzen Satzfragmenten für Aktivitäten, oder im speziellen Fall der EPK auch für Ereignisse, äußern. Der vorliegende Beitrag präsentiert die Ergebnisse einer Bezeichnungsanalyse von 4805 EPKs eines konkreten Modellierungsprojekts, die unter Nutzung expliziter Namenskonventionen erstellt wurden. Die Ergebnisse zeigen, dass die reine Existenz solcher Konventionen zur Sicherstellung der Modellverständlichkeit und -vergleichbarkeit von EPKs nicht ausreicht. Als Alternative wird ein linguistischer Ansatz vorgestellt, der die Bezeichnung mit Satzfragmenten explizit berücksichtigt, standardisiert und automatisiert durchsetzt.

1 Motivation

Eine notwendige Bedingung für die Nutzbarkeit von fachkonzeptionellen Modellen ist nicht nur ihre syntaktische Korrektheit, sondern auch ihre semantische Vergleichbarkeit. Letztere sicherzustellen ist ein äußerst ambitioniertes Unterfangen, wenn die Modelle von unterschiedlichen Personen entwickelt werden. Dies ist allerdings häufig der Fall, insbesondere im Rahmen umfänglicher Modellierungsvorhaben. Empirische Studien zeigen, dass auf diese Weise entwickelte Modelle sich in ihren Bezeichnungen erheblich unterscheiden können, auch dann, wenn sie den gleichen Betrachtungsgegenstand adressieren [HS06]. In der Folge werden gleiche Sachverhalte von unterschiedlichen Modelladressaten ggf. unterschiedlich bewertet und umgekehrt. Dieses Phänomen wird allgemein als Namenskonflikt bezeichnet [BL84]. Im Extremfall können Namenskonflikte dazu führen, dass Modelle unbrauchbar werden. Zur Vermeidung von Namenskonflikten

ist konsequenterweise ein einheitliches Begriffsverständnis bei der Gestaltung fachkonzeptioneller Modelle unerlässlich.

Ziel dieses Beitrags ist es, anhand der Analyse einer umfassenden Modellbasis von 4805 Ereignisgesteuerten Prozessketten (EPK [KNS92]), die im Rahmen eines großen Modellierungsprojekts entstanden sind, zu zeigen, wie diese sich speziell hinsichtlich ihrer Benennung darstellen, und in wieweit diese Benennungspraxis Namenskonflikte begünstigt. Ausgehend von den Ergebnissen wird untersucht, welche Ansätze zur Lösung von Namenskonflikten existieren, und ob diese sich zur Anwendung auf EPKs eignen.

Hierfür wird zunächst in Abschnitt 2 die verwendete Datenbasis vorgestellt und der Aufbau der Analyse detailliert erläutert. Die Analyseergebnisse werden bewertet, und es werden Problembereiche von namenskonfliktauslösenden Bezeichnern identifiziert. In Abschnitt 3 werden existente Ansätze dahingehend untersucht, ob sie in der Lage sind, Namenskonflikte in EPKs aufzulösen oder zu verhindern. In Abschnitt 4 wird ein alternativer Ansatz vorgestellt, der die Besonderheiten der Benennungspraxis in Prozessmodellen im Allgemeinen und EPKs im Speziellen explizit berücksichtigt. Weiterhin wird kurz erläutert, wie sich die in Abschnitt 2 identifizierten Probleme vermeiden lassen. Der Beitrag schließt mit einem Fazit und einem Ausblick auf weiteren Forschungsbedarf in Abschnitt 5.

2 Analyse von Benennungspraktiken in EPKs

Als Vertreter fachkonzeptioneller Prozessmodellierungssprachen weisen EPKs eine besondere Herausforderung in Bezug auf die Benennung von Modellelementen auf. Dies äußert sich darin, dass bei der Benennung von Modellelementen wie bspw. Prozessfunktionen und -ereignisse weniger einzelne Begriffe als vielmehr ganze Satzphrasen Verwendung finden. Ausgehend von dieser Annahme wird nachfolgend anhand einer umfassenden Modellbasis aufgezeigt, wie eine derartige Form der Benennung insbesondere bei EPKs ausgestaltet ist und welche Implikationen sich hieraus in Bezug auf das Auftreten von Namenskonflikten abzeichnen. Speziell wird erstens untersucht, wie viele Wörter im Durchschnitt für die Bezeichnung eines Modellelements verwendet werden. Je mehr Wörter in eine Bezeichnerphrase eingehen, desto größer ist die Gefahr der Entstehung von Namenskonflikten. Zweitens wird ermittelt, wie die verwendeten Satzphrasen hinsichtlich ihrer Struktur variieren (z. B. <Verb, Imperativ> <Substantiv, Akkusativ> vs. <Substantiv, Akkusativ> <Verb, Infinitiv> als Bezeichner für Funktionen). Auch hier begünstigt eine hohe Variation der Strukturen Namenskonflikte. Der Konflikt „Rechnung prüfen“ und „Prüfe Rechnung“ lässt sich bspw. nicht ohne weiteres automatisiert auflösen. Drittens werden die Bezeichner hinsichtlich der verwendeten konkreten Wörter untersucht. So kann z. B. die synonyme oder homonyme Verwendung von Einzelwörtern ebenfalls zu Namenskonflikten führen (z. B. „Rechnung“ vs. „Faktura“ oder „Rechnung (Beleg)“ vs. „Rechnung (Kalkulation)“).

2.1 Datenbasis

Als Grundlage der explorativ angelegten Analyse von Bezeichnungspraktiken in Prozessmodellen dient eine Modellbasis, die im Rahmen eines umfangreichen, vier Monate dauernden Modellierungsprojekts einer großen deutschen öffentlichen Einrichtung entwickelt wurde. Als Zielstellung des Modellierungsprojekts galt es, eine umfassende Erhebung und Dokumentation des Istzustands der gesamten Prozesslandschaft dieser Einrichtung durchzuführen. Zu diesem Zweck wurde ein Gruppe von mehr als 50 Modellierern etabliert, welche der Systematisierung der Architektur integrierter Informationssysteme (ARIS) [Sc00] folgend Modelle der Daten-, Funktions-, Organisations-, Leistungs- und insbesondere Prozesssicht entwickelte. Hierzu wurden Teilbereiche definiert, die verteilt von einzelnen Modellierern bearbeitet wurden. Als Modellierungsplattform wurde der ARIS Business Architect verwendet.

Für die Zwecke der Prozessmodellierung wurden neben der Modellierungstechnik der Wertschöpfungskettendiagramme maßgeblich EPKs verwendet. Einzelne Prozesse wurden hierbei auf verschiedenen Abstraktionsebenen modelliert und über Hinterlegungen miteinander verknüpft.

Zur Sicherung der Qualität und Vergleichbarkeit der Modelle wurden im Rahmen des Modellierungsprojekts Namenskonventionen etabliert. Hierzu wurden in einem ersten Schritt vor der Modellierung sowohl für die Benennung von Modellelementen zu verwendende Begrifflichkeiten in Form eines Glossars als auch grammatikalische Benennungsstrukturen spezifiziert. Das Glossar enthielt die im betrachteten Modellierungsszenario gültigen Bezeichner sowie zur Explikation derer Semantik eine detaillierte Beschreibung. Die Bezeichner umfassten ausschließlich Fachbegriffe, d. h. die betriebswirtschaftlich relevanten Objekte wie z. B. „Rechnung“, „Beleg“, „Lager“, „Teil“, „Baugruppe“ etc. Grammatikalische Bezeichnungsstrukturen, sogenannte Phrasenstrukturen, wurden mit dem Ziel vorgegeben, eine einheitliche Bezeichnerstruktur auch hinsichtlich der Satzsyntax sicherzustellen. Für Funktionen in EPKs wurde die grundlegende Phrasenstruktur <Substantiv, Akkusativ> <Verb, Infinitiv> und für Ereignisse <Substantiv, Nominativ> <konjugiertes Hilfsverb „sein“> <Verb, Partizip Perfekt> vorgegeben.

Die spezifizierten Namenskonventionen wurden allen Modellierern begleitend zum gesamten Prozess der Modellierung in Form eines textuellen Methodenhandbuchs, d. h. in nicht formalisierter Dokumentform, zur Verfügung gestellt. Vorgabe war es, die Konventionen bei der Erstellung sämtlicher Modelle des Projekts einzuhalten.

Weitere Konventionen, die nicht die Benennung von Elementen adressieren, existierten in Form von grafischen Regeln zur Vereinheitlichung der Anordnung von Modellelementen relativ zueinander sowie Syntaxkonventionen für die vereinheitlichte Verwendung von Modellelementtypen. Zu letzteren zählten bspw. eine Beschränkung zulässiger Ressourcentypen sowie eine Anpassung der Kontrollflusssyntax der EPK, die das Aussparen von Ereignissen fordert, welche innerhalb des Kontrollflusses zwei Funktionen direkt verbinden (sog. „Trivialereignisse“).

2.2 Vorgehen bei der Analyse

Die im Rahmen des Modellierungsprojekts entwickelte Modellbasis liegt strukturiert als eine Modelldatenbank im ARIS Business Architect vor. Unter Verwendung einer Exportschnittstelle wurden die in den vorliegenden 4805 EPK-Modellen verwendeten Elementbezeichner von Prozessfunktionen (13935) und -ereignissen (13381) extrahiert und in strukturierter Form zur weiteren Analyse außerhalb der Modellierungsumgebung bereitgestellt.

Zur strukturellen Analyse der Bezeichner einzelner Modellelemente in EPKs wurde das computerlinguistische Verfahren des Part-of-Speech (POS) Tagging verwendet. Mittels POS Tagging ist es möglich, konkrete Wörter auf die zugehörige Wortart zurückzuführen. Exemplarisch kann die Phrase „Rechnung prüfen“ auf die einzelnen Wortarten der beiden Wörter „Rechnung“ und „prüfen“ und somit auf ihre Struktur <Substantiv> <Verb> zurückgeführt werden. Im Rahmen der Analyse wurde das POS-Tagging-System *TreeTagger* [Sc94] verwendet.

2.3 Identifizierte Bezeichnungspraxis

Die Analyse hat u. a. gezeigt, dass zur Bezeichnung von Modellelementen in EPKs Satzstrukturen mit hauptsächlich 2-7 Wörtern bevorzugt werden (vgl. Abbildung 1).

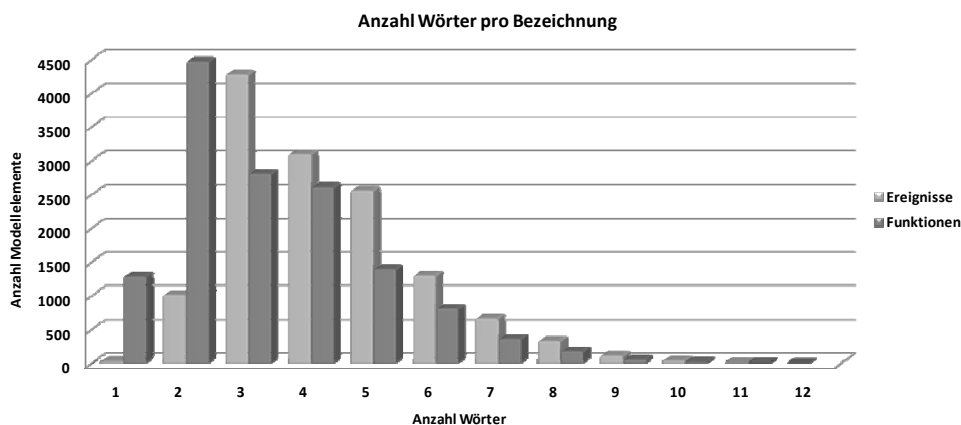


Abbildung 1: Bezeichnungsstruktur der analysierten EPKs

Des Weiteren unterscheiden sich die Satzstrukturen selbst erheblich (vgl. Tabelle 1). So wurden bspw. 1009 Ereignisse von insgesamt 13381 mit Bezeichnungen versehen, die genau zwei Wörter enthalten. Trotz dieser geringen Komplexität der Bezeichnerphrasen ergaben sich 22 unterschiedliche Phrasentypen. Je länger sich eine Bezeichnerphrase gestaltete, umso diverser zeigten sich auch die verwendeten Phrasentypen (bspw. existierten 34 Ereignisse mit 10-Wort-Bezeichnern und entsprechend 33 unterschiedliche Phrasentypen). Bezeichnungen von Funktionen verhielten sich ähnlich.

Anzahl Wörter	Ereignisse		Funktionen	
	Anzahl	Anzahl unterschiedlicher Strukturen	Anzahl	Anzahl unterschiedlicher Strukturen
1	26	6	1276	4
2	1009	22	4466	40
3	4271	144	2799	119
4	3091	330	2605	286
5	2555	549	1394	398
6	1291	566	802	399
7	662	445	355	273
8	327	255	168	142
9	107	97	48	44
10	34	33	18	18
11	5	5	4	4
12	3	3	0	0
Summe	13381	2455	13935	1727

Tabelle 1: Phrasenstrukturen der analysierten EPKs

2.4 Identifizierte Problembereiche

Die große Anzahl von verwendeten unterschiedlichen Phrasenstrukturen stellt einen Hinweis auf mögliche Inkonsistenzen in der Bezeichnung der Modellelemente dar. Um diese vermuteten Probleme zu untersuchen, wurden im zweiten Schritt der Untersuchung die einzelnen Bezeichnungen manuell analysiert und sechs Problembereiche festgestellt. Diese werden im Folgenden anhand einiger ausgewählter Beispiele vorgestellt. Es sei angemerkt, dass die Problembereiche nicht das Ergebnis eines spezifischen Clusterungs- oder Klassifizierungsverfahrens sind, sondern vielmehr den Eindruck widerspiegeln, der sich den Autoren im Rahmen der Analyse geboten hat.

Synonyme

In vielen Fällen werden Begriffe verwendet, die eine semantische Überschneidung aufweisen. Darüber hinaus werden durch einen inkonsequenten Einsatz von Abkürzungen Mehrdeutigkeiten geschaffen, die nur schwer oder gar nicht auflösbar sind. Als Beispiel können hier folgende drei Ereignisse genannt werden: „Dok erstellt“, „Dokument ist generiert“ und „Dokumentation ist unvollständig“. Es fällt auf, dass die Abkürzung „Dok“ sowohl zu dem Begriff „Dokument“ als auch „Dokumentation“ aufgelöst werden kann. Ferner weisen die Begriffe „erstellen“ und „generieren“ eine semantische Überschneidung auf. Ein weiteres Beispiel liefern die Funktionen „Fakturastorno“ und „Rechnungstorno“. Hier werden die synonymen Begriffe „Rechnung“ und „Faktura“ verwendet.

Für die einzelnen Bezeichnungen muss die Frage beantwortet werden, ob die mit semantischen Überschneidungen behafteten Begriffe tatsächlich das gleiche reale Objekt adressieren oder ob es sich hier um eine bewusst vorgenommene Unterscheidung handelt. Vor allem wenn das Modellieren verteilt erfolgt, werden solche Entscheidungen von den einzelnen Modellierern individuell getroffen, was trotz existierender Konventionen zu Inkonsistenzen führen kann.

Metainformationen

In Bezeichnungen, die in diesen Problembereich fallen, werden Informationen gepflegt, die keinen Bezug zum faktischen Inhalt des Modellelements haben, sondern Angaben über das Element selbst und dessen Stellung im Gesamtmodell liefern. Solche Informationen werden in diesem Kontext als Metainformationen bezeichnet. Als erstes Beispiel kann das Ereignis „Bescheid erstellt PROZESSENDE“ genannt werden. In diesem Fall wird eine zusätzliche Angabe gemacht, dass es sich um ein Endereignis des Prozesses handelt. Diese Angabe hat jedoch mit der tatsächlichen Tätigkeit der Erstellung eines Bescheids nichts zu tun. Solche nicht-inhaltliche Angaben sind in Bezeichnungen grundsätzlich unerwünscht und verursachen Probleme bei der Modellanalyse, da sie dort zwangsweise als inhaltliche Information behandelt werden. Darüber hinaus können bei einem Modellvergleich zwei Elemente, die die gleiche inhaltliche Bezeichnung tragen, sich jedoch durch den Zusatz „PROZESSENDE“ unterscheiden, nicht automatisch als inhaltlich äquivalent gekennzeichnet werden.

Weitere Beispiele liefern die Funktionen „Buchung von Transportmittel u. Unterkunft (ausgelöst durch Bestätigung, s.o.“ und „Katalogisierung durchführen (Szenario1: bestehende Orga-Struktur)“. Im ersten Fall werden Metainformationen angegeben, die unnötigerweise das auslösende Ereignis (Bestätigung) nennen und mit dem Zusatz „s.o.“ auf dessen Position im dargestellten Prozessablauf hinweisen. Diese Metainformationen sind hier überflüssig, weil sie sich zwangsweise aus dem formalen, durch den Prozessfluss abgebildeten Teil des Modells bereits ergeben sollten. Im zweiten Fall beinhaltet die Elementbezeichnung Metainformationen, die vermutlich auf ein im Rahmen des Modellierungsprojektes diskutiertes Szenario verweisen. In beiden Beispielen betreffen die besprochenen problematischen Zusätze nicht den faktischen Inhalt des Modellelements.

In einigen Fällen werden solche Metainformationen in einer stark komprimierten Form durch abgekürzte Zusätze dargestellt. Als Beispiele können hier die Funktion „XY_AA-Mittelbedarf Infrastruktur ermitteln_ Mittelverteilung“ oder das Ereignis „Planung_XY ist durchgeführt“ genannt werden (Anm.: Die Kürzel XY und AA bezeichnen durch die Autoren anonymisierte vertrauliche Informationen). Im ersten Fall werden durch vorangestellte Zusätze Hinweise auf relevante Softwaremodule gegeben. Der nachgestellte Zusatz verweist noch auf die hier wahrgenommene betriebswirtschaftliche Aufgabe oder den übergeordneten Prozess. Im Fall des Ereignisses wird durch einen nachgestellten Zusatz der Kontext der Planung weiter konkretisiert. In den beiden Beispielen ergeben sich die Metainformationen bereits aus der Einbettung der Elemente in den Modellkontext und müssen nicht zusätzlich im inhaltlichen Teil der Bezeichnung spezifiziert werden.

Elementtypverletzung

Obwohl die EPK eine im Vergleich zu anderen Prozessmodellierungssprachen niedrige Anzahl von Elementtypen besitzt, weist die hier untersuchte Datenbasis einige Typverletzungen auf. So kann z. B. das Ereignis „siehe HP GesVers“ genannt werden, dessen Bezeichnung sicherlich keinen Prozesszustand beschreibt. Stattdessen wird hier vermutlich auf einen Hauptprozess verwiesen. Des Weiteren zeigt das Beispiel der Funktion „Internationale Meldung ist zu erstellen“, dass einige Funktionen offensichtlich Bezeichnungen tragen, die einen eindeutigen Ereignischarakter aufweisen.

Elementtypverletzungen bereiten Probleme beim Modellvergleich, da zwei Modellelemente, die eigentlich den gleichen faktischen Inhalt adressieren, fälschlicherweise einen unterschiedlichen Typ besitzen und infolgedessen nicht als äquivalent erkannt werden können.

Phrasenstrukturinkonsistenz

Trotz der Etablierung von Modellierungskonventionen in dem untersuchten Modellierungsprojekt ist es in der Datenbasis ersichtlich, dass für analoge Inhalte unterschiedliche Phrasenstrukturen verwendet wurden. Beispielsweise weisen die zwei Funktionen „Prüfen, ob Beschreibung notwendig ist“ und „Prüfung auf vorhandenen Zugang“ unterschiedliche Strukturen auf, obwohl sie analoge Inhalte der Prüfung einer Tatsache adressieren. Hingegen wäre es möglich, die verwendeten Phrasenstrukturen zu vereinheitlichen, ohne sie in ihrer inhaltlichen Ausdrucksstärke zu beschneiden. So kann beispielsweise die Bezeichnung der zweiten Funktion mit „Prüfen, ob Zugang vorhanden ist“ in die Struktur der ersten überführt werden. Ein ähnliches Problem rufen die Funktionen „Übermittlung von XYZ-Informationen an Servicepartner“ und „Individuellen Entwicklungsplan an Mitarbeiter übermitteln“ hervor. Im ersten Fall wird die Tätigkeit der Übermittlung durch ein Substantiv, im zweiten Fall durch ein Verb im Infinitiv dargestellt. Auch hier wäre eine konsequente Verwendung einer Struktur möglich.

Auch bei den Ereignissen ist eine inkonsistente Benutzung analoger Phrasenstrukturen zu beobachten. Die für resultierende Ereignisse typische Struktur <Substantiv, Nominativ> <konjugiertes „sein“> <Verb, Partizip Perfekt> (z. B. „Rechnung ist geprüft“) ist in der Datenbasis 2230-mal vertreten. Gleichzeitig ist jedoch die vollständig äquivalente Struktur <Substantiv, Nominativ> <Verb, Partizip Perfekt> mit 725 Instanzen ebenfalls vertreten. Im gleichen Verhältnis befinden sich die Strukturen <Substantiv, Nominativ> <konjugiertes „sein“> <Adjektiv> (390 Bezeichnungen) und <Substantiv, Nominativ> <Adjektiv> (133 Bezeichnungen). Solche inkonsistente Phrasenstrukturen erschweren den Modellvergleich und die Modellanalyse maßgeblich, da aufwändige Homogenisierungsoperationen durchgeführt werden müssen.

Aggregierte Bezeichnungen

In einigen Fällen beschreiben Modellelementbezeichnungen nicht einen konkreten, abgegrenzten Inhalt, sondern aggregieren gleichzeitig mehrere Inhalte. In dem Fall von Funktionen werden mehrere Tätigkeiten in einem Element dargestellt. Als Beispiel kann die Bezeichnung „Bemerkungen einarbeiten und Schlussfassung an Abt X weiterleiten“

genannt werden. Hier werden zwei Aktivitäten in einer Funktion zusammengefasst. Um den atomaren Charakter der Funktionen zu bewahren, wäre es zweckmäßig, dieses Element aufzuspalten und durch zwei mit einem Kontrollfluss verbundene Funktionen darzustellen.

Das Problem der aggregierten Bezeichnungen ist auch bei den Ereignissen zu beobachten, wie das Beispiel „Ausl. XYZ zuständig: Einzelner Antrag liegt vor, vorl. VersNr nicht notwendig“ zeigt. Hier werden sogar drei Zustände (die Zuständigkeit, das Vorliegen eines Antrags und keine Notwendigkeit einer Nummer) in einem Modellelement aggregiert. Dabei bietet der EPK-eigene UND-Operator die Möglichkeit, den gleichen Inhalt durch drei Ereignisse darzustellen. Ein weiteres Beispiel „Arbeitsplatz/Kapazität ist angelegt/geändert“ demonstriert wiederum das Problem mehrerer aggregierter Zustände, die jedoch nicht durch eine Konjunktion verbunden sind, sondern eher Alternativen darstellen. Solche aggregierte Elementbezeichnungen verursachen Schwierigkeiten beim Modellvergleich, da ein Abstraktionskonflikt entsteht. Dieser kommt zustande, wenn in zwei Modellen der gleiche faktische Inhalt mit einer unterschiedlichen Anzahl von Modellelementen repräsentiert wird [Pf08].

Eingabefehler

In der Datenbasis sind außerdem einige fehlerhafte Bezeichnungen zu beobachten, die vermutlich infolge von Fehlern bei der Eingabe entstanden sind. Als Beispiele können hier die Ereignisse „Personen, Positionen oder Organisationseinheiten zu einem Projektteam für e“, „Equipment Ein-, Ausund Umbau“ sowie die bereits erwähnte Funktion „Arbeitsplatz/Kapazität ist angelegt/geändert“ genannt werden. Im ersten Fall ist die Bezeichnung offensichtlich nicht vollständig. Beim zweiten Ereignis fehlt ein Leerzeichen, wodurch ein in der deutschen Sprache nicht existierendes Wort „Ausund“ zustande kommt. Bei der Funktion wurde das Verb „ist“ falsch geschrieben, infolgedessen wiederum ein unbekanntes Wort „iat“ entstanden ist. Obwohl solche Eingabefehler oft trivial sind, stellen sie trotzdem ein ernstes Problem im Rahmen des Modellvergleichs dar.

2.5 Interpretation der Analyseergebnisse

Die Analyse zeigt, dass die Modelle trotz der zur Verfügung gestellten Modellierungskonventionen eine ganze Reihe von Benennungsschwächen enthalten. Die Konventionen wurden auf unterschiedliche Art und Weise und zudem sehr häufig verletzt. Der Grund hierfür ist sicherlich nicht in böswilliger Absicht zu suchen, sondern vielmehr in dem Unvermögen, die Konventionen umzusetzen. Letzteres ist wohl auch nicht in mangelnder Qualifikation der Modellierer begründet, sondern im schlichten Umfang der Modellierungskonventionen. Unternehmensglossare umfassen meist – so auch im vorliegenden Fall – mehrere hundert bis über tausend Begriffe. Ohne jegliche methodische Unterstützung ist die Auffindung eines geeigneten Begriffs in einer solchen Fülle für den Modellierer erstens äußerst aufwändig, zweitens ermüdend und drittens auch nicht der Qualifikation eines Modellierers entsprechend fordernd. Es wird vermutet, dass Modellierer nach einigen solchen Suchvorgängen schlichtweg resignieren und auf die Verwendung der Konventionen verzichten. Umgekehrt verhält sich das Problem bei Phrasenstrukturkonventionen. Diese beschränken sich im analysierten Fall auf grundlegende Phrasen-

strukturen wie z. B. in Funktionen <Substantiv, Akkusativ> <Verb, Infinitiv>. In den meisten Fällen kann diese Phrasenstruktur nicht eingehalten werden, wie z. B. bereits im Fall der relativ wenig komplexen Bezeichnung „Fehlerhafte Rechnungen aussortieren“. Freilich ist die Vorgabe sämtlicher erlaubter Phrasenstrukturen schwierig, da nicht die Strukturen sämtlicher wahrscheinlich genutzter Bezeichner vorhergesehen werden können. Die Abwesenheit weiterer Phrasenstrukturen führt dann allerdings auch zu einem willkürlichen Gebrauch (vgl. Problembereich *Phrasenstrukturinkonsistenz*).

Die erfolgreiche Durchsetzung von Namenskonventionen erfordert offenbar keine zusätzliche Motivation der Modellierer, sondern deren Entlastung. Für die Realisierung einer solchen Entlastung wird vorgeschlagen, die Aktivitäten, die zur Einhaltung der Konventionen notwendig sind, so weit wie möglich zu automatisieren. Der dazu vorgeschlagene Ansatz unterscheidet sich grundlegend von existenten Ansätzen, die im Folgenden kurz vorgestellt werden.

3 Lösungsbeiträge existenter Arbeiten

In der Vergangenheit ist eine ganze Reihe von Ansätzen entwickelt worden, die das Problem der mangelnden Vergleichbarkeit von Bezeichnungen in konzeptionellen Modellen adressieren. Diese lassen sich durch zwei Dimensionen klassifizieren. Die erste Dimension unterteilt die Ansätze in Ex-ante- und Ex-post-Ansätze: Einerseits werden Verfahren vorgeschlagen mit dem Ziel, das Auftreten konfliktärer Bezeichner von vornherein zu verhindern. Andererseits werden existierende Modelle untersucht, konfliktäre Bezeichner identifiziert und diese durch entsprechende Verfahren aufgelöst. Die zweite Dimension klassifiziert die Ansätze nach der Struktur der betrachteten Bezeichner. Zum Einen werden ausschließlich einzelne Wörter als Bezeichner zugelassen, zum Anderen werden Satzstrukturen, d. h. Phrasen, analysiert.

Einzelwortbezogene Ex-ante-Ansätze

Ansätze, deren Ziel es ist, konfliktäre Bezeichner bereits im Vorfeld zu vermeiden, schlagen sogenannte Namenskonventionen vor. Namenskonventionen werden i. A. in Form von Glossaren oder auch Ontologien spezifiziert, die die für ein Modellierungsprojekt oder eine Modellierungsdomäne gültigen Begriffe enthalten. Während der Modellierung werden diese Vorgaben genutzt, um konventionsgerechte Modelle zu erstellen und so konfliktäre Bezeichner zu vermeiden. Entsprechende Ansätze werden bspw. von [Gr04; BDW07] für Prozessmodelle vorgeschlagen.

Phrasenbezogene Ex-ante-Ansätze

Phrasenbezogene Ex-ante-Ansätze wurden insbesondere in den 1990er Jahren entwickelt und fordern neben einer Standardisierung der gültigen Domänenbegriffe auch eine Standardisierung der zur Benennung von einzelnen Modellelementen erlaubten Satzstrukturen [Ro96; Ku00]. Die gültigen Begriffe werden dabei in sogenannten Fachbegriffsmo-

dellen [KR98] abgelegt. Die Standardisierung der Satzstrukturen erfolgt durch textuelle Empfehlung. Ein alternativer Ansatz definiert Geschäftsobjekte sowie Verrichtungen auf diesen Geschäftsobjekten und führt diese als Anweisungen in Prozessmodellen zusammen [NZ98].

Einzelwortbezogene Ex-post-Ansätze

Frühe Ansätze der 1980er und 1990er Jahre adressieren speziell das Problem der Datenbankintegration und setzen in einem ersten Schritt bei der Integration der Datenbankschemata an [BL84; BLN86; BKK91; LB01; RB01]. Diese Ansätze fokussieren Datenmodellierungssprachen, häufig insbesondere Dialekte des Entity-Relationship-Modells (ERM [Ch76]). Durch den Vergleich von Bezeichnungen der Schemaelemente werden Ähnlichkeiten identifiziert, wobei betont wird, dass ein solcher Vergleich ausschließlich manuell unter Einbeziehung der Schemakonstruktoren erfolgen kann.

Phrasenbezogene Ex-post-Ansätze

Phrasenbezogene Ex-post-Ansätze untersuchen nicht einzelne Wörter als Bezeichner von Modellelementen, sondern sogenannte Konzepte. Konzepte umfassen mehrere Domänenbegriffe, die üblicherweise in Ontologien [Gr93; Gu98] abgelegt sind und durch semantische Beziehungen verbunden sind. So kann bspw. der Begriff „Rechnung“ mit dem Begriff „prüfen“ in Beziehung gesetzt werden, so dass bereits in der Ontologie ausgedrückt ist, dass Rechnungen geprüft werden können. Über diese Konzepte werden die Ähnlichkeitsbeziehungen zwischen Modellen hergestellt [Hö07; EKO07; Sa07].

Abgrenzung des vorliegenden Ansatzes

Grundsätzlich lassen sich alle vorgestellten Ansätze zur Herstellung der Vergleichbarkeit konzeptioneller Modelle anwenden. Speziell für EPKs stoßen jedoch die einzelwortbezogenen Ansätze an ihre Grenzen, da konfliktäre Bezeichner in Form von Satzstrukturen auf diese Weise kaum erkannt werden können. Die hier vorgestellten einzelwortbezogenen Ex-ante-Ansätze wurden zwar für Prozessmodelle entwickelt, betrachten jedoch innerhalb der Satzstrukturen der Bezeichner ausschließlich einzelne Begriffe.

Phrasenbezogene Ansätze scheinen hier geeigneter, da sie die Bezeichnungspraktiken von EPKs explizit berücksichtigen können. Ex-post-Ansätze sind in der Lage, aufgetretene Namenskonflikte durch Gegenüberstellung der Modelle und Analyse ihrer Bezeichnungsstrukturen aufzulösen. Den Autoren erscheint die vorherige Vermeidung von Namenskonflikten allerdings zielführender, da dies eine aufwändige Analyse der fertiggestellten Modelle obsolet macht (freilich ist eine vorherige Vermeidung von Namenskonflikten nur dann möglich, wenn nicht bereits Modelle existieren). Die aufgeführten phrasenbezogenen Ex-ante-Ansätze bieten daher aus Sicht der Autoren eine vielversprechende Grundlage für die Formulierung eines Ansatzes zur Vermeidung von Namenskonflikten in EPKs. Die Ansätze von [Ro96; Ku00] erlauben die Formulierung von Phrasenstrukturen einerseits und Begriffskonventionen andererseits. Es fehlt allerdings

eine Formalisierung, die für eine automatisierte Durchsetzung von Namenskonventionen notwendig ist. Letztere ist für den Erfolg von Namenskonventionen kritisch, wie die Analyse in Abschnitt 2 zeigt. Der Ansatz von [NZ98] ist formalisiert, betrachtet jedoch die Ausgestaltung der Bezeichnerphrasen sehr eingeschränkt auf wenige vorgegebene Fälle.

Der vorliegende Ansatz ist durch

- die explizite Berücksichtigung von Satzfragmenten als Modellelementbezeichner,
- seine Formalisierung und
- die hierdurch automatisierbare Sicherstellung der konventionstreuen Modellierung

von existenten Ansätzen abzugrenzen.

4 Ein Ansatz zur Vermeidung von Namenskonflikten in EPKs

4.1 Konzeption des Ansatzes

Für die Konstruktion eines Ansatzes zur Vermeidung von Namenskonflikten in Prozessmodellen wird die Idee der Namenskonventionen, wie sie in den 1990er Jahren von ROSEMANN vorgeschlagen wurde, wieder aufgegriffen. Die Analyse der EPK-Datenbasis zeigt, dass die alleinige Existenz solcher Namenskonventionen keinesfalls dazu führt, dass Namenskonflikte ausbleiben. Namenskonventionen, die lediglich als Empfehlungen, d. h. in Textform, vorliegen, werden nicht notwendigerweise umgesetzt.

Es wird deshalb vorgeschlagen, Namenskonventionen formal zu spezifizieren und deren Umsetzung während des Modellierungsprozesses automatisiert sicherzustellen. Hierfür sind folgende methodische Komponenten notwendig:

- Ein Domänenlexikon, welches die in der Modellierungsdomäne erlaubten Begriffe enthält. Das Lexikon darf sich nicht auf Substantive beschränken, da auch Verrichtungen und Eigenschaften Teil einer Domäne sein können. Das Lexikon ist daher für Substantive, Verben und Adjektive/Adverbien zu konzipieren. Sämtliche übrigen Wortformen (z. B. Artikel, Konjunktionen, Präpositionen etc.) enthalten keine Domänensemantik, weshalb ihre Spezifikation im Domänenlexikon nicht notwendig ist.
- Ein Repository, das die erlaubten Satzstrukturen definiert (z. B. <Substantiv, Akkusativ, Singular> <Verb, Partizip Perfekt> für ergebnisanzeigende Ereignisse).
- Ein Verfahren, das bei der Modellierung automatisiert die Konformität der eingegebenen Bezeichner mit den Namenskonventionen sicherstellt.

Durch „Einsetzen“ der im Domänenlexikon spezifizierten Begriffe in die im Repository spezifizierten Satzstrukturen ergeben sich mögliche den Namenskonventionen entsprechende Bezeichnungen für Modellelemente. Ein derartiges „Einsetzen“ von Begriffen in Satzschablonen wird durch ein entsprechendes Verfahren bei der Modellierung sicherge-

stellt. Je angelegtem Modellelement werden die validen Satzstrukturen ausgewählt und mit validen Begriffen „gefüllt“ (vgl. Abbildung 2; vgl. zu einer detaillierten formalen Spezifikation des Ansatzes [Be09; DHL09]).

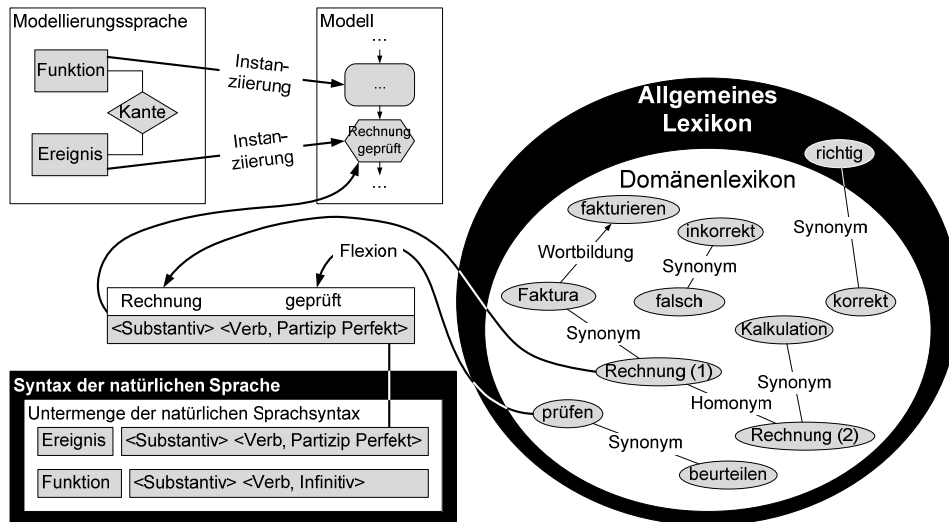


Abbildung 2: Nutzung formalisierter Namenskonventionen

Aus Effizienzgründen sollte es allerdings möglich sein, eine Modellelementbezeichnung wie gewohnt als Freitext einzugeben. Falls eine solche Eingabe dann nicht den Konventionen entspricht, sind dem Modellierer automatisch konventionskonforme Alternativbezeichner anzubieten (z. B. „Rechnung prüfen“ anstatt der evtl. nicht konformen Bezeichnung „Faktura muss validiert werden“). Ob eine eingegebene Bezeichnung den Konventionen entspricht, wird durch ein Verfahren ermittelt, das den Bezeichner sowohl auf verwendete Begriffe als auch auf die verwendete Satzstruktur analysiert. Hierfür werden linguistische Parsingmethoden verwendet, wie sie bspw. von Müller [Mu96; Mu99] vorgeschlagen wurden.

Konkret wird jede Eingabe des Modellierers durch einen entsprechenden Parser in ihre Bestandteile, d. h. ihre Satzstruktur und die Grundformen der verwendeten Wörter (sog. „Lexeme“) zerlegt (vgl. Abbildung 3; (1)). Die Lexeme des Typs *Substantiv*, *Adjektiv/Adverb* und *Verb* werden auf Konformität gegenüber dem Domänenlexikon geprüft (2). Entsprechen ein oder mehrere Lexeme nicht den Konventionen, wird mittels eines allgemeinen Lexikons geprüft, ob die Lexeme durch synonyme, valide Gegenstücke ersetzt werden können (3). Die validen Begriffe werden in mögliche Phrasenstrukturen eingesetzt und dem Modellierer als valide Alternative vorgeschlagen (4). So werden bspw. aus der nicht konventionskonformen Phrase „Falsche Faktura werden ermittelt“ die korrekten Phrasen „Inkorrekte Rechnungen ermitteln“ oder „Rechnungen ermitteln“. Aus diesen kann der Modellierer wählen oder eine neue Bezeichnung eingeben (vgl. zu einer detaillierten formalen Spezifikation dieses Verfahrens nochmals [Be09; DHL09]. Zur algorithmischen Spezifikation des Vorschlagswesens vgl. ausführlich [DHL09]).

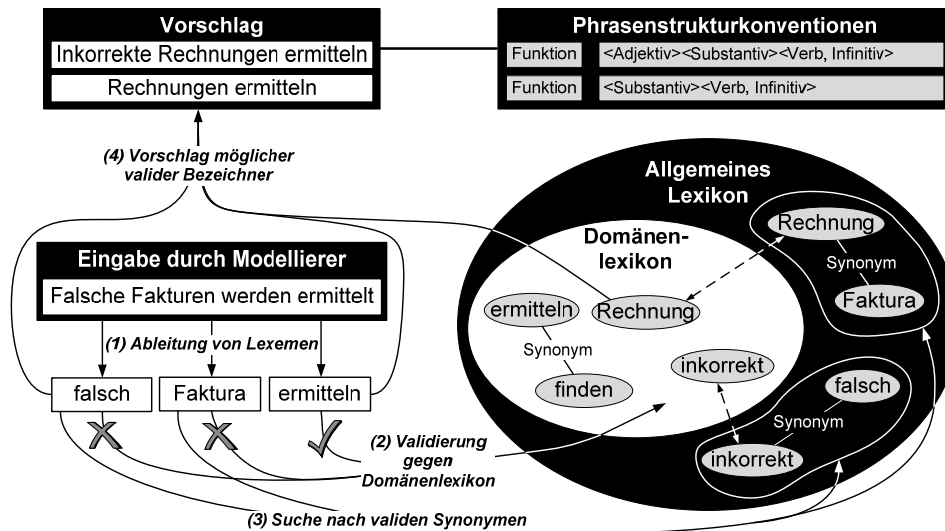


Abbildung 3: Automatisches Vorschlagen validen Bezeichner

4.2 Lösung identifizierter Probleme

Die im Abschnitt 2.4 ermittelten Probleme lassen sich mithilfe des vorgestellten Ansatzes wie folgt vermeiden:

- **Synonyme:** Im Domänenbegriffsmodell sind die zu verwendenden Begriffe eindeutig spezifiziert. Bei Verwendung eines Synonyms wird dieses den Konventionen entsprechend durch den validen Begriff ausgetauscht. Synonymprobleme sind hiermit ex-ante ausgeschlossen.
- **Metainformationen:** Die Namenskonflikte, die sich durch Metainformationen ergeben werden in zweierlei Hinsicht verhindert. Zum Einen befinden sich Begriffe, die nicht Teil der Domäne sind, nicht im Domänenlexikon und können deswegen auch nicht verwendet werden. Die Verwendung von Begriffen wie z. B. „Prozessende“ oder „Planung_XY“ wird dadurch vermieden. Zum Anderen erlauben sinnvolle Phrasenstrukturen für EPKs zumeist kein Voranstellen oder Nachstellen entsprechender Metainformation. Hierfür wäre bspw. die direkte Folge zweier Substantive erforderlich.
- **Elementtypverletzung:** Phrasen, die keinen Ereignischarakter haben, aber als Bezeichner für Ereignisse verwendet werden, werden durch speziell für Ereignisse erstellte Phrasenstrukturkonventionen vermieden. Gleiches gilt für Funktionen.
- **Phrasenstrukturinkonsistenz:** Zur Vermeidung von Phrasenstrukturinkonsistenzen ist die Definition von semantisch äquivalenten Phrasenstrukturen zu vermeiden. Anstatt der beiden äquivalenten Strukturen <Substantiv> <Verb, Partizip Perfekt> und

<Substantiv> <konjugiertes “sein“> <Verb, Partizip Perfekt> ist eine dieser beiden zu wählen und als Standard festzulegen.

- **Aggregierte Bezeichnungen:** Aggregierte Bezeichnungen konnektieren mehrere Phrasen. Solche Multiphrasen sind bei der Spezifikation der Konventionen zu vermeiden. Stattdessen wird der Modellierer dazu angehalten, entsprechend mehrere Elemente anzulegen und diese mit den Operatoren zu verknüpfen.
- **Eingabefehler:** Eingabefehler, die ganze Phrasen betreffen (bspw. abbrechende Phrasen) werden durch die Phrasenstrukturkonventionen abgefangen. Einzelne Begriffe betreffende Eingabefehler können größtenteils durch einen Abgleich mit dem Domänenlexikon und im Rahmen der Synonymsuche mit dem allgemeinen Lexikon erkannt werden.

4.3 Werkzeugunterstützung

Eine konkrete technische Umsetzung des hier vorgestellten Ansatzes liegt als Modellierungswerkzeug vor (vgl. Abbildung 4).

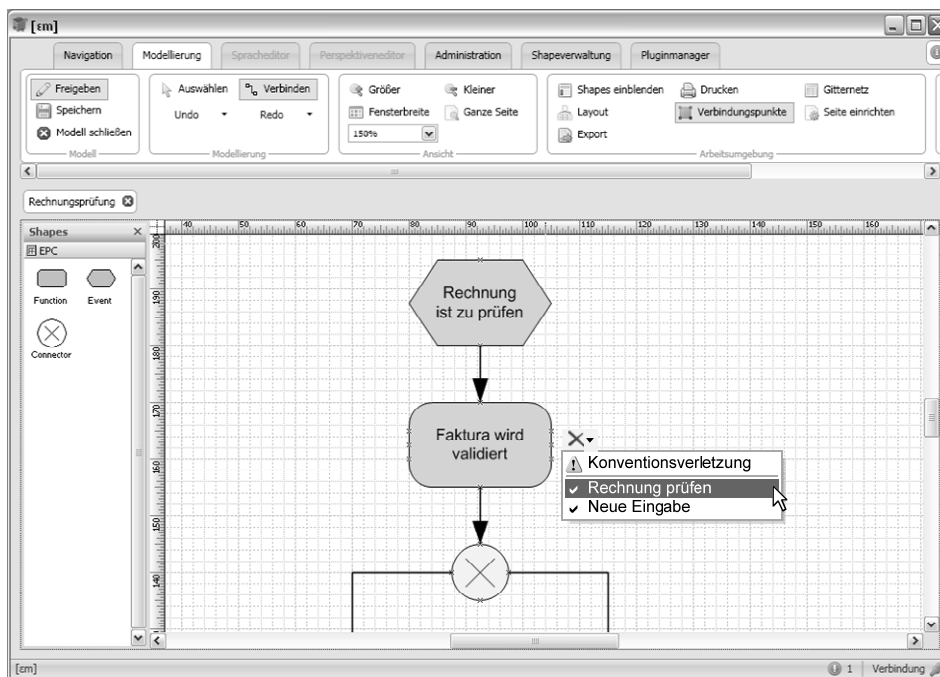


Abbildung 4: Umsetzung des Ansatzes als Modellierungswerkzeug

Der Modellierer wird durch Popup-Fenster auf etwaige Konventionsverletzungen hingewiesen und erhält konventionstreu Alternativvorschläge, die er akzeptiert, falls sie die von ihm intendierte Semantik wiedergeben. Falls die Vorschläge nicht passen, kann

der Modellierer entscheiden, ob er erneut eine Bezeichnung eingibt oder eine Anfrage an den Administrator des Modellierungswerkzeugs (idealerweise ein Modellierungsexperte; vgl. hierzu [DHL09]) stellt, neue Bezeichner bzw. Phrasentypen mit in die Konventionen aufzunehmen.

5 Fazit und Ausblick

Die linguistische Analyse der EPKs hat gezeigt, dass informal spezifizierte Namenskonventionen für die Sicherstellung einer einheitlichen Benennung von Modellelementen nicht ausreichen. Des Weiteren reicht es nicht aus, die Standardisierung von Bezeichnungen auf einzelne Begriffe zu beschränken. Mit dem vorgestellten Ansatz werden Namenskonventionen formalisiert, wobei zwei Aspekte wesentlich sind:

- Mit der Bereitstellung von Bezeichnungskonventionen bereits im Vorfeld der Modellierung wird die Grundlage dafür gelegt, dass Namenskonflikte erst gar nicht auftreten. Da die Semantik der Bezeichnungen bereits im Vorfeld bekannt ist und diese ausschließlich in bekannte Phrasenstrukturen eingeordnet werden, entfällt ein aufwändiger nachträglicher Bezeichnungsabgleich.
- Die automatisierte Anleitung des Modellierers während der Modellerstellung ist von nicht zu unterschätzender Bedeutung, da nur auf diese Weise garantiert werden kann, dass die Bezeichnungskonventionen auch umgesetzt werden.

Die Spezifikation von Begriffs- und Phrasenstrukturkonventionen ist freilich mit einem nicht geringen Aufwand verbunden. Dem entgegen steht die Tatsache, dass diese Spezifikation je Modellierungsprojekt, Unternehmen oder Domäne lediglich einmalig zu erfolgen hat und in der Folge wiederverwendbar ist. Hier liegen auch die Grenzen des Ansatzes, der nur dann Anwendung finden kann, wenn

- die formalisierten Modellierungskonventionen sämtlichen am Modellierungsprozess beteiligten Personen zugänglich gemacht werden können und
- es sich um ein Modellierungsprojekt handelt, das keine bzw. nur eine überschaubare Anzahl bereits existenter Modelle in den Modellierungsprozess mit einbeziehen muss.

Induziert durch letztere Beschränkung könnte dem Ansatz entgegengehalten werden, seine Anwendbarkeit beschränke sich auf einige wenige Ausnahmefälle. Nach Meinung der Autoren käme allerdings eine derart begründete Verwerfung des Ansatzes einer Resignation ob des Vergleichbarkeitsproblems von Modellen gleich. Bestehen solche Probleme, sind sie im Nachgang der Modellierung nur mit äußerst hohem Aufwand wieder zu beheben. Eine breite Durchsetzung des Ansatzes in allen Bereichen der Modellierung ist aber sicherlich erst längerfristig möglich. Zudem existieren Situationen, in denen bereits im Unternehmen vorhandene IST-Modelle, die mit anderen bzw. neuen Modellen abzugleichen sind, von neu zu konstruierenden SOLL-Modellen abgelöst werden. Dies ist etwa im Rahmen von Reorganisations- oder auch von sog. „Mergers & Akquisi-

tions“-Projekten der Fall. Auch hier liegt ein vielversprechendes Anwendungsgebiet des vorgestellten Ansatzes.

Weitere Forschungsfelder eröffnen sich zukünftig in der konkreten Anwendung des vorgestellten Ansatzes. Hier ist zu untersuchen, wie die Anwendung die Geschwindigkeit der Modellerstellung beeinflusst und bis zu welchem Grad die Vergleichbarkeit von Modellen gesteigert werden kann. Weiterhin ist zu überprüfen, ob und wie der Ansatz durch Modellierer akzeptiert wird, die durch die automatisierte Durchsetzung von Modellierungskonventionen in ihren Modellierungsfreiheiten beschränkt werden. Erste Erfahrungen mit der werkzeuggestützten Anwendung des Ansatzes lassen allerdings vermuten, dass eine automatisierte Anleitung zur Erfüllung von Modellierungskonventionen von den betroffenen Personen eher begrüßt als abgelehnt wird. Auch die Antwortzeiten des implementierten Verfahrens zur Durchsetzung der Modellierungskonventionen bewegen sich bisher in einem durch die Modellierer nicht wahrnehmbaren Rahmen.

Literaturverzeichnis

- [BDW07] Born, M.; Dörr, F.; Weber, I.: User-friendly semantic annotation in business process modeling. In: Weske, M.; Hacid, M.-S.; Godart, C. (Hrsg.): Proceedings of the International Workshop on Human-Friendly Service Description, Discovery and Matchmaking (Hf-SDDM 2007) at the 8th International Conference on Web Information Systems Engineering (WISE 2007). Nancy 2007, S. 260-271.
- [Be09] Becker, J.; Delfmann, P.; Herwig, S.; Lis, L.; Stein, A.: Formalizing Linguistic Conventions for Conceptual Models. In: Proceedings of the 28th International Conference on Conceptual Modeling (ER 2009). LNCS 5829. Gramado, Brasilien 2009, S. 70-83.
- [BKK91] Bhargava, H. K., Kimbrough, S. O., Krishnan, R.: Unique Name Violations, a Problem for Model Integration or You Say Tomato, I Say Tomahto. ORSA Journal on Computing 3 (1991) 2, S. 107-120.
- [BL84] Batini, C., Lenzerini, M.: A Methodology for Data Schema Integration in the Entity Relationship Model. IEEE Transactions on Software Engineering 10 (1984) 6, S. 650-663.
- [BLN86] Batini, C., Lenzerini, M., Navathe, S. B.: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18 (1986) 4, S. 323–364.
- [Ch76] Chen, P. P.-S.: The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems 1 (1976) 1, S. S. 9-36.
- [DHL09] Delfmann, P.; Herwig, S.; Lis, L.: Unified Enterprise Knowledge Representation with Conceptual Models – Capturing Corporate Language in Naming Conventions. In: Proceedings of the 30th International Conference on Information Systems (ICIS 2009). Phoenix, Arizona, USA, 2009.
- [EKO07] Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM) 2007. Ballarat 2007.
- [Gr04] Greco, G.; Guzzo, A.; Pontieri, L.; Saccà, D.: An ontology-driven process modeling framework. In: Galindo, F.; Takizawa, M.; Traunmüller, R. (Hrsg.): Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA 2004). Zaragoza 2004, S. 13-23.
- [Gr93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5 (1993) 2, S. 199-220.

- [Gu98] Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (Hrsg.): Proceedings of the 1st International Conference on Formal Ontologies in Information Systems. Trento 1998, S. 3-15.
- [Hö07] Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies. In: Österle, H.; Schelp, J.; Winter, R. (Hrsg.): Proceedings of the 15th European Conference on Information Systems (ECIS 2007). St. Gallen 2007, S. 1620-1631.
- [HS06] Hadar, I., Soffer, P.: Variations in conceptual modeling: classification and ontological analysis. *Journal of the AIS* 7 (2006) 8, S. 568-592.
- [KNS92] Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. In: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89. Saarbrücken 1992.
- [KR98] Kugeler M., Rosemann, M.: Fachbegriffsmodellierung für betriebliche Informationssysteme und zur Unterstützung der Unternehmenskommunikation. In: Informationssystem Architekturen. Fachausschuss 5.2 der Gesellschaft für Informatik e. V. (GI), 5 (1998) 2, S. 8-15.
- [Ku00] Kugeler, M.: Informationsmodellbasierte Organisationsgestaltung. Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation, Berlin 2000.
- [LB01] Lawrence, R., Barker, K.: Integrating Relational Database Schemas using a Standardized Dictionary. In: Proceedings of the 2001 ACM symposium on Applied computing (SAC). Las Vegas 2001.
- [Mu96] Müller, S.: The Babel-System – An HPSG-Fragment for German, a Parser and a Dialog Component. In: Proceedings of the 4th International Conference on the Practical Application of Prolog. London 1996.
- [Mu99] Müller, S.: Deutsche Syntax deklarativ – Head-Driven Phrase Structure Grammar für das Deutsche. Tübingen 1999.
- [NZ98] Nüttgens, M.; Zimmermann, V.: Geschäftsprozeßmodellierung mit der objektorientierten Ereignisgesteuerten Prozeßkette (oEPK). In: Maicher, M.; Scheruhn, H.-J. (Hrsg.): Informationsmodellierung – Branchen, Software- und Vorgehensreferenzmodelle und Werkzeuge. Wiesbaden 1998, S. 23-36.
- [Pfo8] Pfeiffer, D.: Semantic Business Process Analysis – Building Block-based Construction of Automatically Analyzable Business Process Models. Münster 2008.
- [RB01] Rahm, E.; Bernstein, P. A.: A Survey of Approaches to Automatic Schema Matching. *The International Journal on Very Large Data Bases* 10 (2001) 4, S. 334-350.
- [Ro96] Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden 1996.
- [Sa07] Sabetzadeh, M.; Nejati, S.; Easterbrook, S.; Chechik, M.: A Relationship-Driven Framework for Model Merging. Workshop on Modeling in Software Engineering (MISE'07) at the 29th International Conference on Software Engineering, Minneapolis 2007.
- [Sc00] Scheer, A.-W.: ARIS – Business Process Modelling. 3. Auflage, Berlin 2000.
- [Sc94] Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Proceedings of the First International Conference on New Methods in Natural Language Processing. Manchester 1994, S. 44-49.

EPK-Varianten auf dem Prüfstand: Explorative Studie zur Gebrauchstauglichkeit von eEPK und oEPK

Frank Hogrebe¹; Alexander Jürgens²; Sven Pagel²; Markus Nüttgens³

¹Landeshauptstadt Düsseldorf
Organisations-, Personal-, IT- und Wirtschaftsförderungsdezernat
Burgplatz 1, D-40213 Düsseldorf
frank.hogrebe@stadt.duesseldorf.de

²Fachhochschule Düsseldorf
Betriebswirtschaftslehre, insb. Kommunikation und Multimedia
Forschungsschwerpunkt Kommunikationsforschung
Universitätsstraße, Gebäude 23.32, D-40225 Düsseldorf
alexander.juergens|sven.pagel@fh-duesseldorf.de

³Universität Hamburg
Fakultät Wirtschafts- und Sozialwissenschaften
Lehrstuhl für Wirtschaftsinformatik
Von-Melle-Park 5, D-20146 Hamburg
markus.nuettgens@wiso.uni-hamburg.de

Zusammenfassung: Der Schwerpunkt bisheriger Arbeiten zur Gebrauchstauglichkeit semiformaler Modellierungssprachen in der Wirtschaftsinformatik liegt im Anwendungsbereich der Softwareentwicklung, mit der Folge, dass maschinenbezogene Anforderungen regelmäßig im Vordergrund stehen. Gleichwohl dienen semiformale Modellierungssprachen auch zur Erfassung und Beschreibung von Informationsmodellen als Instrumente einer „Mensch-zu-Mensch“-Kommunikation. Der vorliegende Beitrag stellt die Ergebnisse einer explorativen Studie zur Gebrauchstauglichkeit (engl. Usability) von EPK-Varianten vor, die sowohl Ersteller als auch Nutzer von Informationsmodellen einbezieht. Als Grundlage zur Messung und Bewertung der Usability der EPK-Varianten wird ein Untersuchungsrahmen zugrunde gelegt, der Konzepte der Wirtschaftsinformatik mit Konzepten der Kommunikationsforschung kombiniert. Dabei werden sowohl einschlägige Teile der Usability-Norm EN ISO 9241 einbezogen als auch die Methode des Eyetrackings (Blickbewegungsregistrierung). Die Ergebnisse der Arbeit adressieren sowohl die wissenschaftliche Forschung als auch die Praxis.

1 Motivation und Untersuchungsgegenstand

Zur Beurteilung von Informationssystemen haben sich zwei wesentliche Forschungsströme herausgebildet [WiTo05, S. 85]. Der Erste stellt die „Benutzerzufriedenheit (user satisfaction)“ bei der Erstellung und Nutzung von Informationssystemen in den Vordergrund [Se97], [Me90], [IOB83], [BaPe83]. Die zweite Forschungsrichtung legt den Schwerpunkt auf die „Technikakzeptanz (technology acceptance)“ bei der Erstellung und Nutzung von Informationssystemen [VMD03], [Sz96], [HaBa94], [Da89]. Semiformale Modellierungssprachen sind der Ausgangspunkt zur Visualisierung betrieblicher Strukturen und Abläufe in Unternehmen und Verwaltungen und bilden damit die Grundlage für die Beschreibung von Informationsmodellen, die grundlegend für die Entwicklung von Informationssystemen sind. Die vorliegende Untersuchung legt den Fokus auf die Benutzerzufriedenheit (user satisfaction) bei der Messung und Bewertung semiformaler Modellierungssprachen und ihrer Modelle (Untersuchungsgegenstand). Die Gebrauchstauglichkeit (engl. Usability) hängt von den Anforderungen ihrer Benutzer ab. Systematische Untersuchungen zur Usability sind unter anderem ein Forschungsfeld der Kommunikationsforschung. Es handelt sich dabei um die empirische Sozialforschung innerhalb der Kommunikationswissenschaft. Der vorliegende Beitrag bewegt sich somit an der Schnittstelle von Wirtschaftsinformatik und Kommunikationswissenschaft. Arbeiten zur Usabilityforschung finden sich bei [CHL06], [RoPu07], [HSS02], [SSD02], [HaSc00]. Einschlägige Normen zur Usability finden sich in den Teilen 11, 12 und 110 der Europäischen Usability-Norm EN ISO 9241 [ISO99], [ISO00], [ISO08].

Usability-Untersuchungen zur Messung und Bewertung der Gebrauchstauglichkeit von Modellierungssprachen und ihren Modellen sind bisher nicht publiziert. Zur Vorbereitung einer solchen Betrachtung wird ein Untersuchungsrahmen entwickelt, der sowohl Erkenntnisse der Wirtschaftsinformatik als auch der Kommunikationsforschung einbezieht. Als Instrumentarium wird neben (klassischen) Befragungstechniken auch die Eyetracking-Methode (Blickbewegungsregistrierung) eingesetzt, um neben qualitativen Indikatoren (wie Zufriedenheit mit einer Modellierungssprache bzw. ihrer Modelle) auch quantitative Indikatoren (wie Einfluss von Blickbewegungen auf die Zeitdauer der Modellierung oder Fixierungsinformationen bei der Modellnutzung) untersuchen zu können.

Der Beitrag ist wie folgt aufgebaut: Im zweiten Abschnitt werden zunächst die verschiedenen Benutzergruppen von Modellierungssprachen und Informationsmodellen näher betrachtet. Der dritte Abschnitt befasst sich mit der Usability-Norm EN ISO 9241 als Ausgangspunkt zur Evaluation der Usability von Modellierungssprachen. In diesem Kontext wird auch die Eyetracking-Methode (Blickbewegungsregistrierung) einbezogen. Im vierten Abschnitt wird ein Untersuchungsrahmen zur Messung und Bewertung der Usability von Modellierungssprachen vorgestellt, der als Grundlage für Usability-Eyetracking-Studien dient. Dieser wird im Abschnitt 4 auf einen Vergleich von zwei EPK-Varianten exemplarisch angewandt. Die Arbeit schließt mit einer Zusammenfassung, Hinweisen zur Limitation der Ergebnisse und einem Ausblick auf weiteren Forschungsbedarf.

2 Benutzer von Modellierungssprachen und Informationsmodellen

Urheber und Adressaten modellierungssprachiger Äußerungen können Menschen oder Maschinen sein. Maschinen sind automatische formale Systeme, die mittels einer gegebenen Menge endlich vieler Operationen eine ebenfalls gegebene Menge von Objekten manipulieren [Pa06, S. 167]. Dabei sind Maschinen als physische Geräte (Rechner) dynamisch äquivalent zu Automaten. Menschen, die sich aufgrund derselben Schriftsprache verständigen, bilden eine soziolinguistische Sprachgemeinschaft. Urheber und Adressaten können in der betrieblichen Praxis zu Personengruppen zusammengefasst werden, bei denen ein ähnliches Interesse an der Art der Modellerstellung /-nutzung besteht [BSG99, S. 6f.]:

	Urheber	Adressaten
Menschen	<ul style="list-style-type: none"> ▪ <i>Konstrukteure</i>, als Entwickler von Modellierungssprachen ▪ <i>Ersteller</i> (Modellierer) von Informationsmodellen (wie Fachanwender, Organisationsgestalter und Unternehmensberater) 	<ul style="list-style-type: none"> ▪ <i>Programmierer</i>, die aus Modellinhalten Programme erzeugen ▪ <i>Nutzer</i> (wie Mitarbeiter in Fachbereichen, Organisationsabteilungen und DV-Abteilungen), die Informationsmodelle als Arbeits- und Entscheidungsgrundlage einsetzen; gleiches gilt für Controller, Unternehmensberater oder Mitarbeiter von Zertifizierungsstellen
Maschinen	<ul style="list-style-type: none"> ▪ <i>Maschinen</i>, bei der automatischen Generierung von Programmen 	<ul style="list-style-type: none"> ▪ <i>Maschinen</i>, die Programme ausführen

Tabelle 1: Urheber und Adressaten von Modellierungssprachen /-modellen

Tabelle 1 zeigt in Anlehnung an [BSG99, S. 7f.], [Pa06, S. 116] und [Du01, S.64] exemplarisch Gruppen von Urhebern und Adressaten von Modellierungssprachen /-modellen. *Mitarbeiter in DV-Abteilungen* (Datenverarbeitungsabteilungen) setzen vor allem Modelle zur Informationssystemgestaltung ein. Der Fokus der DV-Abteilungen liegt in der Betrachtung des gesamten Informationsmodells einer Unternehmung. Dabei stehen die Informationsflüsse in den Prozessen im Vordergrund. Aufgrund der im Regelfall geringeren Methodenkompetenz von *Mitarbeitern in Fachbereichen* sind für diesen Adressatenkreis leicht verständliche grafische Modelle hilfreich. Für die Nutzenden steht dabei die intuitive Nachvollziehbarkeit im Vordergrund. Mitarbeiter von Fachabteilungen erwarten von Informationsmodellen eine detaillierte Beschreibung ihrer Prozesse; dabei stehen weniger die systemrelevanten, als die sachlogischen Zusammenhänge im Vordergrund [BSG99, S. 30]. *Mitarbeiter in Organisationsabteilungen* erstellen und nutzen Informationsmodelle zur Beschreibung von Aufbau- und Ablauforganisationen. Sie besitzen ein vorrangiges Interesse an Organisations-, Funktions- und Prozessmodellen.

3 Normen zur Usability und Eyetracking-Methode

3.1 Usability-Norm EN ISO 9241

Untersuchungen zur Usability von Informationssystemen finden sich bei [SND05], [TM04], [PI04], [GKP03], [Ku03]. Solche Untersuchungen sind unter anderem ein Untersuchungsfeld in Kommunikationsforschung. Arbeiten finden sich in den Bereichen der Web-Usability ([CHL06], [RoPu07], [HSS02], [SSD02], [HaSc00]), Usability mobiler Systeme [KjSt04] und TV-Usability ([OBB07], [Ca06], [PeGr03]). Dabei wird im Bereich von Web-Angeboten die Usability nicht nur als ein kritischer Erfolgsfaktor im e-Business angesehen, sondern bildet ein wesentliches Qualitätskriterium für Web-Angebote kommerzieller wie nicht-kommerzieller Organisationen und Institutionen [Ni99, S. 9]. Das Methodenspektrum zur Evaluation von Usability ist vielschichtig, vgl. [Pe00], [ST02], [HaSc00]. Einschlägige Normen finden sich in den Teilen 11, 12 und 110 der DIN EN ISO 9241:

- Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 11: Anforderungen an die Gebrauchstauglichkeit – Leitsätze; EN ISO 9241-11:1998 [ISO99],
- Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 12: Informationsdarstellung; EN ISO 9241-12:1998 [ISO00],
- Ergonomie der Mensch-System-Interaktion, Teil 110: Grundsätze der Dialoggestaltung; EN ISO 9241-110: 2006 [ISO08].

Die Teile 12 und 110 unterstützen das Konzept der Gebrauchstauglichkeit, das Inhalt von Teil 11 der DIN EN 9241 ist [ISO99]. Gebrauchstauglichkeit wird definiert als das „Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzenkontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ [ISO99, S. 4]. Die ISO 9241-12 definiert charakteristische Eigenschaften dargestellter Informationen und gibt Empfehlungen zur Darstellung als Teil des Dialoges (Informationsdesign). Dargestellte Informationen sind aus einer technikbezogenen Betrachtung zwangsläufig in Dialogen enthalten, die auf den Grundsätzen vom Teil 110 von ISO 9241 basieren. Diese unterstützen wiederum vorrangig die Gestaltung des dynamischen Verhaltens eines interaktiven Systems (Interaktionsdesign). Die charakteristischen Eigenschaften der dargestellten Informationen aus ISO 9241-12 unterstützen insbesondere die Selbstbeschreibungsfähigkeit und Erwartungskonformität des Dialogs [ISO08, S. 21]. Bei der Untersuchung von benutzerbezogenen Anforderungen an die Usability von Modellierungssprachen und deren Modellen sind zwei Aspekte zu unterscheiden:

- Eigenschaften der Informationsdarstellung [ISO00] adressieren direkt die Benutzer von Modellierungssprachen und Informationsmodellen,
- Grundsätze der Dialoggestaltung [ISO08] adressieren hingegen primär technische Aspekte der (software)technischen Realisierung durch Modellierungstools.

Die Anwendung der Grundsätze aus den Teilen ISO 9241-110 und ISO 9241-12 unterstützen insgesamt die Gebrauchstauglichkeit, d. h. Effektivität, Effizienz und Zufriedenstellung aus ISO 9241-11 [ISO98, S. 4]. Im Rahmen der pilotierten Anwendung des Untersuchungsrahmens (Abschnitt 4) werden alle drei Teile der DIN EN ISO 9241 berücksichtigt.

3.2 Eyetracking-Methode

Eyetracking oder Blickbewegungsregistrierung bezeichnet eine Methode, mit der Blickverläufe einer Person beim Betrachten eines Gegenstandes oder einer Anwendung gemessen werden [RDD08], [Du03]. Diese Methode soll auf die Messung und Bewertung der Usability von Modellierungssprachen und deren Modellen angewandt werden. Bei der Kombination von Eyetracking- und Usability-Tests [BPW07], [GoWi03], [JaKa03], [PGJ08] werden die qualitativen Ergebnisse aus den Usability-Untersuchungen durch quantitative Eyetracking-Daten ergänzt. Der Eyetracking-Test wird dabei aufgabengestützt von den Probanden ohne Eingreifen eines Testleiters durchgeführt und im Anschluss durch eine strukturierten Befragung ergänzt (qualitativer Aspekt). Zusätzlich wird bei der Eyetracking-Methode der Blickverlauf des Probanden mit einer entsprechenden Software aufgezeichnet und gemessen (quantitativer Aspekt). Unterschiedliche Fragestellungen können so untersucht werden: Was sehen Benutzer und was nehmen sie wirklich wahr? Welche Text-/Modellteile werden betrachtet, welche werden wirklich gelesen? Welche Reihenfolge wählt ein Benutzer bei der Betrachtung und wie lange benötigt dieser zur Erfassung von Sachverhalten? Welche Text-/Modellteile entgehen der Aufmerksamkeit völlig (sog. dead zones, [St02, S. 188])? Durch die Einbeziehung der Eyetracking-Methode in die Messung und Bewertung der Usability von Modellierungssprachen und deren Modellen bieten sich unterschiedliche Anwendungsaspekte:

- So kann das subjektive Empfinden von Benutzern über die Modellierungsdauer bei der Verwendung unterschiedlicher Modellierungssprachen mit den Blickbewegungs- und zeitlichen Messwerten aus den Eyetracking-Daten verglichen werden,
- Modellierungswege der Ersteller von Modellen können nachverfolgt und ausgewertet werden,
- zudem kann die Usability von eingesetzten Modellierungstools gleichermaßen in einer Untersuchung Berücksichtigung finden.

4 Untersuchungsrahmen zur Usability von Modellierungssprachen

4.1 Kernelemente und Rahmenkonzept

Der Untersuchungsrahmen hat das Ziel, die Messung und Bewertung der Usability von Modellierungssprachen und deren Modellen durch eine Kombination von Usability-Testelementen (wie Befragungstechniken) und der Eyetracking-Methode zu strukturieren. Der Untersuchungsrahmen ist ein Rahmenkonzept zur Planung und Durchführung zweck- und adressatenspezifischer Usability-Eyetracking-Studien [HoNü09]. Das Rahmenkonzept umfasst sechs Kernelemente, die schrittweise das Untersuchungsdesign konkretisieren. Dabei kann die Reihenfolge und Ausprägung der Vorgehensschritte abhängig vom Untersuchungsziel / -kontext und den spezifischen Rahmenbedingungen variieren. Durch die Anwendung des Untersuchungsrahmens wird sukzessive ein validiertes Vorgehensmodell zur Evaluation der Usability von Modellierungssprachen und ihrer Modelle entwickelt:

Schritt 1: Festlegung des Modellierungszwecks

Informationsmodelle dienen regelmäßig der Erreichung und Unterstützung von Unternehmenszielen [FrvL03, S. 51 - 60]. Je nach Modellierungszweck sind unterschiedliche Ausprägungen des Untersuchungsdesigns erforderlich. Die Qualität eines Informationsmodells bestimmt sich dabei wesentlich durch den jeweiligen Verwendungszweck des Modells [BSG99, S. 22]. Da der Modellierungszweck die Grundlage für die Auswahl der zu untersuchenden Einheiten darstellt, sollte dieser ausreichend detailliert beschrieben werden.

Schritt 2: Auswahl der Untersuchungseinheiten (Modellierungssprachen /-tools)

Das Angebot am Markt existierender Modellierungstools, die regelmäßig mehrere Modellierungsnotationen unterstützen, ist umfangreich. In einer Marktstudie aus dem Jahr 2008 werden allein im Bereich des Business Process Management für den deutschsprachigen Raum 160 Modellierungstools gezählt [IAO08, S. 27]. Bei der Notwendigkeit, das Anwendungswissen von den Anwendern zu erhalten bzw. nach Erfassung zu kommunizieren, gewinnen anschauliche Modelle an Bedeutung [BSG99, S. 7]. Die Sprache, mit der ein Modell entwickelt wird, sollte daher nicht nur hinsichtlich des Modellierungszwecks, sondern auch für den Nutzerkreis geeignet sein [Le06, S. 295]. Nach [BSG99, S. 25] sind anschauliche Prozessmodelle existentiell, damit sich bestehende Schwachstellen schnell erschließen lassen. Die Auswahl der zu untersuchenden Einheiten orientiert sich folglich am Modellierungszweck, dem Benutzerkreis und ggf. weiteren projektbezogenen Anforderungen.

Schritt 3: Bildung von Hypothesen und korrespondierender Aufgaben / Fragestellungen

Hypothesen sind korrigierbare Aussagen über den Wahrheitsgehalt einer Aussage; sie können durch Induktion, Deduktion und Spekulation gewonnen werden [Bu02, S. 277ff., Op02, S. 56, Po73, S. 7]:

- Bei induktiv gewonnenen Hypothesen werden Aussagen mit neuem, erweitertem Gehalt aus singulären Beobachtungen gebildet.
- Bei deduktiv gewonnenen Hypothesen werden Aussagen aus bereits existierenden überprüften Hypothesen (wie Gesetzmäßigkeiten vorhandener Theorien) abgeleitet.
- Bei spekulativ gewonnenen Hypothesen werden Aussagen ohne Beobachtung und ohne sonstige Vorbilder erzeugt. Induktive Fehlschüsse treten damit nicht auf, jedoch ist Spekulation auf Intuition angewiesen und entzieht sich einer Planbarkeit.

Die Art und Weise der Hypothesenbildung ist abhängig vom Modellierungszweck und den angestrebten Zielsetzungen. Hypothesen sind durch geeignete Aufgaben und Fragestellungen zu überprüfen. Dabei ist darauf zu achten, dass diese ausreichend verständlich und eindeutig interpretierbar sind. In wie weit Aufgaben und Fragestellungen geeignet sind, sollte in Voruntersuchungen überprüft werden. Auch in der qualitativen Sozialforschung wird deshalb mit Hypothesen im Sinne von ‚Forschungsfragen‘ gearbeitet. [Fl09, S.139]. Darüber hinaus kann „auch im qualitativen Forschungsprogramm das Testen von Hypothesen einen legitimen Platz einnehmen“ [Me00, S. 274].

Schritt 4: Auswahl der Evaluationstechniken und Instrumente

Unterschiede zwischen Modellierungssprachen lassen sich i.d.R. nicht direkt beobachten, sondern müssen in geeigneter Weise gemessen werden. Messung bedeutet dabei, skalenabhängige Werte von Größen unter genau festgelegten Bedingungen nach be-

stimmt Kriterien zu ermitteln [HKS92, S, 25ff.]. Mit Blick auf die Zielsetzung des Untersuchungsrahmens und der in der Usability-Eyetracking-Forschung regelmäßig eingesetzten Techniken und Instrumente bieten sich Befragungstechniken (z.B. strukturierte Fragebögen) sowie der Einsatz eines Eyetracking-Labors an. Welche Mess- und Bewertungsinstrumente konkret geeignet sind, ist projektspezifisch zu entscheiden. Einschlägige Instrumentarien finden sich bei [Yo03], [RDD08], [Du03], [BPW07], [GoWi03] und [JaKa03].

Schritt 5: Akquirierung der Probanden / Untersuchungsgruppen

Bei der Frage der Usability von Modellierungssprachen und ihrer Modelle stehen zwei Zielgruppen im Fokus: Die Ersteller von Informationsmodellen, die eine Modellierungssprache zur Modellierung einsetzen sowie die Nutzer der erzeugten Informationsmodelle (vgl. Kap. 2). Wenngleich die Meinung vorherrscht, dass 3-8 Probanden genügen, um die meisten Aspekte zu erforschen [SaBr06, S. 167], [Ho05, S. 25f.], [DuRe94, S. 128], [NiLa93, S.206ff.] wird in der nachfolgenden Usability-Eyetracking-Untersuchung eine wesentlich größere Anzahl gewählt, da ein großes Interesse daran besteht möglichst viele Aspekte beurteilen zu können. Gleichwohl ist die Anzahl projektindividuell zu bestimmen. Im prototypischen Anwendungsfall (Abschnitt 4.2) wird die Usability-Eyetracking-Untersuchung mit 24 Probanden durchgeführt.

Schritt 6: Durchführung der Usability-Eyetracking-Untersuchung. Auswertung der Untersuchungsergebnisse und Ableitung von Handlungsempfehlungen

Dieser Schritt bildet die letzte Stufe des Untersuchungsrahmens. Die Usability-Eyetracking-Studie ist durchzuführen und auszuwerten. Die aufgestellten Hypothesen (Schritt 3) werden anhand des gewonnenen empirischen Datenmaterials bewertet. Um eine weitestgehende Neutralität und Objektivität bei den Untersuchungsergebnissen zu erreichen, sollten die Adressaten der Ergebnisse bei der Durchführung und Auswertung der Usability-Eyetracking-Untersuchung möglichst nicht direkt einbezogen werden. Gleiches gilt für die Konstrukteure der Modellierungssprachen und -tools, die als Untersuchungseinheiten ausgewählt werden. Untersuchungsergebnisse und Handlungsempfehlungen sind adressatenspezifisch aufzubereiten.

4.2 Prototypische Anwendung des Untersuchungsrahmens

Die Anzahl der in der Informatik entwickelten Modellierungssprachen verlangt eine Eingrenzung dessen, was in einer Studie zur Usability von Modellierungssprachen untersucht werden kann. Im prototypischen Anwendungsfall werden daher zwei Untersuchungseinheiten für die Durchführung eines Usability-Eyetracking-Vergleiches bewusst ausgewählt. Die Auswahl basiert auf folgenden Kriterien:

- Die Modellierungssprache muss für den geplanten Einsatzzweck einschlägig sein, d.h. mit Blick auf den domänenspezifischen Anwendungsfall geeignet sein, Prozessbeschreibungen grafisch modellieren zu können (Zweckeignung).
- Vor dem Hintergrund praktisch verwertbarer Forschungsergebnisse, werden nur solche Modellierungssprachen in die Überlegungen einbezogen, die in Modellierungstools auch technisch realisiert sind (Toolunterstützung).

Im Rahmen einer pilotierten Usability-Eyetracking-Untersuchung werden, aufgrund ihrer weiten Verbreitung [IAO08], zwei Varianten der Ereignisgesteuerten Prozesskette

(EPK) ausgewählt: die erweiterte EPK (eEPK) und die objektorientierte EPK (oEPK). Arbeiten zu den Varianten finden sich bei [NüRu02], [KNS92], [NüZi98], [SNZ97]. Als Modellierungswerkzeug wird die Bflow* Toolbox [Bfl09] eingesetzt, die beide EPK-Varianten unterstützt. Da für beide Varianten die gleiche Software eingesetzt werden konnte, ist ein Einfluss durch die Softwarewahl auf das Messergebnis zwar möglich, aufgrund der vergleichenden Betrachtung aber zu vernachlässigen. Als Domänebereich dient der öffentliche Sektor. Die Auswahl erfolgt vor dem Hintergrund, dass dieser traditionell durch (teil-)redundante papierbasierte Formulare geprägt ist. Grundvoraussetzung für die Digitalisierung, Bündelung und Virtualisierung der Dienstleistungen ist die Schaffung eines gemeinsamen Modellverständnisses bei allen Beteiligten, sowohl der Ersteller (wie IT-Berater und Organisatoren) als auch der Nutzer (wie Sachbearbeiter und Führungskräfte). Weiterhin kommt der Modellierung im Rahmen der Schaffung der Voraussetzungen zur Umsetzung der EU-Dienstleistungsrichtlinie [EU06] eine besondere Bedeutung zu. Arbeiten hierzu finden sich bei [AlIn08], [BIT08], [DO08], [Gü08], [HoNü08]. In Tabelle 2 werden die Kernelemente des Untersuchungsrahmens bezogen auf den prototypischen Anwendungsfall im Domänebereich des öffentlichen Sektors konkretisiert. Der Anwendungsfall dient einerseits der prototypischen Evaluierung der EPK-Varianten zur Frage der Gebrauchstauglichkeit bei der Modellierung von Verwaltungsprozessen. Andererseits wird der Untersuchungsrahmen für die Durchführung von Usability-Eyetracking-Untersuchungen einer ersten praktischen Validierung unterzogen. Die wiederholte Anwendung bei der Durchführung unterschiedlicher Untersuchungen hat insgesamt ein Vorgehensmodell zur Evaluation der Usability von Modellierungssprachen auf Basis von Usability-Eyetracking-Untersuchungen zum Ziel.

<u>Rahmenkonzept</u> Kernelemente für Usability-Eyetracking-Studien	<u>Anwendungsfall</u> Usability-Eyetracking-Studie zu eEPK/oEPK
<u>Schritt 1:</u> Festlegung des Modellierungszwecks	Modellierung unternehmensbezogener Verwaltungsprozesse zur Umsetzung der EU-Dienstleistungsrichtlinie.
<u>Schritt 2:</u> Auswahl der Untersuchungseinheiten (Modellierungssprachen /-tools)	Vergleich der EPK-Varianten, eEPK und oEPK, technisch unterstützt durch die Bflow* Toolbox als Modellierungswerkzeug.
<u>Schritt 3:</u> Bildung von Hypothesen und korrespondierender Aufgaben / Fragestellungen	Eine Übersicht über die aufgestellten Hypothesen erfolgt in Abschnitt 4.3; die Aufgaben und Fragestellungen finden sich bei [PJN09].
<u>Schritt 4:</u> Auswahl der Evaluationstechniken und Instrumente	Die Usability-Eyetracking-Untersuchung erfolgt auf Basis strukturierter Befragungen und dem Einsatz eines Eyetracking-Labors.
<u>Schritt 5:</u> Akquirierung der Probanden / Untersuchungsgruppen	Die Untersuchung erfolgt auf Basis von 24 Probanden (12 Verwaltungsmitarbeiter und 12 Studenten als Kontrollgruppe). Jeweils die Hälfte beider Gruppen wird als „Ersteller“ eingesetzt; analoges gilt für die „Nutzer“. Die Vergleichbarkeit der Probanden wird durch eine identische Schulung aller Teilnehmer sichergestellt.
<u>Schritt 6:</u> Durchführung der Usability-Eyetracking-Untersuchung, Auswertung und Ableitung von Handlungsempfehlungen	Die Durchführung, Auswertung und Ableitung von Handlungsempfehlungen erfolgt auf Basis der Schritte 1 bis 5. Zu den Kernergebnissen siehe Abschnitt 4.3.

Tabelle 2: Anwendungsfall zum Usability-Eyetracking-Untersuchungsrahmen

4.3 Durchführung der Usability-Eyetracking-Untersuchung

Die Durchführung der Untersuchung (Schritt 6 des Untersuchungsrahmens) folgt dem Schema des Usability Testings nach Rubin [Ru94, S. 79ff.], Dumas und Redish [DuRe94, S. 105ff.]:

- Entwicklung des Testplans (Developing the test plan)
- Akquirierung und Auswahl der Testpersonen (Acquiring and selecting participants)
- Vorbereitung des Tests (Preparing the test materials)
- Durchführung des Tests (Conducting the test)
- Nachbesprechung mit den Probanden (Debriefing the participants)
- Datenanalyse und Auswertung (Transforming data into findings and recommendations).

Für die Untersuchung wurde das Eyetracking-Labor des Forschungsschwerpunktes Kommunikationsforschung an der Fachhochschule Düsseldorf genutzt. Hierbei kam ein Tobii T60 als Eyetracking-System zum Einsatz. Zur Anwendung des Untersuchungsrahmens beim Vergleich der EPK-Varianten, eEPK und oEPK, werden acht Hypothesen (H1 bis H8) formuliert. Diese sind das Ergebnis eines mehrstufigen iterativen Entwicklungsprozesses eines interdisziplinären Projektteams, bestehend aus je drei Experten aus dem Bereich der Kommunikationswissenschaften und der Wirtschaftsinformatik sowie vier Experten aus der öffentlichen Verwaltung. Im Sinne der wissenssoziologischen Definition kennzeichnen sich Experten, z.B. gegenüber spezialisierten Laien, durch ein Sonderwissen, das sie vielfach im Berufszusammenhang erworben haben [Bo05, S. 33ff.]. Alle Experten zeichneten sich durch mehrjährige Erfahrung in ihren Forschungsgebieten und die Durchführung einer Vielzahl von Projekten aus.

Gruppe Ersteller

- H.1: Ein vorgegebener Prozess kann von den Probanden mittels der EPK-Varianten, eEPK und oEPK, dargestellt werden (Fokus: Effektivität – grundsätzliche Eignung).
- H.2: Ein mittels oEPK aufgezeichneter Prozess enthält weniger Fehler als ein mittels eEPK dargestellter Prozess (Fokus: Effektivität - Output im Informationsverarbeitungsprozess).
- H.3: Ein textlich vorgegebener Prozess kann mittels oEPK schneller modelliert werden als mittels eEPK (Fokus: Effizienz - Prozessmodellierung).
- H.4: Die Probanden geben in der Befragung an, dass die Darstellung eines Prozesses mittels oEPK schneller durchgeführt werden kann als ein eEPK-Prozess (Fokus: Effizienz - Methodik).
- H.5: Die Probanden haben nach dem Test eine positivere Einstellung zur oEPK-Notation als zur eEPK-Notation (Fokus: Zufriedenheit - Methodik).

Gruppe Nutzer

- H.6: Ein mittels oEPK dargestellter Prozess wird von Probanden besser verstanden als ein mittels eEPK dargestellter Prozess (Fokus: Effektivität - Input im Informationsverarbeitungsprozess).
- H.7: Ein mittels oEPK dargestellter Prozess kann von den Probanden mit weniger

Aufwand verarbeitet werden als ein eEPK-Prozess (Fokus: Effizienz - Informationsverarbeitung).

H.8: Ein mittels oEPK dargestellter Prozess wird von den Probanden positiver wahrgenommen als ein eEPK-Prozess (Fokus: Zufriedenheit - Prozessdarstellung).

Ad. H.1: Alle Probanden konnten sowohl mit der eEPK als auch der oEPK den vorgegebenen Prozess (einen Teilprozess aus dem Verwaltungsverfahren zur Gewerbe-Anmeldung) darstellen. Hypothese 1 ist damit vorläufig bestätigt. Aufgrund der kleinen Stichprobe wurde auf die Durchführung statistischer Hypothesentests verzichtet. Für eine quantitative Modell- und Hypothesenbildung sind weitere Untersuchungen mit einer größeren Anzahl von Probanden notwendig. Die Hypothese kann folglich nur „vorläufig“ bestätigt werden. Dies gilt auch für alle folgenden Hypothesen im oben dargestellten Sinne von qualitativen Forschungsfragen.

Ad. H.2: Bei der Modellierung mittels eEPK traten insgesamt 121 Fehler auf (vgl. Tabelle 3), bei der Modellierung mittels oEPK 31 Fehler. Basis für die Analyse bildete ein strukturierter Bewertungsrahmen mit 22 Fehlerkategorien [PJN09, S. 111]. Auch ein Vergleich zwischen den beiden Gruppen zeigt eine vergleichbare Tendenz, wobei die Studenten insgesamt deutlich weniger Fehler machten als die Verwaltungsmitarbeiter. Hinsichtlich der dokumentierten Fehleranzahl ergibt sich folglich ein deutlicher Unterschied zwischen eEPK und oEPK. Die Hypothese 2 wird damit vorläufig angenommen.

	Anzahl Fehler eEPK	Anzahl Fehler oEPK
Verwaltungsmitarbeiter	99	24
Studierende	22	7
Gesamt	121	31

Tabelle 3: Fehler bei der Modellierung mit eEPK im Vergleich zu oEPK (Ersteller)

Ad. H.3: Mit Blick auf Tabelle 4 wird ein Prozess im Durchschnitt mittels oEPK im Test ca. 32% schneller modelliert als der identische Prozess mittels eEPK. Die Differenz fällt dabei bei der Kontrollgruppe „Studenten“ mit rd. 46% deutlicher aus als bei der Gruppe der „Mitarbeiter“ mit rd. 18%. Die Hypothese kann damit bestätigt werden.

	eEPK Dauer (in Min.)	oEPK Dauer (in Min.)	Differenz (Min.)	Differenz (in Prozent)
Verwaltungsmitarbeiter	9,4	7,7	-1,7	-18,06%
Studenten	9,5	5,1	-4,4	-46,27%
Gesamt	9,4	6,4	-3,0	-32,24%

Tabelle 4: Vergleich der Modellierungsdauer eEPK zu oEPK (Ersteller)

Ad. H.4: Zur Überprüfung der Hypothese 4 wurden die Ersteller nach erfolgter Modellierung befragt, mit welcher EPK-Variante sie den Prozess schneller modellieren konnten. Zur Modellierungsvariante eEPK äußerte dies keiner der Probanden. Elf der Ersteller bestätigten dies hingegen bezogen auf die oEPK; einer machte keine Angaben (Tabelle 5).

le 5). Das subjektive Empfinden deckt sich demnach mit den objektiven Zeitmessungen (Tabelle 4). Die Hypothese 4 wird vorläufig angenommen.

Ersteller	eEPK	oEPK	k.A.	Gesamt
Mit welcher EPK-Variante haben Sie den Prozess schneller modelliert?	0	11	1	12
in Prozent	0%	91,67%	8,33%	100%

Tabelle 5: Vergleich der Modellierungsdauer eEPK zu oEPK (Ersteller)

Ad. H.5: Zur Beurteilung der Hypothese 5 (zur Frage, zu welcher EPK-Variante die Probanden nach dem Test eine positivere Einstellung haben), wurden vier Teilfragen gestellt, was eine differenzierte Bewertung ermöglicht. Im Ergebnis erhält die oEPK bezogen auf alle vier Teilfragen eine höhere Zustimmung der Probanden als die eEPK (vgl. Tabelle 6). Die Hypothese wird damit vorläufig angenommen.

Weiterempfehlungsquote	eEPK	oEPK	k.A.	Gesamt
Teilfrage 5.1: Welche Modellierungssprache würden Sie weiterempfehlen?	4	18	2	24
Ersteller u. Nutzer	17%	75%	8%	100%
Wiederbenutzungsquote	eEPK	oEPK	k.A.	Gesamt
Teilfrage 5.2: Welche Modellierungssprache würden Sie wieder benutzen?	5	15	4	24
Ersteller u. Nutzer	21%	63%	17%	100%
Verständlichkeitsgrad	eEPK	oEPK	k.A.	Gesamt
Teilfrage 5.3: Welche Modellierungssprache finden Sie besser verständlich?	6	16	2	24
Ersteller u. Nutzer	25%	67%	8%	100%
Nutzbarkeitsgrad	eEPK	oEPK	k.A.	Gesamt
Teilfrage 5.4: Mit welcher Modellierungssprache kamen Sie bei der Umsetzung der textlichen Prozessbeschreibung besser zurecht?	3	6	3	12
Ersteller	25%	50%	25%	100%

Tabelle 6: Subjektive Einstellung zu eEPK und oEPK (Ersteller und Nutzer)

Ad. H.6: Zur Beurteilung der Hypothese 6 wurden den Nutzern, auf Grundlage von zwei modellierten Prozessausschnitten, in eEPK und oEPK, Verständnisfragen gestellt [PJN09, S. 92-100]. Es waren jeweils vier Antwortmöglichkeiten vorgegeben, von denen nur zwei zutrafen. Tabelle 7 zeigt die Ergebnisse dieses Testes. Der Vergleich zeigt, dass bei der eEPK etwas mehr richtige Antworten als bei der oEPK gegeben wurden (66>61). Die Hypothese 6 wird vorläufig nicht angenommen. Der Unterschied zur oEPK ist aber marginal. Das Ergebnis aus der Verständnisüberprüfung deckt sich dabei nicht mit den

Ergebnissen zur Frage nach dem subjektiven Empfinden zur Verständlichkeit (vgl. Hypothese 5).

Verständnisfragen zum eEPK-Prozessmodell	richtige Antworten	falsche Antworten	keine Entscheidung	gesamt
Mitarbeiter	33	10	5	48
Studenten	33	13	2	48
gesamt	66	23	7	96
Verständnisfragen zum oEPK-Prozessmodell	richtige Antworten	falsche Antworten	keine Entscheidung	gesamt
Mitarbeiter	29	18	1	48
Studenten	32	14	2	48
gesamt	61	32	3	96

Tabelle 7: Verständnisüberprüfung eEPK zu oEPK (Nutzer)

Ad. H.7: Die Hypothese 7 beschäftigt sich mit der Frage, bei welcher EPK-Variante der Proband die Informationen mit weniger Aufwand verarbeiten kann. Um wissenschaftliche Erkenntnisse über die „unsichtbare“ menschliche Informationsverarbeitung zu gewinnen, legt die Kognitionspsychologie beobachtbare Hilfsgrößen fest, die in einer Wahrscheinlichkeit mit dem eigentlichen Erkenntnisobjekt, den kognitiven Prozess, korrelieren [Pa06, S. 126]. Zur Beurteilung dieser Hypothese wird das Instrumentarium der Blickbewegungsregistrierung eingesetzt. Für die Nutzer erfolgt die Auswertung des Informationsverarbeitungsaufwands sowohl anhand von sog. Heatmap-Darstellungen, als auch mittels Zeit- und Fixationsmessungen [HNP09]. Bei der Anwendung von Heatmaps werden Aufmerksamkeitsschwerpunkte der Probanden auf die Elemente der Modelle visualisiert. Grundlage für die ausgewählten Darstellungen war eine Aufgabe an die Probanden, einen relevanten Prozessteil zu finden und für mindestens 3 Sekunden darauf zu schauen. Abbildung 1 zeigt beispielhaft die Heatmaps am Ende der Betrachtungsphase (letzten 30 Sek.). Hierbei handelt es sich um automatisch generierte Abbildungen der Auswertungssoftware Tobii Studio 1.5 und eine kumulierte Darstellung der Fixationsverteilung. Das gesuchte Element ist jeweils blau hervorgehoben. Im Gegensatz zur oEPK-Darstellung ist beim eEPK-Modell deutlich zu erkennen, dass die Aufmerksamkeitsschwerpunkte länger im mittleren Teil des Modells verbleiben (rot umkreist). Dies ist ein Indiz aus der grafischen Analyse, dass bei der oEPK ein geringerer Aufwand bei der Informationsverarbeitung auftritt als bei der eEPK, was durch die objektiven Zeit- und Fixationsmessungen (Tabelle 8) bestätigt wird. Die Hypothese 7 wird damit vorläufig angenommen. Die Ergebnisse für die Ersteller zeigen ein vergleichbares Ergebnis [HNP09], [PJN09].

Aufgabe 7.1: Schauen Sie bitte für mindestens 3 Sekunden auf den eEPK-Prozesseil, wo die Mitteilung an den Antragsteller gesendet wird!

Gruppe Nutzer	Ø Dauer bis Symbol angesehen (Sek.)	Ø Länge der Fixationen (in Sek.)	Ø Anzahl der Fixationen
Verwaltungsmitarbeiter	35,99	1,45	3,50
Studenten	42,13	4,32	8,00
Ø Gesamt	39,02	2,86	5,75

Aufgabe 7.2: Schauen Sie bitte für mindestens 3 Sekunden auf den oEPK-Prozesseil, wo die Mitteilung an den Antragsteller gesendet wird!

Gruppe Nutzer	Ø Dauer bis Symbol angesehen (Sek.)	Ø Länge der Fixationen (in Sek.)	Ø Anzahl der Fixationen
Verwaltungsmitarbeiter	26,04	0,68	1,8
Studenten	28,52	2,85	5,3
Ø Gesamt	27,28	1,77	3,55

Tabelle 8: Zeit- und Fixationsmessung eEPK und oEPK (Nutzer)

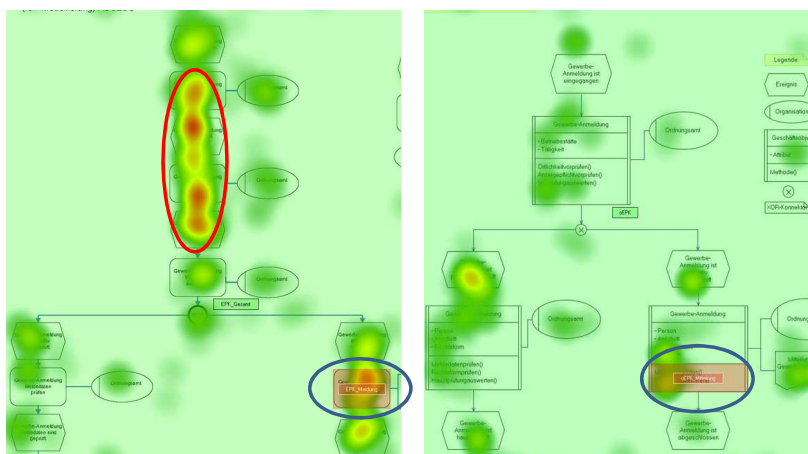


Abbildung 1: Heatmaps Nutzer - Zeichnung eEPK (links) und oEPK (letzten 30 Sek.)

Ad. H.8: Hypothese 8 beschäftigt sich mit der Frage, bei welcher EPK-Variante der Proband die Informationsdarstellung im Prozessmodell positiver wahrnimmt. Als Indikator dient hier der Grad der Zufriedenheit, welchen die Probanden bewerten konnten. Der Vergleich in Tabelle 9 zeigt, dass diese in ihrer subjektiven Einschätzung insgesamt etwas zufriedener mit der oEPK als mit der eEPK sind (oEPK: 33% sehr zufrieden und 50% zufrieden; eEPK: 0% sehr zufrieden und 83,3% zufrieden). Vier der Probanden sind mit der oEPK „sehr zufrieden“ und sechs „zufrieden“, wohingegen mit der eEPK insgesamt zehn „zufrieden“ sind. Die Hypothese wird vorläufig angenommen. In einer Untersuchung zu B2B-Websites stellt Nielsen et al. [NLN07, S.25] eine positive Korrelation zwischen dem (objektiven) Grad der Aufgabenerfüllung und der (subjektiven) Zufrie-

denheit von Testteilnehmern fest. Ob dies auf die Usability von Modellierungssprachen übertragbar ist, bleibt weiteren Studien vorbehalten, die Ergebnisse dieser Untersuchung geben hierzu erste Hinweise.

Frage 8.1: Wie zufrieden sind sie mit der Darstellung des Prozessablaufs in eEPK?

Nutzergruppe	sehr zufrieden	zufrieden	unzufrieden	sehr unzufrieden	keine Angaben
Mitarbeiter	0	6	0	0	0
Studenten	0	4	2	0	0
Gesamt	0	10	2	0	0
in %	0%	83,3%	16,7%	0%	0%

Frage 8.2: Wie zufrieden sind sie mit der Darstellung des Prozessablaufs in oEPK?

Nutzergruppe	sehr zufrieden	zufrieden	unzufrieden	sehr unzufrieden	keine Angaben
Mitarbeiter	2	3	1	0	0
Studenten	2	3	1	0	0
Gesamt	4	6	2	0	0
in %	33,3%	50%	16,7%	0%	0%

Tabelle 9: Zufriedenheitsgrad eEPK zu oEPK (Nutzer)

5 Erkenntnisse und weiterer Forschungsbedarf

5.1 Erkenntnisse

Der vorliegende Beitrag stellt die Ergebnisse einer explorativen Studie zur Gebrauchstauglichkeit von EPK-Varianten vor, die sowohl auf Ersteller als auch auf Nutzer von Modellierungssprachen fokussiert. Als Grundlage zur Messung und Bewertung der Usability der EPK-Varianten, eEPK und oEPK, wird ein Untersuchungsrahmen zugrunde gelegt, der Konzepte der Wirtschaftsinformatik mit Konzepten der Kommunikationsforschung kombiniert. Dabei werden sowohl einschlägige Teile der Usability-Norm EN ISO 9241 einbezogen, als auch die Methode des Eyetrackings (Blickbewegungsregistrierung). Die Durchführung der Usability-Eyetracking-Untersuchung erfolgt in Form eines explorativen Experimentes, wobei der Vergleich der EPK-Varianten technisch durch das Open Source-Modellierungswerkzeug der Bflow* Toolbox unterstützt wird. Von acht zugrunde gelegten Hypothesen werden sieben vorläufig bestätigt. Die Hypothese H6 (Verständnisüberprüfung) kann als einzige nicht eindeutig beurteilt werden, hier waren die Ergebnisse zwischen eEPK und oEPK gleichtendierend. Insgesamt wurde der Test mit 24 Probanden in zwei Gruppen durchgeführt, jeweils eingeteilt in Verwaltungsmitarbeiter und Studierende als Kontrollgruppe. Deutliche Unterschiede gab es zugunsten der Gruppe der Studierenden bei der geringeren Anzahl von Modellierungsfehlern (Hypothese 3) und der geringeren Modellierungsdauer (Hypothese 4). Die Gruppe der Verwaltungsmitarbeiter schloss hingegen bei der Orientierung in den Prozessmodellen besser ab (Hypothese 7). Insgesamt sprechen die Ergebnisse dafür, die oEPK-Notation der eEPK-Notation bei der Modellierung von Verwaltungsprozessen vorzuziehen.

5.2 Limitationen und weiterer Forschungsbedarf

Der dargestellte Untersuchungsrahmen wird auf Basis der EPK-Varianten, eEPK und oEPK, erstmals prototypisch angewandt. In wieweit andere Modellierungssprachen / -tools für die Zielsetzung geeignet sind, bleibt weiteren Untersuchungen vorbehalten. Die Ergebnisse der Usability-Eyetracking-Untersuchung geben Hinweise zur Usability von eEPK- und oEPK-Notationen für den Einsatz in der öffentlichen Verwaltung, diese sind im Rahmen weiterer Untersuchungen zu validieren. Durch die Anwendung des Untersuchungsrahmens bei der Durchführung unterschiedlicher Usability-Eyetracking-Studien wird sukzessive ein Vorgehensmodell zur Usability von Modellierungssprachen und ihren Modellen entwickelt. Diese und weitere wissenschaftliche Fragestellungen werden im Rahmen zukünftiger Arbeiten zu untersuchen sein. Hier stellt sich eine Vielzahl weiterer Forschungsansätze, die in Form von disziplinübergreifenden Kooperationsprojekten zwischen Vertretern der Integrationsdisziplin Wirtschaftsinformatik und der Kommunikationsforschung realisiert werden können.

Literatur

- [Alln08] Algermissen, L.; Instinsky, M. (2008): Prozessmanagement für Verwaltungen als Grundlage verbesserter Dienstleistungen – Motivation einer prozessorientierten Umsetzung der EU-Dienstleistungsrichtlinie, in: Stember, J.; Göbel, A. (Hrsg.): *Verwaltungsmanagement für Unternehmen: Zwischen EU-Dienstleistungsrichtlinie, Bürokratieabbau und Wirtschaftsförderung*. Münster 2008, S. 277-298.
- [BaPe83] Bailey, J. E.; Pearson, S. W.: Development of a Tool for Measuring and Analysing Computer User Satisfaction, in: *Management Science*, Vol. 29, No. 5, May 1983, S. 530 – 545.
- [Bfl09] Bflow Toolbox. Online verfügbar unter: <http://sourceforge.net/projects/bflowtoolbox/>, zuletzt geprüft am 09.09.2009.
- [BIT08] BITCOM (2008): Empfehlungen zur Umsetzung der EU-Dienstleistungsrichtlinie (20.03.2008): Online verfügbar unter: http://213.216.17.150/DOL/Anlagen/Anlage_A2_BITKOM-Empfehlungen.pdf, zuletzt geprüft am 08.09.2009.
- [Bo05] Bogner, A.; Menz, W. (2005): Das theoriegenerierende Experteninterview – Erkenntnisinteresse, Wissensformen und Interaktion., in: Bogner, A.; Littig, B.; Menz, W. (Hrsg.): *Das Experteninterview – Theorie, Methode, Anwendung*, Verlag für Sozialwissenschaften, Wiesbaden, 2005; S. 33-70.
- [BPW07] Bernhaupt, R.; Palanque, P.; Winckler, M.; Navarre, D.: Usability Study of Multi-modal Interfaces Using Eye-Tracking, in: Baranauskas, C. et al. (Eds.): *IFIP International Federation for Information Processing (INTERACT 2007)*, 2007, S. 412 – 424.
- [BSG99] Becker, J.; Schütte, R.; Geib, T.; Ibershoff, H.: Grundsätze ordnungsgemäßer Modellierung (GoM). Abschlussbericht, 1999. Online verfügbar unter: <http://emedien.sub.uni-hamburg.de/han/gvk-plus/edok01.tib.uni-hannover.de/edoks/e001/303489413.pdf>, zuletzt geprüft 07.09.2009).
- [Bu02] Bunge, M.: *Philosophy of Science: From Problem to Theory* (Vol. I). Transaction Publ., London, 2002.
- [Ca06] Carmichael, A.: *Style Guide for the Design of Interactive Television Services for Elderly Viewers*. 2006
<http://www.computing.dundee.ac.uk/projects/UTOPIA/publications/Carmichael%20-%20DesignStyleGuideFinal.pdf>, zuletzt geprüft am 07.09.2009.

- [CHL06] Cyr, D.; Head, M.; Larios, H.; Pan, B.: Exploring Human Images in Website Design Across Cultures: A Multi-Method Approach, in: Proceedings of Special Interest Group on Human-Computer Interaction (SIGHCI 2006), 2006, S. 55 – 59.
- [Da89] Davis, F.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, Vol. 13, No. 3, 1989, S. 319 – 339.
- [DO08] Deutschland Online (2008): Deutschland-Online-Vorhaben: IT-Umsetzung der Europäischen Dienstleistungsrichtlinie. Projektbericht (24.09.2008). Online verfügbar unter: http://www.deutschland-online.de/DOL_Internet/binarywriterservlet?imgUid=49e386de-b9fc-c11d-88ef-1ac0c2f214a8&uBasVariant=22222222-2222-2222-2222-222222222222, geprüft am 08.09.2009.
- [DuRe94] Dumas, J. S.; Redish, J. G.: A practical guide to usability testing. Ablex Publ. Norwood NJ, 1994.
- [Du03] Duchowski, A. T.: Eye Tracking Methodology: Theory and Practice. Springer, London, 2003.
- [Du01] Dumke, R.: Software Engineering: Eine Einführung für Informatiker und Ingenieure: Systeme, Erfahrungen, Methoden, Tools. Vieweg Verlag, Braunschweig/Wiesbaden, 1993.
- [EU06] Europäisches Parlament und Europäischer Rat (2006): Richtlinie 2006/123/EG über Dienstleistungen im Binnenmarkt (EU-Dienstleistungsrichtlinie). 2006/123/EG, vom 12.12.2006. In: Amtsblatt der Europäischen Union, Brüssel.
- [Fl09] Flick, U. (2009): Qualitative Sozialforschung – Eine Einführung; Rowohlt Verlag, Reinbeck bei Hamburg.
- [FrvL03] Frank, U.; van Laak, B. L.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 34, Universität Koblenz/Landau, 2003.
- [Ga07] Gartner Research Group (eds.): Magic Quadrant for Business Process Analysis Tools, M. J. Blechar.
- [GoWi03] Goldberg, H. J., Wichansky, A. M.: Eye tracking in usability evaluation: A practitioner's guide, in: HyÖna, J., Radach, R., Deubel, H. (eds.): The mind's eye: Cognitive and applied aspects of eye movements research, Elsevier, Amsterdam, 2003, S. 493 – 516.
- [Gü08] Günther, O.: Forschungsprojekt "Prozess-Blaupause für die EU-Dienstleistungsrichtlinie", Institut für Wirtschaftsinformatik, Berlin. Online verfügbar unter: <http://www.prozessbibliothek.de/results.php?detail=fachprozesse>, 2008, geprüft am 08.09.2009.
- [HaBa94] Hartwick, J.; Barki, H.: Explaining the role of user participation in information system use, in: Management Science, Vol. 40, No. 4, 1994, S. 440 – 465.
- [HaSc00] Harms, I.; Schweibenz, W.: Usability Engineering Methods for the Web. Results from a Usability Study, in: Knorz, G.; Kuhlen, R. (Hrsg.): Informationskompetenz - Basiskompetenz in der Informationsgesellschaft, Proceedings des 7. Internationalen Symposiums für Informationswissenschaft (ISI 2000, Dieburg 8.-10.11.2000). Schriften zur Informationswissenschaft 38, Konstanz, 2000, 17-30.
- [HKS92] Hellmund, U.; Klitzsch, W.; Schumann, K.: Grundlagen der Statistik. Moderne Industrie, Landsberg/Lech, 1992.
- [HNP09] Hoglebe, F.; Nüttgens, M.; Pagel, S. (2009): Einsatz von Eyetracking zur Messung und Bewertung der Usability von Modellierungssprachen für das Prozessmanagement. Ergebnisse eines Experimentes zum praktischen Nutzen von Heatmaps und Gazeplots. ERP Management, Schwerpunktheft Usability, Nr. 4/2009. [im Druck]
- [HoNü09] Hoglebe, F.; Nüttgens, M. (2009): Rahmenkonzept zur Messung und Bewertung der Gebrauchstauglichkeit von Modellierungssprachen: Literaturlauswertung und Untersuchungsrahmen für Usability-Eyetracking-Studien, in Nüttgens, M. (Hrsg.): Arbeitsberichte zur Wirtschaftsinformatik der Universität Hamburg Nr. 7, 04/2009, Hamburg 2009.

- [HoNü08]Hogrebe, F.; Nüttgens, M. (2008): Integrierte Produkt- und Prozessmodellierung: Rahmenkonzept und Anwendungsfall zur EU-Dienstleistungsrichtlinie, in: Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D. (Hrsg.): MobIs 2008 - Modellierung betrieblicher Informationssysteme: Modellierung zwischen SOA und Compliance Management, Proceedings GI-Tagung, Saarbrücken, 2008, S. 239-252.
- [Ho05] Holzinger, A.: Usability Engineering Methods for Software Developers, in *Communications of the ACM*, 48(1), 2005, 71–74.
- [HSS02] Harms, I.; Schweibenz, W.; Strobel, J.: Usability Evaluation von Web-Angeboten mit dem Web Usability Index, in: *Proceedings der 24. DGI-Online-Tagung 2002 - Content in Context*. Frankfurt am Main 4.-6. Juni 2002. Frankfurt/Main: DGI, 2002, 283-292.
- [IAO08] Institut Arbeitswirtschaft und Organisation, Fraunhofer IAO. Spath, D.; Weisbecker, A. (Hrsg.): *Business Process Management Tools 2008 – Eine evaluierende Marktstudie zu aktuellen Werkzeugen*, Fraunhofer IRB Verlag, Stuttgart, 2008.
- [IOB83] Ives, B.; Olson, M. H.; Baroudi, J. J.: The Measurement of User Information Satisfaction, in: *Communications of the ACM*, Vol. 26, No. 10, October 1983, S. 785 - 793.
- [ISO08] Europäische Norm EN ISO 9241-110: Ergonomie der Mensch-System-Interaktion, Teil 110: Grundsätze der Dialoggestaltung, 2006.
- [ISO00] Europäische Norm EN ISO 9241-12: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten; Teil 12: Informationsdarstellung. DIN Deutsches Institut für Normung e.V., Beuth Verlag, Berlin, 2000.
- [ISO99] Europäische Norm EN ISO 9241-11: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten; Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze. DIN Deutsches Institut für Normung e.V., Beuth Verlag, Berlin, 1999.
- [JaKa03] Jacob, R.J.K., Karn, K. S.: Eye tracking in human-computer interaction and usability research: Ready to deliver the promises, in: Hyona, J. (ed.): *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements*, Elsevier, Oxford, 2003.
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.W.: Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). 1992. http://www.wiso.uni-hamburg.de/fileadmin/WISO_FS_WI/Team/Mitarbeiter/Prof._Dr._Markus_Nuettgens/Publikationen/oEPK.pdf, zuletzt geprüft am 08.09.2009.
- [KjSt04] Kjeldskov, J., Stage, J.: New Techniques for Usability Evaluation of mobile systems. *International Journal on Human-Computer Studies* 60(5), 2004, 599 – 620.
- [Le06] Leist-Galanos, S.: *Methoden zur Unternehmensmodellierung. Vergleich, Anwendungen und Integrationspotenziale*. Logos Verlag Berlin 2006.
- [Me00] Meinefeld, W. (2000): Hypothesen und Vorwissen in der qualitativen Sozialforschung, in: Flick, U.; von Kardorff, E.; Steinke, I. (Hrsg.): *Qualitative Forschung – Ein Handbuch*, S. 265-275, Rowohlt Verlag, Reinbeck bei Hamburg.
- [Me90] Melone, N.: A theoretical assessment of the user-satisfaction construct in information system research, in: *Management Science* Vol. 36, No. 1, 1990, S. 76 – 91.
- [Ni99] Nielsen, J.: *Designing Web Usability*. Indianapolis, Indiana: New Riders Publishing, 1999.
- [NiLa93] Nielsen, J.; Landauer, T. K.: A mathematical model of the finding of usability problems," *Proceedings of ACM INTERCHI'93 Conference*, Amsterdam, 1993, S. 206-213.
- [NLN07] Nielsen, J.; Loranger, H.; Nodder C.: *B2B Website Usability: Design Guidelines for Converting Business Users into Leads and Customers*. Nielsen Norman Group, Fremont CA, USA, 2007.
- [NüRu02]Nüttgens, M.; Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten. 2002. http://www.wiso.uni-hamburg.de/fileadmin/WISO_FS_WI/EPK-Community/Promise2002_Nuettgens_Rump.pdf, zuletzt geprüft am: 08.09.2009.

- [NüZi98] Nüttgens, M.; Zimmermann, V. (1998): Geschäftsprozeßmodellierung mit der objektorientierten Ereignisgesteuerten Prozeßkette (oEPK). In: Maicher, M.; Scheruhn, H.-J (Hg.): Informationsmodellierung - Branchen, Software- und Vorgehensreferenzmodelle und Werkzeuge, S. 23-36. Wiesbaden.
- [OBB07] Obrist, M.; Bernhaupt, R.; Beck, E.; Tscheligi, M.: Focusing on Elderly: An iTV Usability Evaluation Study with Eye-Tracking, in: Cesar, P. et al. (Eds.): EuroITV 2007, Springer, Berlin, 2007, S. 66 – 75.
- [Op02] Opp, K.-D.: Methodologie der Sozialwissenschaften: Einführung in Probleme ihrer Theoriebildung und praktischen Anwendung. Westdeutscher Verlag, Wiesbaden, 2002.
- [Pa06] Patig, S.: Die Evolution von Modellierungssprachen (Habil.). Frank & Timme Verlag, Berlin, 2006.
- [PeGr03] Pemberton, L., Griffiths, R.: Usability Evaluation Techniques for Interactive Television, in: Stephanidis, C. (Ed.). Proceedings of HCI International 2003 (4), Universal Access in HCI, 2003, S. 882–886.
- [PJN09] Pagel, S.; Jürgens, A.; Nüttgens, M.; Adelskamp, P.; Hogrebe, F.; Lange, R.; van Kempen, B. (2009): Usability-Analyse von Prozessmodellierungsmethoden. Abschlussbericht. Verbundprojekt der Fachhochschule Düsseldorf, Universität Hamburg, Landeshauptstadt Düsseldorf, Düsseldorf, 07/2009. [*im Druck*].
- [PGJ08] Pagel, S.; Goldstein, S.; Jürgens, A.: Erste methodische Erkenntnisse zur Usability-Analyse von Video-Inhalten auf Websites mittels Eyetracking, in: Brau, Henning; Diefenbach, Sarah; Hassenzahl, Marc; Koller, Franz; Peissner, Matthias; Röse, Kerstin (Hrsg.): Usability Professionals 2008, Stuttgart 2008, S. 177-181.
- [Po73] Popper, K. P.: Logik der Forschung. Mohr, Tübingen, 1973.
- [RDD08] Ramanauskas, N.; Daunys, G.; Dervinis, D.: Investigation of Calibration Techniques in Video Based Eye Tracking System, in: Miesenberger, K. et al. (Eds.): ICCHP 2008, LNCS 5105, Springer, Berlin, Heidelberg, 2008, S. 1208–1215.
- [RoPu07] Rosen, D. E.; Purinton, E.: Website Design: Viewing the Web as a Cognitive Landscape, Journal of Business Research, 57 (7), 2007, 787 – 794.
- [Ru94] Rubin, J.: Handbook of usability testing. How to plan, design, and conduct effective tests. Wiley, New York, 1994.
- [SaBr06] Sarodnick, F.; Brau, H.: Methoden der Usability Evaluation. Wissenschaftliche Grundlagen und praktische Anwendungen. Huber, Bern., 2006.
- [Se97] Seddon, P.: A respecification and extension of the DeLone and McLean model of IS success, in: Information Systems Research, Vol. 8, No. 3, 1997, S. 240 – 253.
- [SNZ97] Scheer, A.W.; Nüttgens, M.; Zimmermann, V. (1997): Objektorientierte Ereignisgesteuerte Prozeßketten (oEPK) – Methode und Anwendung. Institut für Wirtschaftsinformatik, Heft 141, Universität des Saarlandes. Saarbrücken.
- [SSD02] Steinbrück, U.; Schaumburg, H.; Duda S.; Krüger, T.: A Picture Says More than a Thousand Words: Photographs as Trust Builders in E-commerce Websites, CHI '02, Human factors in computing systems, Minneapolis, Minnesota, 2002.
- [St02] Sterne, J.: Web Metrics: Proven Methods for Measuring Web Site Success, Wiley, 2002.
- [Sz96] Szajna, B.: Empirical evaluation of the revised technology acceptance model, in: Management Science, Vol. 42, No. 1, 1996, S. 85 – 92.
- [VMD03] Venkatesh, V.; Morris, M.; Davis, G.; Davis, F.: User acceptance of information technology: Towards a unified view, in: MIS Quarterly, Vol. 27, No. 3, 2003, S. 425 – 478.
- [WiTo05] Wixom, B. H.; Todd, P. A.: A Theoretical Integration of User Satisfaction and Technology Acceptance, in: Information System Research, Vol. 16, No. 1, 2005, S. 85 - 102.
- [Yo03] Yom, Miriam: Web usability von Online-Shops. Better-Solutions-Verl. Gierspeck, Göttingen, 2003.