Requirements-Based Development of BPMN Extensions: The Case of Clinical Pathways

Richard Braun
Technische Universität Dresden
Chair of Wirtschaftsinformatik, esp. Systems Development
Dresden, Germany
richard.braun@tu-dresden.de

Hannes Schlieter
Technische Universität Dresden
Chair of Wirtschaftsinformatik, esp. Systems Development
Dresden, Germany
hannes.schlieter@tu-dresden.de

***Abstract*—In recent years, the Business Process Model and Notation (BPMN) has evolved to one of the most applied modelling General Purpose Languages in the Business Process Management discipline. Due to its application in different domains, it becomes frequently necessary to extend the BPMN by domain-specific concepts. Extending the BPMN fosters an adequate communication between system engineers and domain experts and enhances the semantically correct representation of the domain. Therefore, the BPMN meta-model provides an extension mechanism. However, there is a remarkable lack in procedure models for the design of such extensions. This research article outlines an extension method that focuses on domain analysis, extension requirements and the derivation of domain-specific extension concepts within BPMN. Therefore, the method of STROPPI ET AL. (2011) is extended in regard to domain analysis. The approach is motivated and demonstrated by the case of Clinical Pathways (CPs).**

***Index Terms*—BPMN Extensions, BPMN Extension Method, Conceptual Modelling, Modelling Language Design, Clinical Pathways, E-Health.**

I. INTRODUCTION AND MOTIVATION

In the discipline of Business Information Systems Engineering, conceptual models have been established as a key instrument for documentation, communication and optimization of information systems [6], [28]. Currently, conceptual models are applied in different domains with diverse scopes. For instance, they are used for enterprise architecture management, configuration of workflow management systems, and customization of enterprise-resource-planning software like SAP ®. Conceptual models are also applied for requirement engineering tasks during early stages in software projects, where they are very useful to visualize dependencies between different goals and requirements. In particular, goal-oriented requirement modelling techniques like i* [5], [29], ReqML [18] or the KAOS approach [9], [26] are powerful methods to conduct a requirements analysis efficiently on the basis of conceptual models.

Beside the requirement engineering methods, the Business Process Model and Notation (BPMN) is currently one of the most popular modelling languages for Business Process Management tasks [19]. The BPMN is specified as General Purpose Language, which means that the modelling concepts are generic elements in the context of business processes, without any domain characteristics. Thus, the BPMN is not restricted to a single domain. However, while process analysis, it could be necessary to implement domain-specific concepts to enrich the modelling language by typical domain concepts, which foster a better communication with domain experts and help to describe the domain adequately. Therefore, the BPMN meta-model specification provides extension mechanisms to supplement the basic language set [12]. Nevertheless, it does not give any procedural advises in terms of a procedure model that guides the extension process. While there are various BPMN extensions, only very few research articles explain the meta-model of the extension and consider its validity in terms of the BPMN meta-model (e.g., [20], [25]). Thus, comprehensibility, model quality and model exchangeability are hampered significantly.

One reason for that might be the above mentioned fact, that there is a lack in a detailed procedure model that guides the analysis, design of development of an extension. So far, there is only one approach addressing this problem: STROPPI ET AL. (2011) introduce firstly a procedure model for BPMN extensions, which focuses on the transformation of conceptual domain models into a valid BPMN extension model. The derived extension model can also be transformed into a BPEL models in order to support process model execution [24]. However, STROPPI ET AL. (2011) do not focus on analysing the domain in order to construct the initial domain model that is the foundation for all following transformation steps. Also, it is not considered, whether a domain concept needs to be implemented as new extension concept or whether it is semantically sufficient to represent a concept by basic BPMN elements. With respect to the mentioned general-purpose nature of the BPMN, this aspect can cause controversies.

Considering the numerous publications, which describe a language extension only by implementing a “new” graphical notation or by adding new concepts in an “ad hoc” manner, we illustrate a procedure for a systematic extension design within BPMN. After outlining the procedure model, we demonstrate the first steps of the procedure by the example of building a BPMN extension for clinical processes, the so-called Clinical Pathways (CPs). In particular, we focus on the method steps, which are missing in the approach of STROPPI ET AL. (2011):

978-1-4799-6345-4/14/$31.00 © 2014 IEEE

domain analysis, requirements analysis and comparison between extension need and the BPMN basic elements. For the purposes of a research in progress paper, further research activities are discussed finally.

This research article aims to develop a generic method for the derivation of BPMN extension models. Hence, the article is aligned to the paradigm of design-oriented information systems research [15], [10]. We argue, that a domain-oriented, holistic BPMN extension method can help systems engineers to develop BPMN extensions that are both conform to the BPMN meta-model and satisfy the requirements of domain experts.

II. BPMN Extensibility

BPMN provides an "extension by addition" mechanism ([24], p. 3) that enables the definition and integration of domain-specific concepts and ensures the validity of the BPMN core elements ([12], p. 44). Thus, BPMN provides the possibility to adjust a general-purpose process modelling language for domain-specific elements. In that way, domain-specific modelling approaches can make use of the strengths of the BPMN (e.g., standardized stable core elements; popularity; expressible elements [19]) and add their specific concepts. Thus, the level of reuse is very high, which could lead to a foreshortening of the development time or the reduction of design faults. It seems to be more promising to focus on the development of language extensions instead of building dedicated languages (referring to [2]), which need to contain all basal process modelling aspects (e.g., nodes, control flows, decision points). The following section describes the BPMN extension mechanism and the approach of Stroppi et al. (2011) in order to present the state of the art and emphasize existing shortcomings.

BPMN specifies the following elements for the definition of valid BPMN extensions: An *Extension Definition* is a named group of new attributes, which can be used by BPMN elements. Thus, new elements can be built implicitly. An *Extension Definition* consists of several *Extension Attribute Definitions* that define particular properties. The *Extension Attribute Value* class specifies possible values of the *Extension Attribute Definitions*. Therefore, primitive types from the Meta Object Facility (MOF) can be used [13]. The element *Extension* binds the entire extension definition and its attributes to a BPMN model definition. By doing so, all extension elements are accessible for existing BPMN elements [12]. Despite the fact that BPMN offers a well-defined extension interface, there is no procedure model for guiding extension development.

To the best of our knowledge, there is only one research article addressing the stated problem: Stroppi et al. (2011) define a model-transformation based procedure model for the methodical development of valid BPMN extensions. The procedure model consists of the following steps ([24], p. 5):

1. Conceptualizing the domain by defining a Conceptual Domain Model of the Extension (CDME) as UML class diagram [14]
2. Transformation of the CDME into a valid BPMN extension model by using UML profiles (BPMN+X)
3. Transformation of the BPMN+X model into a XML Schema Extension Definition Model
4. Transformation of the XML Schema Extension Definition Model into a XML Schema Extension Definition Document

The approach of Stroppi et al. (2011) is based on model transformation rules and makes use of UML profiles to point out several types of classes within the extension definition. Due to the fact that our research work focuses primarily on domain aspects, only the first two steps of the extension method will be considered in the following.

In the first step, a UML class diagram of the domain is designed (CDME). Classes of the CDME model are typed as standard elements from BPMN (*BPMN Concepts*) or new extension elements (*Extension Concepts*; [24], p. 7). In the next step, the BPMN+X model is derived by the application of 15 model transformation rules for all possible CDME configurations ([24], p. 9). As a UML profile, the BPMN+X model consists of several stereotypes that express the role of specific class in context of the extension (e.g., *BPMN Element*, *Extension Element*, *Extension Definition*, *Extension Relationship*; [24], p. 7).

The approach of Stroppi et al. (2011) differentiates between standard and extension elements, but there is no analysis of the question whether a considered domain concept corresponds to the semantics of a BPMN element. Also, the analysis of the domain and the definition of requirements on the extension are not considered, since the approach focuses on the respective model transformations. We argue, that a deeper analysis and explication of the conceptual phase within the development of a BPMN extension is necessary for several reasons:

a) Understanding the domain by the explication of domain requirements and its concepts
b) Check, whether BPMN is the most suitable process modelling language for the considered purpose at all
c) Reasonable explication of extension elements by a detailed comparison with BPMN elements
d) Methodical guidance for the entire extension process: From domain concepts (semiformal, informal) to BPMN and BPEL concepts (formal, semiformal)

Thus, it is possible to evolve a consistent method for the development of BPMN extensions that gives strong guidance to the method developer (systems engineer) and enables a reasonable design of extension elements. Further, the aimed method could help to minimize the semantic gap between the domain and possible workflow systems. Also, comprehensibility and traceability can be improved.

III. Method for the Development of BPMN Extensions

Figure 1 outlines the proclaimed method for the domain-oriented development of BPMN extensions. The method consists of six procedure steps that extend the approach of Stroppi et al. (2011): First, a requirements analysis based on domain use scenarios or literature reviews has to be conducted in order to define domain requirements. This step captures the need for a domain-specific modelling technique that provides concepts for problem solutions. Subsequently, it needs to be analysed whether BPMN is generally able to satisfy the identified requirements. Therefore, alternative modelling languages should be examined. At this point, a "go or stop" decision is made regarding the application of the BPMN. If the decision is positive, a domain ontology has to be built in order to conceptualize the context. Thus, first concepts, its relations and properties are explicated. This ontology is very important for the third step of the method, the element equivalence check. Within this step, all semantically described elements are examined regarding their matching to basic BPMN elements. This check allows the identification of elements that need to be defined as BPMN extension. Thus, the output of that step is a set of "extension requirements". The fourth and fifth step of the method are borrowed from the extension method of STROPPI ET AL. (2011) and result in a valid BPMN extension model (abstract syntax of the extension). The last step of the proposed method contains the definition of the concrete syntax of the extension and adds the graphical elements according to BPMN guidelines.

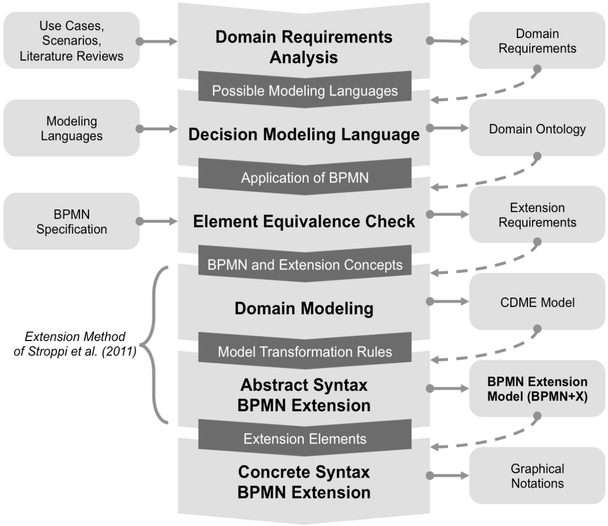
Use Cases, Scenarios, Literature Reviews
Domain Requirements Analysis
Domain Requirements
Possible Modeling Languages
Modeling Languages
Decision Modeling Language
Domain Ontology
Application of BPMN
BPMN Specification
Element Equivalence Check
Extension Requirements
BPMN and Extension Concepts
Extension Method of Stroppi et al. (2011)
Domain Modeling
CDME Model
Model Transformation Rules
Abstract Syntax BPMN Extension
BPMN Extension Model (BPMN+X)
Extension Elements
Concrete Syntax BPMN Extension
Graphical Notations

Fig. 1: Designed method for the domain-oriented development of BPMN extensions.

A. Domain Requirements Analysis

This step aims to both understand the domain in detail and derive requirements to the modelling approach [17]. Requirements are understood as the explication of domain concepts and configurations that need to be covered by the language (functional requirements) and also as additional constraints such as language extensibility (referring to [17]). Regarding to Requirements Engineering (RE), it is necessary to focus on the early-phase RE and its respective techniques [5], [29]. If there is a group of users of the prospective modelling language (e.g., practitioners), the description of use cases or use scenarios can be applied in order to identify requirements. If there is no specific user group, also literature reviews can provide insights into requirements (in a more argumentative manner). The identified domain requirements to the modelling language should be explicated by a semiformal requirements modelling approach due to the identification of concepts (see step 2). Therefore, i*, KAOS or ReqML can be applied [29], [26], [18]. We recommend the KAOS approach since its integration of the goal and the object perspective supports the identification of relevant concepts.

B. Selection of Modelling Language

Based on the previously derived requirements, it has to be checked whether the BPMN as process modelling language is generally usable or other modelling languages are more appropriate. Therefore, techniques for the comparison of modelling grammars can be utilized (e.g., [7], [27], [28]). This step should ensure, that BPMN is the appropriate modelling language for the addressed purpose. Within this step, it is further recommended to build a domain ontology based on the requirements. Thus, relevant concepts, their relations and attributes are structured and explicated in order to prepare the concept-based comparison in step 3. The ontology language OWL is recommended for that purpose [1].

C. Element Equivalence Check

This step contains an equivalence check in order to consider whether the domain concept is semantically covered by an existing BPMN element or not. In that way, the CDME model and its stereotypes are prepared. The following comparison rules are defined:

Equivalence: There is a semantically equivalent construct in the BPMN. A construct is understood as a permitted combination of elements or a single element. In this case, no extension is necessary (→ CDME: *BPMN Concept*).

Conditional equivalence: There is no obvious semantic matching with BPMN elements. Rather, it has to be decided individually whether the semantics of a basic BPMN concept still corresponds to the semantics of a domain-specific concept. Therefore, it is necessary to give reasons for a possible mapping or to explain why it is not reasonable. This discussion is necessary since the BPMN meta model specifies some elements in a very wide range (e.g., [12], p. 306). Methodical support for the discussion comes from the field of model comparison (e.g., [7], [8], [11]).

Depending on the result of the discussion, the concept is treated as ”equivalence” or ”no equivalence”.

No equivalence: There is no equivalence to any BPMN element for three reasons. First, the entire concept is missing (→ CDME: *Extension Concept*). Second, a relationship between two concepts is missing (→ CDME: *Association between concepts*). And third, attributes owned by a concept are missing (→ CDME: *Property of a concept*).

D. Domain Modelling and Abstract Syntax of the Extension

The previous identified extension concepts are the input for building the CDME model. Afterwards, model transformation rules have to be applied in order to generate a valid BPMN extension model (see [24] and section II).

E. Concrete Syntax of the BPMN Extension

With respect to the objective of this paper, step 3 and step 4 of the procedure model of STROPPI ET AL. (2011) will not be applied, since these steps focus the derivation of executable BPEL models and XML documents. Instead, the concrete syntax of the BPMN extension should be designed with respect to the graphical restriction of the BPMN specification: "An extension SHALL NOT change the specified shape of a defined graphical element or marker (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.)" ([12], p. 8).

IV. DEMONSTRATION: BPMN EXTENSION FOR CLINICAL PATHWAYS

This section demonstrates the application of the presented extension method. With respect to the limited space of this paper, only the first two steps will be presented in detail (requirements and domain ontology).

A. Domain Requirements

Modelling medical treatment processes by GPL encounters rapidly the limits of language capability, because medical processes are subjected to various influences and they rarely follow a linear, predictable process sequence. Currently, there is a range of contributions, which analyse requirements for modelling languages of medical treatment processes.

The first language evaluations on CPs are carried out by medical informatics with regard on formalization of the clinical algorithm and its computer-based interpretation [16]. However, they also formulate language requirements, which introduce domain-specific concepts firstly. We consolidate those requirements as the need for basic modelling concepts:

- **Requirement R1**: A language for modelling clinical pathways should provide the basic concepts of the medical business process modelling (patient state, treatment step, decision, process flow) and the ability to integrate information objects and responsibilities.

SARSHAR ET AL. (2004) conduct an analysis in order to evaluate the capability of Event-driven Process Chains (EPC) and their extensions for modelling medical processes [22]. Their requirement set mainly addresses the flexibility of process flows and decisions as well as the modelling of time aspects, which is later consolidate by BURWITZ ET AL. (2013) to following Requirement:

- **Requirement R2**: A language for modelling clinical pathways should provide concepts for describing indefinite order relations as well as compulsory parallel relations between treatment steps and iterating treatment steps.

SCHLIETER (2012) lays the foundation for the requirement set we are using for the language evaluation and extension [23]. He motivates his requirements analysis from the view point how Clinical Practice Guidelines can be modelled and transferred into CPs. Beside more method specific requirements, SCHLIETER (2012) claims that it is necessary to describe the evidence given by clinical practice guidelines for any modelled facts, as well as an reference on the source of evidence. This leads to following requirement:

- **Requirement R3**: A language for modelling clinical pathways should provide concepts for describing evidence-class of any recommendation and linking the source of evidence. Additionally, a concept to describe evidence-based decision is required.

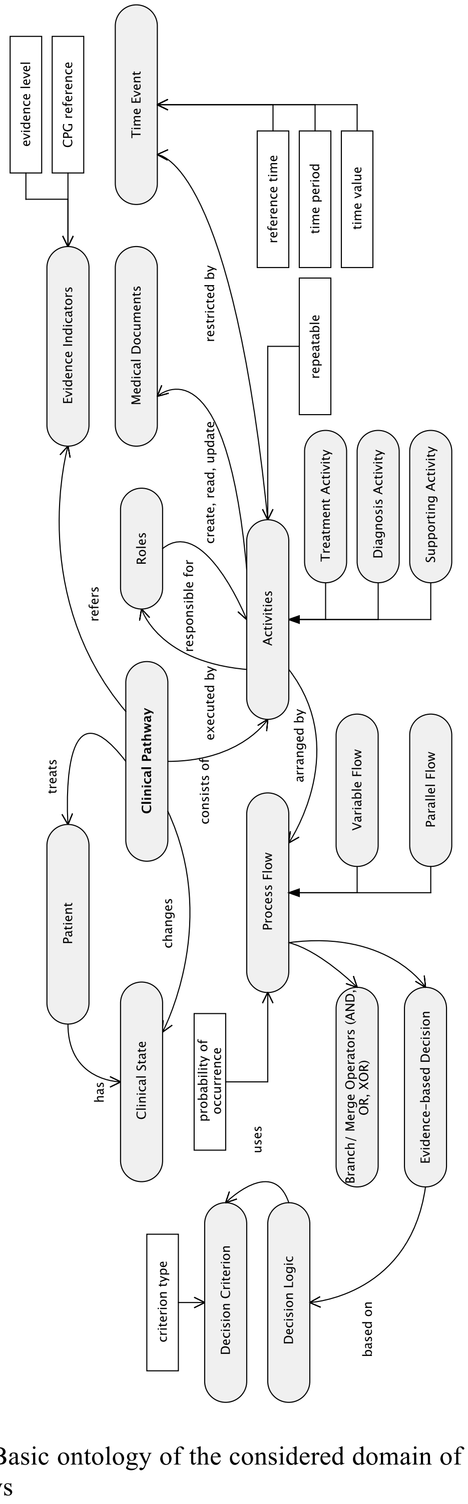

Fig. 2: Basic ontology of the considered domain of clinical pathways

While building a domain specific language for Clinical Pathways, BURWITZ ET AL. (2013) consolidate the work of SASHAR ET AL. (2004) and SCHLIETER (2012) and pick up the aspect temporal dependencies and explicit time events. This leads to following requirement:

- **Requirement R4**: A language for modelling clinical pathways should provide concepts to describe temporal dependencies and explicit time events.

B. Clinical Pathway Ontology

With respect to the second step of the extension method, it has to be decided, whether BPMN is generally appropriate for satisfying the stated requirements. We argue, that BPMN fulfils main aspects of the requirements (e.g., basic concepts of business process modelling, temporal dependencies), but needs domain-specific extension. However, also other process modelling languages such as EPC (ARIS), flow charts or UML activity diagrams would need these extensions. Thus, BPMN is an adequate modelling language for the domain problem.

Figure 2 depicts the evolved domain ontology. The central concept is the "Clinical Pathway", which consists of several activities. A "Process Flow" organises and arranges all activities within the pathway. The concepts "Variable Flow" and "Parallel Flow" address the semantics of requirement R2. Further, a process flow contains several basic decision points (AND, OR, XOR) and evidence-based decision points. These decision points are specified by a defined logic and a set of criteria. Activities can be divided into "Treatment Activities", "Diagnosis Activities" and "Supporting Activities". Activities are executed by a specific "Role" that is responsible for an activity. Also, various documents can be created or updated by some activities. Activities are also restricted or triggered by time events. Within the entire clinical pathway, "Evidence Indicators" can be assigned to different elements in order to explain their evidence level and give a reference to a clinical pathway guideline (CPG).

The depicted concepts are further used as "input elements" for the equivalence check in the third step of the extension method. The implementation of this step and the derivation of both the CDME model and the BPMN+X model will be presented in a further research article [3].

V. CONCLUSION AND FURTHER RESEARCH

The research article provides theoretical implications in the field of method development by extending the method by STROPPI ET AL. (2011) by a requirements-based domain analysis and a comparison of domain-specific concepts with BPMN basic concepts.

In our opinion, there are several aspects for further research both in the field of BPMN extensions and conceptual modelling of CPs. From a methodological point of view, it seems to be necessary to intensify research on the development of holistic procedure models for the design of domain-specific modelling languages and BPMN extensions. Especially, methodical support for the semantic comparison of model concepts needs to be implemented within the outlined method. Also, the integration of structured domain knowledge in terms of ontologies, taxonomies or similar conceptualizations needs to be intensifying for reasons of reutilisation and integration.

Furthermore, there are several aspects for further investigation in the field of conceptual modelling in healthcare. For example, we proclaim the potential benefit of the developed BPMN extension for medical workflow systems based on the integration of BPMN and BPEL. Another interesting aspect is the integration of medical and clinical resource objects within BPMN and the design of a holistic, multi-perspective hospital modelling language.

REFERENCES

[1] Antoniou, G., Van Harmelen, F.: Web ontology language: OWL. In: Handbook on ontologies, Springer, 67-92 (2004)

[2] Becker, J.: Interview with Reinhard Schütte on "Managing Large-Scale BPM Projects". Business & Information Systems Engineering, 6(3), 1-3 (2014)

[3] Braun, R., Schlieter, H., Burwitz, M., Esswein, W.: BPMN4CP: Design and Implementation of a BPMN Extension for Clinical Pathway. Unpublished (2014)

[4] Burwitz, M., Schlieter, H., Esswein, W.: Modelling Clinical Pathways - Design and Application of a Domain-Specific Modelling Language. Wirtschaftsinformatik Proceedings 2013, Paper 83 (2013)

[5] Castro, J., Kolp, M., and Mylopoulos, J.: Towards requirements-driven information systems engineering: the Tropos project. Information Systems 27(6), 365-389 (2002)

[6] Frank, U.: Conceptual modelling as the core of the information systems discipline - perspectives and epistemological challenges. AMCIS Proceedings (1999)

[7] Gehlert, A.: Migration fachkonzeptueller Modelle. Logos-Verlag (2007)

[8] Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modelling techniques. Requirements Engineering, 9(4), 248-260 (2004)

[9] Heaven, W., Finkelstein, A.: UML profile to support requirements engineering with KAOS. IEE Proceedings-Software, 151(1), 10-27 (2004)

[10] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly, 28(1), 75-105 (2004)

[11] Kashyap, V., Sheth, A.: Semantic and schematic similarities between database objects: a context-based approach. International Journal on Very Large Data Bases, 5(4), 276-304 (1996)

[12] Object Management Group (OMG): Business Process Model and Notation (BPMN), Version 2.0. http://www.omg.org/spec/BPMN/2.0/PDF/ (2011)

[13] Object Management Group (OMG): Meta Object Facility (MOF), Version 2.4.2. http://www.omg.org/spec/MOF/ (2014)

[14] Object Management Group (OMG): Unified Modelling Language Specification Version 2.4.1.

http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF (2011)

[15] Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E. J.: Memorandum on design-oriented information systems research. European Journal of Information Systems, 20, 7-10 (2011)

[16] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Hall, R., Johnson, P.D.: Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. Journal of the American Medical Informatics Association 10(1), 52-68 (2003)

[17] Pohl, K.: Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated (2010)

[18] Quartel, D., Engelsman, W., Jonkers, H., van Sinderen, M.: A goal-oriented requirements modelling language for enterprise architecture, IEEE International Enterprise Distributed Object Computing Conference (EDOC 2009), 3-13 (2009)

[19] Recker, J.: Opportunities and constraints: the current struggle with BPMN. Business Process Management Journal, 16(1), 181-201 (2010)

[20] Saeedi, K., Zhao, L., Sampaio, P.R.: Extending BPMN for Supporting Customer-Facing Service Quality Requirements. Proceedings of the 2010 IEEE International Conference on Web Services, 616-623 (2010)

[21] Sarshar, K., Loos, P.: Klassifikation von Sprachen zur Modellierung medizinischer Behandlungspfade. In: Rebstock, M. (ed.) Modellierung betrieblicher Informationssysteme - MobIS 2004, 43-59 (2004)

[22] Sarshar, K.P., Dominitzki, P., Loos, P.: Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne - Eine Empirische Methodenevaluation. EPK 2005, 97-116 (2005)

[23] Schlieter, H.: Ableitung von Klinischen Pfaden aus Medizinischen Leitlinien - Ein Modellbasierter Ansatz. Dissertation, TU Dresden (2012)

[24] Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extending BPMN 2.0: Method and Tool Support. Business Process Model and Notation, 59-73 (2011)

[25] Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models. XIV Congreso Iberoamericano en Software Engineering (2011)

[26] Van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. Fifth IEEE International Symposium on Requirements Engineering, 249-262 (2001)

[27] Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Journal of Information Systems, 3(4), 217-237 (1993)

[28] Wand, Y., Weber, R.: Research Commentary: Information Systems and Conceptual Modelling - A Research Agenda. Information Systems Research, 13(4), 363-376 (2002)

[29] Yu, E. S.: Towards modelling and reasoning support for early-phase requirements engineering. Proceedings of the Third IEEE International Symposium on Requirements Engineering, 226-235 (1997)