# Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare

Richard Braun[1], Hannes Schlieter[1], Martin Burwitz[1], Werner Esswein[1]

[1] TU Dresden, Chair of Wirtschaftsinformatik, esp. Systems Development
{richard.braun,hannes.schlieter,martin.burwitz,werner.esswein}
@tu-dresden.de

**Abstract.** It is often required to provide a modeling language that enables the representation of domain-specific problems and concepts. Domain-specific modeling approaches can be applied for that. However, these approaches usually suffer from low dissemination, missing tool support and high design costs. Thus, it might be more reasonable to adapt and extend common standard modeling languages. This research article presents an extension of the common process modeling language BPMN for modeling clinical pathways in the healthcare sector. The extension is designed methodically by application of the extension design method of Stroppi et al. (2011), which was extended regarding to a deeper domain analysis. The domain analysis considers the design of a domain ontology, requirements analysis as well as an equivalence check between domain concept and BPMN concepts. Finally, the evolved extension is compared with the CPmod modeling language of Burwitz et al. (2013) in order to discuss strengths and limitations.

**Keywords:** Process Modeling, Language Extension, Meta Model Extension, BPMN, Domain-Specific Modeling Language

## 1 Introduction and Motivation

Clinical Pathways (CPs) have evolved as important tools in the fields of clinical process management, quality management documentation, and derivation of IT configurations [28, 19, 5]. However, building CPs is a very time-consuming and costly task, where different stakeholders such as physicians, nurses and management employees have to be involved [7]. Up to now, process modeling in the healthcare sector is widely performed by physicians themselves or supported by quality managers using general purpose modeling languages that do not completely fit the needs of the clinical domain or by specific domain-specific modeling languages (DSMLs), which limit the model usage to the tool chain of the DSML-world.

In opposite to dedicated DSMLs, it seems to be reasonable to adapt existing standard languages for healthcare processes and extend them domain-specifically, which combines the advantage of standard languages and domain-specific languages. Using (de-facto) standard modeling languages like BPMN or UML implies several benefits: Generally, these languages provide a set of established concepts with a clear syntax

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

Braun, R.; Schlieter, H.; Burwitz, M.; Esswein, W. (2015): Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare, in: Thomas. O.; Teuteberg, F. (Hrsg.): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015), Osnabrück, S. 468-481

and widely accepted semantics, what facilitates communication between stakeholders. Consequently, also tool support and capabilities for model interchange are better as in case of dedicated DSML development. So far, modeling approaches in the context of CPs mostly draw on DSMLs rather than applying standard modeling languages. This impedes the comparison of both approaches. We therefore aim to design a CP extension for the leading process modeling notation BPMN. The extension should be built based on a systematic design method in order to ensure rigor [16]. Beyond, this research article aims to compare the derived BPMN extension with a dedicated DSML approach, namely [5], in order to provide respective benefits and limitations.

Accordingly, the structure of the paper is as follows: Section 2 introduces a brief overview of modeling language extensibility and the BPMN extension mechanism. Also, the detailed extension method is presented. In the beginning of Section 3, the status quo of modeling clinical pathways is summarized. In the further course, Section 3 provides the BPMN extension design; particularly domain analysis as well as abstract and concrete syntax. Section 4 compares our results to a DSML-based approach. The paper ends with a short conclusion and outlook on further research.

## 2 Extensibility of BPMN

### 2.1 Extensibility of Modeling Languages

As stated before, the broad usage of common modeling languages often causes situations where these languages need to be adapted or extended in order to satisfy requirements and needs coming from the peculiarities of a specific industry, business or problem [4, 27]. Next to this argument of domain-specific adaptation, it might be also reasonable, to focus on a few modeling languages in order to avoid an increasing method pluralism, focusing on their enhancement and thus facilitate better communication both in academia and practice (referring [20, 3]). This strategy also seems to be appropriate in terms of model exchange and interoperability, although noticeable research on that has to be conducted [1]. Also, the stated extension approach could facilitate language evolution over time, when often-used extensions are integrated into the language or some language dialects are standardized (see choreographies in BPMN [8, 6]). Extending a modeling language evokes alteration of the underlying meta model and can affect both abstract and concrete syntax. Currently, only very few research articles addressing this topic explicitly (see [1, 4, 35]). Nevertheless, it can be stated that there are at least four general extension mechanisms: In-built, metamodel customization, annotation and multi level modeling [1]. The in-built mechanism refers to the profile mechanism from UML and similar approaches [26]. Meta model customization refers to any (ad hoc) alteration of the meta model. Model annotation covers adding extra information or an integration with external models. Multi level extensions intend to add new vertical abstraction layers for domain concepts [2]. However, language extensibility also requires an adaption of the language method at all what refers to the procedure model of the language (e.g., [11]), which is not considered in literature so far. Thus, meta model extensibility as well as procedural aspects of extending BPMN are considered subsequently.

### 2.2 BPMN Extension Mechanism and its Shortcomings

BPMN provides an extension mechanism that is similar to the stated profiles mechanism. The mechanism enables the integration of domain-specific concepts while ensuring BPMN core validity ([24], p. 44) and consists of mainly four classes: An *Extension Definition* specifies a named group of new attributes, that can be used by standard BPMN elements. Thus, both new concepts and new additional attributes can be defined. Consequently, an Extension Definition consists of multiple *Extension Attribute Definitions* that define the particular attributes. Permitted values of them need to be defined by the *Extension Attribute Value* class by leveraging basic types from the Meta Object Facility [25]. Finally, the *Extension* class binds the entire extension definition and its attributes to a BPMN model definition in order to make them technically accessible ([24], p. 58). Despite the fact that BPMN offers a well-defined extension interface, only very few BPMN extensions make use of it [4], what hampers comprehensibility, comparability between developed extensions and impedes the straightforward integration of extensions in modeling tools. We suppose, that the missing procedure model for extension building in BPMN causes this lack of rigor.

### 2.3 BPMN Extension Approach of Stroppi et al. (2011) and its Enhancement

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 integrated development of valid BPMN extensions that consist of the following steps:

1. Conceptualizing the domain as an UML class diagram (CDME model)
2. Transforming the CDME into an extension model (BPMN+X model)
3. Transforming the BPMN+X into a XML Schema Extension Definition and transforming that into a XML Schema Extension Definition Document

The approach is based on UML profiles ([36], p. 7) and respective stereotypes in order to enable model transformation based on set of migration rules for diverse model element constellations ([36], p. 9). Although the approach of [36] is very well defined and applicable, it lacks in terms of considering a deeper domain analysis and design preparation. Thus, real need for extension is reasoned by an in-depth analysis of the BPMN specification. This leads implicitly to the derivation of the CDME model and its stereotypes. The following rules are defined for this equivalence check:

*Equivalence*: There is a semantically equivalent construct in the BPMN in the sense of a permitted combination of elements or just a single element. In this case, no extension is necessary and the domain concept is represented as BPMN concept.

*Conditional equivalence*: There is no obvious semantic matching with standard elements, but rather situational discussion is necessary in order to provide arguments for a possible mapping or to explain why it is not feasible. This situation is caused by the partial under specification of BPMN elements (e.g. [24], p. 306). Consequently, the concept is either treated as equivalent concept or as non-equivalent concept.

*No equivalence*: There is no equivalence to any standard element for three reasons: First, the entire concept is missing. In this case, the domain concept is represented as Extension Concept in the CDME model. Second, a relation between two concepts is missing. Therefore, an association between the affected concepts is constructed in the CDME model. Third, properties of a concept are missing. Then, an owned property is assigned to the element in the CDME model.

We argue that an extension should widely make use of standard elements in order to exhaust the vocabulary of BPMN and reduce new elements to a minimum. With respect to the objective of this paper, step 3 and step 4 of the approach of Stroppi et al. (2011) are neither applied nor considered. Instead, the concrete syntax of the extension will be defined in a final step within the development process (according to [24], p. 8). Figure 1 depicts the consolidated procedure model that is applied in Section 3.

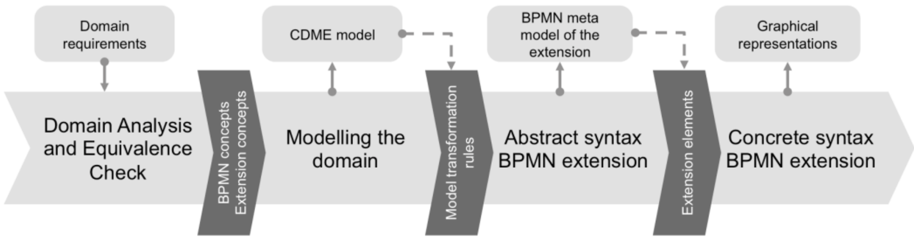


**Fig. 1.** Integrated procedure model for the development of BPMN extensions

## 3 Design of the Extension "BPMN4CP"

### 3.1 Modeling of Clinical Pathways – Status Quo

The current literature provides several contributions in the field of using general purpose modeling languages (GPML) [18, 31], extensions of GPML [22, 32, 33] as well as DSML [5, 12, 34] for modeling medical treatment processes. The review of the state of the art however reveals that only some authors clarify their decision for using a specific language. In particular, there are only a few publications in the BPMN area, which use the BPMN as core language and systematically extend it by domain specific aspects. Müller & Rogge-Solti (2011), for example, focus on the modeling of shared execution of tasks [22]. In the domain-specific area of treatment processes, Scheuerlein et al. (2012) conduct a case study to develop two treatment models with BPMN focusing the applicability of the tangible BPMN construction method [33]. To enable treatment models to be executed and simulated, Hashemian & Abidi (2012) developed a semantic interoperability framework to provide a mapping of domain concepts and BPMN concepts [14].

Each related work is focusing specific modeling questions, while there is no approach using or extending the BPMN covering the typical needs of treatment processes by an in-depth requirements analysis of treatment scenarios like Burwitz et al. (2013) or Sarshar et al. (2005) did [5, 32]. Burwitz et al. (2013) consolidate the main stream of the previously work and design a requirements set for developing modeling languages for CPs. On this basis, they developed a DSML for CP (called "CPmod"). Because of the well-structured requirements set and the possibility to compare the design effort of this DSML against an extended GPML (BPMN), we decide to use this prior research result in Section 3.2 and Section 3.3.

### 3.2 Domain Analysis

Figure 2 depicts the ontology, which was designed in order to get an appropriate understanding of the domain, its concepts and attributes (referring [13, 15]).

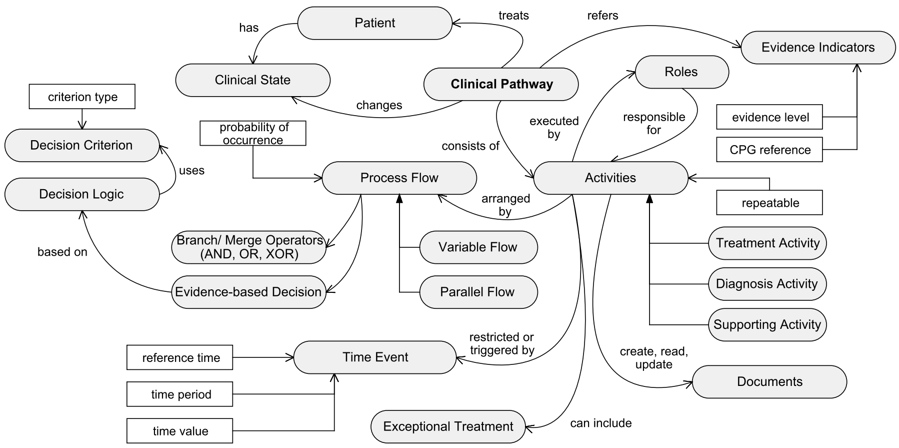


**Fig. 2.** Basic OWL Lite ontology [37] of CPs for the preparation of the extension design

The central concept is the *Clinical Pathway*, which consists of several activities. A *Process Flow* organizes and arranges all activities within the pathway that are necessary for the appropriate treatment of the patient. During treatment, the Patient has several *Clinical States* such as "diagnosed" or "treated". The concept *Variable Flow* covers the circumstance, that some parts of a treatment process are not exactly definable in advance. For example, some containing treatments steps are optional or the sequence of single steps is not determined but case specific. Also, single (sub) parts of one treatment step might be executed by different roles (e.g., different specialists). The concept *Parallel Flow* covers the parallel and simultaneous execution of activities. Simultaneity addresses the execution of parallel tasks at the same time. These tasks must not be executed successively. Further, a process flow contains several basic decision points (AND, OR, XOR) and Evidence-based decision points. Evidence-based decisions are predicated on evidence criteria that are processed within a specific *Decision Logic* (see [29, 30]). *Evidence Indicators* can be assigned to different elements in order to explain their level of evidence and to provide a reference to a clinical pathway guideline (CPG). Activities can be divided into *Treatment Activities*, *Diagnosis Activities* and *Supporting Activities*. Besides, possible complications during treatment need to be taken into account: It is often necessary to define exceptional procedures for common complications that can be described in advance. This occurrence probability of process parts can be described, if appropriate evidences exist. Activities are executed by a specific *Role* that is responsible for the correct execution of an activity. Also, *Documents* for both medical and administrative purposes can be created or updated by activities (e.g., patient files). In the context of CPs, the consideration of time related aspects is crucial (e.g., [21]). Single activities of a treatment process are often restricted or triggered by *Time Events*. These events can be associated with reference times, specific periods or other time-wise values.

### 3.3 Requirements and Equivalence Check

Based on both the stated domain concepts and requirements, the comparison with standard BPMN is conducted in order to identify a reasonable need for extension. According to the presented domain ontology, each concept is described briefly and examined regarding its semantically equivalence with standard elements. Therefore, the respective element descriptions, rules and explanations within the BPMN specification were analyzed in-depth. Table 1 provides the conducted equivalence check and its implications for the CMDE model (classification as *BPMN Concept* or *Extension Concept*). For reasons of clarity, the comparison is divided into a set of generic process concepts and a set of CP specific concepts like evidence indicators. As result of the equivalence check, a classification as is made for each concepts.

### 3.4 Abstract Syntax: CDME and BPMN+X Model

The CDME model was designed with regard to the stated domain ontology and the equivalence check (see Figure 3; standard BPMN classes are colored gray). Regarding to the single extension classes: A *Task* is specified by a *Diagnosis Task*, a *Therapy Task* and a *Supporting Task*. The *Simultan Parallel Gateway* specifies the *Parallel Gateway* in terms of the concurrent execution of activities. The *Evidence-based Gateway* specifies the *Complex Gateway* by defining a new *Expression* type in order to represent complex *Evidence-based Expressions*. These new expressions are composed of and at least one *Decision Criterion* and an optional *Decision Logic* that could specify individual logical expressions or formulas. A *Decision Criterion* has a *Criterion Type* that can be instantiated as *Boolean Criterion Type* or *Interval Criterion Type*. Each *Interval Criterion Type* has a specific range that is explicated by the minimum and maximum properties. Further, several BPMN elements can be associated with *Evidence Indicators*. Both the *Evidence Level* property and the *Document Type* property are designed as enumeration elements giving a domain-specific set of values (e.g., patient files, results from laboratory tests or treatment contracts). The stated properties of the *CPG Reference* element can be used for a detailed description

**Table 1.** Equivalence Check and Derivation of Concepts for the CDME Model

| Req. | Concept | Semantics | Equivalence Check | CDME |
|---|---|---|---|---|
| | | | **Generic concepts from the CP domain and its representation within BPMN** | |
| R1 | **Activity** | Process step within an entire process. | **Equivalence** → Task, Activity | BPMN Concept |
| R1 | **Branch and Merge Operators** | Common decision points within the process flow (XOR, AND, OR). | **Equivalence** → Exclusive Gateway (XOR), Parallel Gateway (AND), Inclusive Gateway (OR) | BPMN Concept |
| R1 | **Process Flow** | Directed control flow between tasks or activities. | **Equivalence** → Sequence Flow | BPMN Concept |
| R1 | **Iteration** | Repeatability of a process part, containing a set of activities. Exit conditions need to be defined. | **Equivalence** → Loop Marker for iterations; Conditional Events for exit conditions | BPMN Concept |
| R1 | **Responsibility (Role)** | Definition of execution responsibilities. | **Equivalence** → Pools, Lanes | BPMN Concept |
| R2 | **Variable Flow** | Some parts of treatment processes are not exactly definable in advance. For example, the execution of some treatment steps is variable or single treatment steps are optional. Also, there can be conditions regarding the number of executable activities ("0..n", "1..n", at least two etc.). Besides, the execution of activity bundles by different roles should be possible. | (a) Variable execution: **Equivalence** → Ad Hoc Processes: Ad Hoc Processes contain a set of tasks having no pre-defined sequence. Both sequence and occurrence are determined by the performer of the process ([24], p. 180-182).<br>(b) Role-across execution: **Conditional equivalence** → Pools, Lanes: Pools are representations of participants and reflect responsibilities for an entire process. Lanes serve the organization of activities ([24], p. 306 et seq.). Lanes can be used for roles, systems or departments. Below, Lanes are used for the representation of roles and Pools are used for the entire CP. Further, Groups can be used for the visualization of logically contiguous activities over different roles.<br>(c) Variable decision points: **Equivalence** → Gateways: Gateways offer the opportunity to define very different decision logics. If necessary, Complex Gateways and Annotations can be used for the explication of nontrivial logics. | BPMN Concept |
| R2 | **Parallel Flow** | Parallel and simultaneous execution of activities. | **Conditional equivalence** → Parallel Gateway: Parallel Gateways only define the synchronization of parallel sequence flows. Simultaneity regarding the particular tokens is not required ([24], p. 293; p. 434). Thus, Parallel Gateways cannot be used. Further, a combination of Parallel Gateways and Conditional Events is possible. However, the semantics of this concept should be realized within one element with respect to model clarity. | Extension Concept (specification of parallel gateway concerning simultaneity) |
| R4 | **Time Events** | Temporal aspects, dependencies and restrictions (reference time, time period, specific time value) | **Equivalence** → Timer Events | BPMN Concept |
| | | | **Specific concepts from the CP domain and its representation within BPMN** | |
| R1 | **Treatment Steps (Diagnosis, Therapy, Supporting)** | Specification of treatment steps as diagnosis task, therapy task or supporting task. | **No equivalence** (no appropriate Marker) | Extension Concept |
| R1 | **Documents** | Specification of documents for medical or administrative purposes. | **Conditional equivalence** → Data Objects: The use of Data Objects is possible but would cause a high level of abstraction. Also, the use of Properties is not possible since only Processes, Activities and Events can be associated with them ([24], p. 210). | Extension Concept |
| R1 | **Clinical States** | Representation of the patient's condition based on events, e.g. a start event represents the initial condition of a patient, a terminal event represents the state of the patient after treatment. | **Equivalence** → Events | BPMN Concept |
| R3 | **Evidence-based Decision** | Complex path decisions based on at least one decision criterion. | **Conditional equivalence** → Complex Gateway: The use of a Complex Gateway and an associated Annotation containing the activation Condition is imaginable ([24], p. 84 and p. 295). However, this configuration lacks both in terms of formalization and the integration of evidence-based criteria. | Extension Concept |
| R3 | **Decision Criterion** | Evidence based decision criterion having defined limit values. | **No equivalence** | Extension Concept |
| R3 | **Criterion Type** | The type of a decision criterion. | **No equivalence** | Extension Concept |
| R3 | **Decision Logic** | Evidence based decision logic that contains decision criteria and their application. | **No equivalence** | Extension Concept |
| R3 | **Evidence Indicator** | Indicator that can be bound to activities, decisions, process parts and the entire process. | **No equivalence** (a simple visual grouping of elements with Groups ([24], p. 68) is not preferred) | Extension Concept |
| R3 | **Evidence Level** | Maturity and quality of an evident statement (levels A - F). | **No equivalence** | Extension Concept |
| R3 | **CPG Reference** | Reference to a specific clinical practice guideline. | **No equivalence** | Extension Concept |
| R5 | **Exceptional Treatment** | Treatment of complications or other kinds of irregularities (both predictable and unpredictable) | **Equivalence** → Cancel Events (e.g., cancellation of a therapy), Error Events (e.g., unexpected complications), Escalation Events (report to the next highest responsible role), Compensation Events (treatment of predictable irregularities) | BPMN Concept |

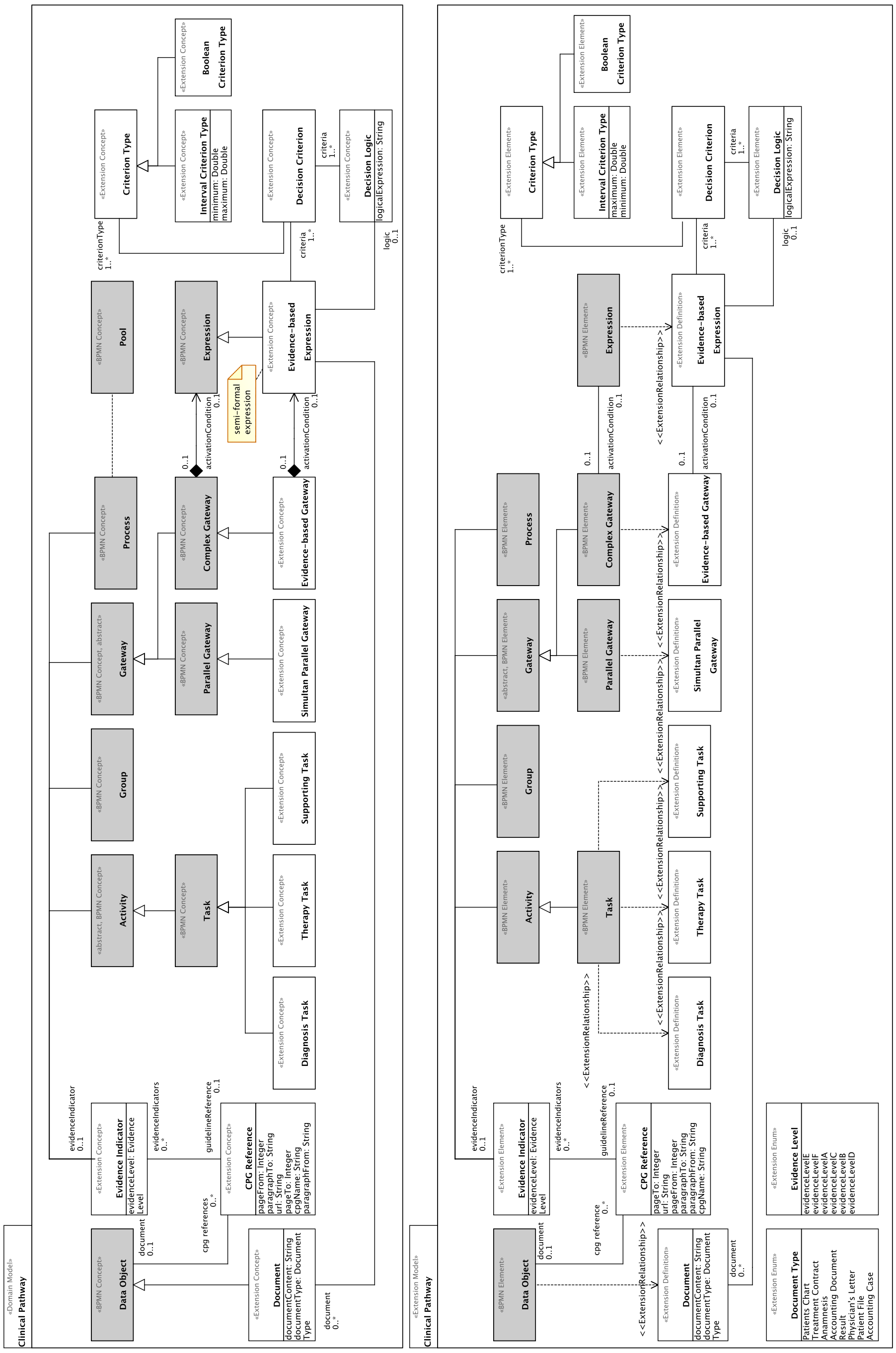


**Fig. 3.** Abstract syntax of the BPMN4CP extension (CDME and derived BPMN+X model)

of the respective *CPG document* in order to reference to the relevant statements.

Based on the model transformation rules stated in Stroppi et al. (2011), the CDME model was then transformed into the BPMN+X model (Figure 3). With respect to the limited space of this paper, the application of each applied transformation rule cannot be presented.

### 3.5 Concrete Syntax and Demonstration

The concrete syntax of the extension is presented in Figure 4. The *Task* element is specified by new markers for representing *Diagnosis Tasks*, *Therapy Tasks* and *Support Tasks*. A Document is depicted as specified *Data Object* with an icon on the left side that reflects the selected *Document Type* of the element. Further, the graphical representations of the *Simultan Parallel Gateway* and the *Evidence-based Gateway* are provided. An *Evidence Indicator* is depicted as orange hand containing the respective level of evidence. Following the determinations in the extension meta-model, the icon can be assigned to nearly all flow elements.


**Fig. 4.** New notations: New task marker, specified data object (icon represents the document type), *Simultan Parallel Gateway*, *Evidence-based Gateway* and *Evidence Level* (left to right)

Figure 5 demonstrates the evolved BPMN extension by presenting a simplified CPG of a wisdom tooth treatment whose semantic is as follows: The treatment starts with the diagnostical activities of anamnesis and radiographic test. Since the order of these steps is not explicitly determined while both activities are obligatory, we use parallel gateways. The surrounding group additionally visualizes the semantic integrity of both steps being diagnostics. The radiograph created is necessary for the later decision of therapy. Computer tomography and biopsy are additional but optional diagnostic steps, expressed using inclusive gateways. All these actions are typed diagnostic, illustrated by the corresponding icon.

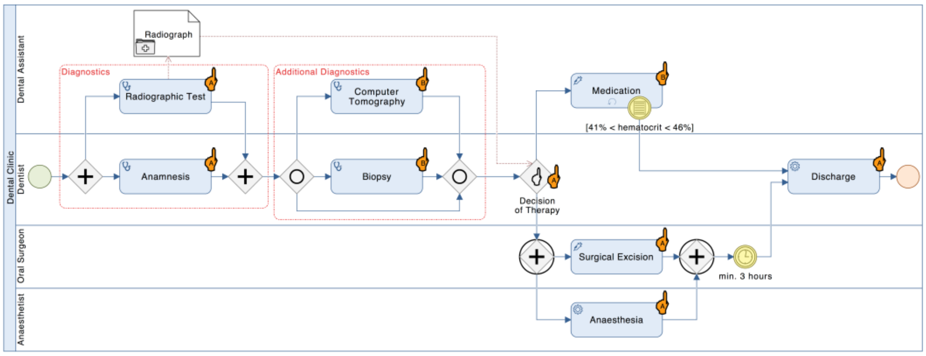


**Fig. 5.** Instance of BPMN4CP demonstrating a wisdom tooth treatment process

The dentist is responsible for the evidence-based decision of therapy. The decision is associated with several decision criteria (e.g., radix aberration [true/false], carious defect [true/false]) and decision logic, expressing that if a patient has no radix aberration but a carious defect, the alternative of a surgery is usually preferred. Otherwise, an iterative medication should be done, until the patient's hematocrit reaches an acceptable range, before being discharged finally. Since the dentist decides for a surgical excision, this activity is simultaneously supported by an anesthesia. In this case, the patient cannot be discharged before three hours after surgery. Evidence indicators are attached to various process elements expressing their evidence level. The radiographic test and the anamnesis e.g. are highly approved by the current medical science (A-level).

# 4 Discussion und Further Research

## 4.1 BPMN Extension vs. DSML Approach

As stated at the beginning of this article, it was also intended to conduct a comparative analysis of the designed BPMN extension and a dedicated DSML approach, namely Burwitz et al. (2013). The analysis is feasible, since both approaches mainly repose on the same requirements (see Section 3.1). Therefore, an argumentative consideration of advantages and disadvantages was conducted (see Table 2).

**Table 2.** Comparison between extension and DSML approach (legend: ● fulfilled, ⊙ partly, − not fulfilled)

| Criterion | Extension Approach | | DSML Approach (see [5]) | |
|---|---|---|---|---|
| *Domain Representation* | Nearly complete<br>* Discussion on conditional equivalences necessary (see Table 1) | ● (*) | Complete | ● |
| *Meta Model* | Level M2 / $M2_{profile}$ | ● | Level M2 | ● |
| *Abstr. Syntax* | Limited capabilities (missing element constraints), no views possible | ⊙ | Complete specification<br>Views are possible | ● |
| *Concr. Syntax* | Limited by BPMN style ([24], p. 8) | ⊙ | Not limited | ● |
| *Procedure* | Limited guidance [1, 36] | ⊙ | Limited guidance [10] | ⊙ |
| *Concept Reuse* | Broad reuse (e.g., task type)<br>Focus on domain concepts | ● | Re-design of basal process concepts<br>Focus on domain and process concepts | − |
| *Cost of Design* | Depends on relations to BPMN | ⊙ | Depends on language | ⊙ |
| *Tool Support* | Good (range of tools) | ● | Limited | − |
| *Dissemination* | Very good | ● | Limited | − |
| *Integration* | Integration with other extensions<br>Reuse of other BPMN interfaces | ● | Dedicated interfaces required<br>No integration reuse | − |
| *Execution* | BPEL integration with adaption | ⊙ | Requires dedicated design | − |

As can be seen from Table 2, DSMLs generally benefit from their capabilities in describing the domain adequately, since both syntax and semantics can be defined completely in accordance to the problem. Of course, the extension approach suffers from some shortcomings in this context due to limitations of the extension mechanism and its capabilities in defining new concepts. For instance, BPMN does not provide a specification of the BPMN class that is extended in particular. Rather, each sub class of the *Base Element* class can have a relation to some extension class. It has also to be taken into account, that language extension requires both a very deep understanding of all language concepts and a reasonable consideration of semantic equivalences, what can be very time-consuming (see Section 3.3). On the other side, language extension benefits from the reuse of general concepts, constraints and rules. A dedicated definition of – for instance – basic process logic concepts is not necessary.

However, it cannot be stated in general, that designing a DSML is more expensive than designing an extension, since some appropriate tools and frameworks exist for that (e.g., [9]). In regard to modeling tool support and the integration of further (process) extensions, the extension approach has advantages due to the dissemination of BPMN and the existence of interfaces, tools and also knowledge. In DSML approach, corresponding interfaces need to be implemented separately. The ability of BPEL based workflow integration is another plus of the BPMN extension approach, which supports model execution.

In the context of CPs, we found that the BPMN extension approach is more suitable in terms of language design, since a range of BPMN tools could be reused or specialized (see Table 1). Further elements were added as kind of information objects with specific properties. However, a "make or extend" decision always depends on both the properties of the considered domain and the entire project scope. For instance, CPs are common in healthcare and hospitals perhaps even apply BPMN for CP modeling. Thus, BPMN seems to be better for reasons of scalability. In contrast, a DSML can be more powerful for very specific CP modeling cases that imply special requirements to the control flow, the message flow or model transformations.

### 4.2 Conclusion and Outlook

This research article provides one major contribution as well as two related contributions: First, a valid BPMN extension for CPs is provided, which can be applied by domain experts or even customized by model engineers due to its comprehensive and precise specification. Second, we propose an enhancement of the BPMN extension approach of Stroppi et al. (2011) in order to conduct a deeper domain analysis with better consideration of comparing extension concepts with standard BPMN concepts. Third, we also introduced a comparison of the CP extension approach with a DSML based approach in order to state respective advantages and disadvantages.

There are some issues for investigation in further research: First of all, it seems to be necessary to intensify research on decision support for language engineers in terms of extending a standard language or building a dedicated DSML from scratch. There should be also better methodical support of extension development; both regarding syntactical aspects (e.g., extension patterns) and also regarding to better guidance in development. It is also necessary to intensify research on the "extend or build" decision (see above), which depends on domain characteristics and the particular project context. With regard to BPMN4CP it has to be stated, that the extension has to be evaluated in a broader application space [16]. Currently, first prototypical application is done in the EU founded healthcare infrastructure project *"CCS Telehealth East*

*Saxony"*, in which the extension is used for building CPs that are used in the context of education, quality management certification and configuration of workflows for the case management tasks.

Generally, there are also several aspects for further investigation in the field of conceptual modeling in healthcare: Industrial processes can be designed very strictly during process design-time since there is relatively low risk of unexpected events. In contrast, medical processes have a high potential of deviation while process run-time due to numerous reasons such as individual choices of patients, experiences of physicians or very untypical case complexes. Only few research addresses this topic so far. Concerning that, the recently published Case Management Model and Notation (CMMN) is very promising, as CMMN facilitates modeling ad-hoc reactions to changing, case-depended conditions [23]. Within this research article, we focus on BPMN due its dissemination in the healthcare sector and its benefits (see Section 1).

Finally, we have also identified problems in modeling role across (shared) activities in BPMN (see also [22]). Indeed, it is possible to model a shared set of activities by using *Lanes* and *Groups* in one diagram, but this configuration does not allow to model iterations within the grouped set of activities. Separating the shared activity into several diagrams is possible at this point, but impedes model readability.

## References


1. Atkinson, C., Gerbig, R., Fritzsche, M.: Modeling language extension in the enterprise systems domain. IEEE International EDOC Conference, 49-58 (2013)
2. Atkinson, C., Gutheil, M., Kennel, B.: A flexible infrastructure for multilevel language engineering. IEEE Transactions on Software Engineering, 35(6), 742-755 (2009)
3. Becker, J.: Interview with Reinhard Schütte on "Managing Large-scale BPM Projects". Business & Information Systems Engineering, 1-3 (2014).
4. Braun, R., Esswein, W.: Classification of Domain-Specific BPMN Extensions. In: Frank, U., Pastor, O., Loucopoulos, P., Petrounias, I. (Eds.) Lecture Notes in Business Information Processing, 197, 42-57 (2014)
5. Burwitz, M., Schlieter, H., Esswein, W.: Modeling Clinical Pathways-Design and Application of a Domain-Specific Modeling Language. Wirtschaftsinformatik Proceedings, Paper 83 (2013)
6. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard. Computer Standards & Interfaces, 34(1), 124-134 (2012)
7. De Bleser, L., Depreitere, R., De Waele, K., Vanhaecht, K., Vlayen, J., Sermeus, W.: Defining Pathways. Journal of Nursing Management 14(7), 553-563 (2006)
8. Decker, G., Kopp, O., Leymann, F., and Weske, M. BPEL4Chor: Extending BPEL for modeling choreographies. IEEE Internat. Conference on Web Services, 296–303 (2007)
9. Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) Conceptual Framework and Modeling Languages. HICCS Proceedings, 1258-1267 (2002)
10. Frank, U.: Some Guidelines for the Conception of Domain-Specific Modeling Languages. EMISA Proceedings, 190, 93-106 (2011)
11. Greiffenberg, S. (2004). Methodenentwicklung in Wirtschaft und Verwaltung. Kovac.
12. Heise, D., Hess, M., Strecker, S., Frank, U.: Rekonstruktion eines klinischen Behandlungspfads mithilfe domänenspezifischer Erweiterungen einer Geschäftsprozessmodellierungssprache. DLM 2010, 210-227 (2010)

13. Happel, H. J., Seedorf, S.: Applications of ontologies in software engineering: Proceedings of Workshop on Sematic Web Enabled Software Engineering on the ISWC, 5-9 (2006)
14. Hashemian, N., Abidi, S. S. R.: Modeling clinical workflows using business process modeling notation. International Symposium on. Computer-Based Medical Systems (2012)
15. Heijst, G., Schreiber, A.T., Wielinga, B.J.: Using Explicit Ontologies in KBS Development. International Journal of Human-Computer Studies, 46(2), 183-292 (1997)
16. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly, 28(1), 75-105 (2004)
17. ISO (International Organization for Standardization): ISO/IEC 19510:2013, Object Management Group Business Process Model and Notation (2013)
18. Kirchner, K. et al.: Experience from collaborative modeling of clinical pathways. Tagungsband des Workshops Modellierung im Gesundheitswesen, 13-22 (2014)
19. Lenz, R., Reichert, M.: IT Support for Healthcare Processes–premises, Challenges, Perspectives. Data and Knowledge Engineering, 61(1), 39-58 (2007)
20. Loos, P., Mettler, T., Winter, R., Goeken, M., Frank, U., Winter, A.: Methodological pluralism in business and information systems engineering? Business & Information Systems Engineering, 5(6), 453-460 (2013).
21. Lin, F. R., Chou, S. C., Pan, S. M., Chen, Y. M.: Mining time dependency patterns in clinical pathways. International Journal of Medical Informatics, 62(1), 11-25 (2001)
22. Müller, R., Rogge-Solti, A.: BPMN for Healthcare Processes. Third Central European Workshop on Services and their Composition, 65-72 (2011)
23. Object Management Group (OMG): Case Management Model and Notation Version 1.0. http://www.omg.org/spec/CMMN/1.0 (2014)
24. Object Management Group (OMG): Business Process Model and Notation (BPMN), Version 2.0. http://www.omg.org/spec/BPMN/2.0/PDF/ (2011)
25. Object Management Group (OMG): Meta Object Facility (MOF), Version 2.4.2. http://www.omg.org/spec/MOF/ (2014)
26. Object Management Group (OMG): Unified Modeling Language Specification Version 2.4.1. http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF (2011)
27. Pardillo, J.: A Systematic Review on the Definition of UML Profiles. Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science (6394), 407-422.
28. Rotter, T., L. Kinsman, E. James, A. Machotta, H. Gothe, J. Willis, P. Snow, J. Kugler: Clinical Pathways: Effects on Professional Practice, Patient Outcomes, Length of Stay and Hospital Costs. Cochrane Database Systems Review,3 (2010)
29. Sackett, D. L., Rosenberg, W., Gray, J. A., Haynes, R. B., Richardson, W. S.: Evidence based medicine: what it is and what it isn't. BMJ, 312(7023), 71-72 (1996).
30. Sackett, D. L.: Evidence-based medicine. John Wiley & Sons, Ltd. (2000)
31. Sarshar, K., Loos, P.: Klassifikation von Sprachen zur Modellierung medizinischer Behandlungspfade. In: Rebstock, M. (ed.) Tagungsband MobIS 2004, 43-59 (2004)
32. 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)
33. Scheuerlein, H., Rauchfuss, F., Dittmar, Y., Molle, R., Lehmann, T., Pienkos, N., Settmacher, U.: New methods for clinical pathways – Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t.BPM). Langenbeck's Archives of Surgery, 397(5), 755-761 (2012)
34. Schlieter, H.: Ableitung von Klinischen Pfaden aus Medizinischen Leitlinien - Ein Modellbasierter Ansatz. Dissertation, Technische Universität Dresden (2012)

35. Selic, B.: A Systematic Approach to Domain-Specific Language Design Using UML. 10th IEEE International Symposium on Object and Component- Oriented Real-Time Distributed Computing, 2-9 (2007)
36. Stroppi, L. J. R., Chiotti, O., Villarreal, P.D.: Extending BPMN 2.0: Method and Tool Support. Business Process Model and Notation, 5973 (2011)
37. W3C: OW+L 2 Web Ontology Language - Document Overview; W3C Recommendation. http://www.w3.org/TR/owl2-overview (2012)