Classification of Domain-Specific BPMN Extensions

Richard Braun and Werner Esswein

Technische Universität Dresden
Chair of Wirtschaftsinformatik, esp. Systems Development
01062 Dresden, Germany
{richard.braun,werner.esswein}@tu-dresden.de

Abstract. BPMN is a standard for modeling business processes and provides meta model concepts for the design of extensions. Thus, domain-specific extensions of the BPMN are facilitated. This research article provides an overview of BPMN extension development by the descriptive analysis and classification of 30 BPMN extensions. An extensive literature review was conducted in order to find published extensions. Further, a classification framework was designed to enable a comprehensive analysis of each extension. The analysis showed, that four out of five extensions are not compliant with the BPMN standard. Also, we found several methodological shortcomings that should be tackled in further research.

Keywords: BPMN Extensions, Domain-specific Extension, Modeling Languages, DSML, DSML Repository, Business Process Modeling.

1 Introduction and Motivation

The Business Process Model and Notation (BPMN) is an ISO standard for modeling business processes and a de-facto standard in professional practice [1], [2]. BPMN provides a set of generic business process elements, independent from a specific domain. However, it is often necessary to extend BPMN with individual concepts in order to represent characteristics of a particular domain (e.g., health care or security management). On the one hand, such domain-specific aspects can be integrated within a dedicated domain-specific modeling language (DSML) [3], [4]. On the other hand, BPMN can be extended with domain-specific concepts in order to reuse the modeling language, take advantage of its benefits (e.g., standardization, tool support) and avoid expensive development of a DSML from the scratch. This research article investigates the current state of the art of BPMN extension development. A BPMN extension is understood as the enhancement of functionality of the BPMN, following the extension mechanism defined in specification. In its own, the standard-conform BPMN extension is neither useful nor functional (referring to [5], [6]).

U. Frank et al. (Eds.): PoEM 2014, LNBIP 197, pp. 42–57, 2014.

1.1 BPMN Extensibility

BPMN is one of very few modeling languages that provides generic extension elements within the meta model that enables the definition of domain-specific language extensions [7]. BPMN provides an extension by addition mechanism that ensures the validity of the BPMN core elements ([8], [7], p. 44). The following elements are defined for the specification 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 the particular attributes. Values of these *Extension Attribute Definitions* can be defined by the *Extension Attribute Value* class. Therefore, primitive types from the Meta Object Facility can be used [7]. 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 ([7], p. 58). Further, external relationships can used for the integration of BPMN artifacts and UML elements, for instance (see [7], p. 62). Despite the fact that BPMN provides a well-defined extension interface, a process model for the straightforward development of extensions is missing. To the best of our knowledge, there is only one research article addressing this problem: [8] defines a model-transformation based procedure model for the methodical development of valid BPMN extensions models based on conceptual domain models. However, the approach lacks in terms of a detailed analysis and consideration of the domain since it is a more engineering driven approach that aims to provide clear transformation rules. Therefore, [9] extends the method with regard to the domain analysis and outline several preceded steps in order to conceptualize the domain and identify a reasoned need for extension.

1.2 Research Objective

As stated above, a detailed process model for the application of these extension elements is missing and the development of an extension remains more or less "ad hoc". Especially from a design science perspective, this lack of rigor is insufficient (e.g., [10]). For example, there is neither guidance in terms of the domain conceptualization nor a semantic analysis between a specific domain concept and BPMN elements. The mentioned approaches address this issue, but either lack in terms of domain analysis [8] or level of detail and applicability [9]. We argue, that it is crucial to evolve a holistic process model for BPMN extension development to ensure standard conformity, comprehensibility and falsifiability. Therefore, it is unavoidable to gain a comprehensive overview of the state of the art in the context of BPMN extensions. Thus, this research article aims to provide a systematic, descriptive analysis of BPMN extensions in order to give indications of both methodological and domain-specific aspects within BPMN extension development.

1.3 Research Method

In order to find published BPMN extensions, a systematic literature review was conducted. We have applied the method of [12] that was configured as follows: The scope of our review was a broad analysis of BPMN extensions. According to [11] the review is conceptual, has a research outcome focus, aims to integrate existing results (to a classification schema), has an exhaustive coverage and addresses a general audience. Second, the topic was conceptualized by the definition of relevant search phrases and keywords (see [12]), such as "BPMN extension", "extend BPMN", "enhance BPMN", "extending BPMN", "domain-specifc BPMN" and "domain BPMN". Third, the literature search process was conducted [12]. Therefore, the journal and conference list of the german research organization *WKWI* was used [13]. Also, literature databases and search engines like *Google Scholar*, *Springer Link*, *Science Direct*, *AIS Digital Library* and the *IEEE Xplore Digital Library* were used. Besides, each found article was used for a backward search. This search procedure resulted in a set of 39 articles, whose content were reviewed. Publications, focusing on early BPMN extensions that are now part of the language (e.g., [14]) or articles that did not provide any conceptual advices on their extension (e.g., [15], [16]), were discarded and a set of 30 articles remain for in-detail analysis that was conducted subsequently. Therefore, a multi-perspective analysis framework has been designed in order to facilitate a comparison of the identified extensions. The systematization of all BPMN extensions and the derivation of the state of the art represent the synthesis of the review process. Finally, research gaps and aspects for further research were derived.

The structure of the article is as follows. Section 2 presents the extension analysis framework containing four main classes and all relevant criteria. In section 3, the results of the literature review are analyzed within the framework. Section 4 provides indications as a result of the classification. The article ends with a short summary.

2 BPMN Extension Analysis Framework

The reasonable analysis of BPMN extensions requires the definition of a description framework. In the context of BPMN, there are no comparable approaches, that could be leveraged for the derivation of such criteria. However, there are few research articles addressing a systematic overview or classification of extensions in the field of the workflow modeling language BPEL [6] and UML profiles [17]. [6] evolves a classification framework for BPEL extensions based on the analysis of 62 publications. Since their work focusses on workflow aspects, the reuse of the entire classification framework is not reasonable. Nevertheless, some criteria like standard conformity, extension purpose and basic characteristics are adapted in the context of BPMN. [17] provides a systematic review of UML profiles based on the analysis of 39 publications. Although the focus lies on the analysis of UML profiles, the consideration of extended meta classes (see [17], p. 413) is promising in the context of BPMN since both modeling languages are

defined by the Meta Object Facility (MOF). Referring to the mentioned works and the research objective of this paper, the following classes for descriptive analysis of BPMN extensions were defined: "Basic attributes", "standard conformity", "applied method" and "extension". Each class, its containing criteria and all classification values are described in the tables subsequently. If necessary, detailed explanations of single criteria are given.

Table 1. Basic Attributes

Criterion	Description	Values
Authors	Authors of the publication	(reference)
Year	Year of publication	2007 - 2014
Version	Affected BPMN version	BPMN 1.x; BPMN 2.0 (since 2011)
Medium	Publication medium	J (journal); P (proceedings); O (others)
Title	Title of the extension	e.g., BPMN4WSN
Domain	Affected domain or area of discourse	e.g., Artifacts or Resources
Purpose	Derived purpose	D (descriptive); A (analytic); E (execution)

The criterion *Domain* describes the affected domain, the application fields or the general area of discourse of the extension. During analysis, similar domains (e.g., Security Management and Risk Management) were merged to single domains (e.g., Risk Management) in order to consolidate them. Criterion *Purpose* stands for the primarily purpose of the extension. An extension was classified as "descriptive" (D) if its focus lies on the description of a domain. It was classified as "analytic" (A) if the main purpose consists in facilitating some kind of analysis of existing BPMN models. If the extension aims to support process execution (e.g., supporting domain-specific transformation to BPEL), the extension was classified as "execution" (E).

Table 2. Standard Conformity

Criterion	Description	Values
Definition	Type of extension definition	Valid Ext; Own Ext; Own Ext Notation; None
Abstract Syntax	Definition of the meta model	e.g., UML, Ext MM (BPMN extension meta model)
Concrete Syntax	Definition of new notations	explicit; implicit (by example); none
Semantic Conflicts	Are there any semantic conflicts with the BPMN standard?	no; yes

The "Standard Conformity" class contains criteria regarding the syntactical and semantic correctness of the extension in the light of the BPMN standard (see section 1). Criterion *Definition* describes the way the extension is defined and explicated. "Valid Ext" stands for the definition as BPMN extension model. "Own Ext" outlines the application of a dedicated definition (e.g., UML model). "Own Ext Notation" stands for a solely graphical definition (e.g., by new icons).

Table 3. Method

Criterion	Description	Values
Requirements Analysis	Is there any analysis or consideration of requirements to the extension?	explicit; implicit; no
Semantic Fit Check	Is there any discussion of the semantic fit of domain concepts with BPMN elements for the identification of extension need?	yes; partly; no
Reuse of Artifacts	Many domains already provide some artifacts such as ontologies. The reuse and integration of them might be useful.	yes; partly; no
Process Model	Is any methodological approach applied (if yes, which one)?	Stroppi et al.; BPMN ext; yes (own); no

Further, the definition of customized or new graphical elements is considered by the criterion *Concrete Syntax*. Also, we have analyzed whether a single extension contains obvious semantic conflicts.

As stated at the beginning of the paper, the methodological development of BPMN extensions is important, but BPMN standard does not provide any guidance and only very few publications addressing this topic. Thus, both methodological and domain-analysis aspects are investigated within the class "Method". For instance, requirements analysis is perceived as essential for the development of artifacts. It might be reasonable to reuse existing domain artifacts for reasons of redundancy and communication with domain experts. Also, a discussion of the semantic fit with BPMN elements is necessary to constitute the need for extension elements.

The class "Extension" describes all extensions and customizations for the integration of domain-specific aspects in BPMN. The first part contains all newly added elements, relations, properties and diagrams. Therefore, first of all it was analyzed whether the extension was defined by a meta model. If not, we have

Table 4. Extension

Criterion	Description	Values
New elements		
Elements	New elements and enumerations (up to three example elements are stated)	(individual)
Count	Number of new elements (if the number is in brackets, a meta model is missing and the elements are derived logically; e.g., [18], [19])	(individual)
Size Class	Derived extension size class, based on the number of extension elements	Heavy (>17); large (11-17); light (6-10); tiny (<6)
Diagrams	Does the extension provide a new diagram?	yes; no
Extended or customized elements		
Relations	Extending a BPMN element by new navigable relations to or from the element	BPMN element(s)
Properties	New owned properties of a BPMN element	BPMN element(s)
Specialization	Adding new sub classes to a BPMN element	BPMN element(s)
Enhancement	Adding a new super class to a BPMN element	BPMN element(s)
Graphical Custom.	Specifying a BPMN element by a new graphical representation (see [7], p. 44)	BPMN element(s)
Count	Number of extended elements	(individual)
Extension Style	Identified extension styles	Codes from table 5

Table 5. Extension Styles

Code	Name	Description
		Abstract Syntax
AS-Sp	Specialization	Specialization of elements by inheriting from the standard element and extending it (e.g., by additional properties).
AS-A	Additive (various)	Set of both new elements and new relations or properties (both optional and mandatory). Thus, the meta model extension is largely integrated within the BPMN meta model.
AS-A-B	Additive (block)	Set of new elements that is related to the BPMN core model by only one or two relationships. Thus, the meta model extension looks like a well definable extension block.
AS-En	Enumeration	Domain-specific ranges in the form of enumeration elements.
		Semantics
Sem-Co	Concretisation	Specification of under specified elements (e.g., Lanes [7]).
Sem-Ch	Change	Dedicated change of some element's semantics, which is not permitted within BPMN.
		Concrete Syntax
CS-Dg	Diagram, view	Adding a new diagram or view to BPMN (e.g., resource diagram as complement of the collaboration diagram).
CS-Cu	Customization	Customization of graphical elements (e.g., data objects).
CS-Co	Color	Color highlighting of elements or parts with special semantics.
CS-Ah	Ad hoc	Elusive definition of an extension by graphical icons, without any abstract syntax.

tried to identify new elements based on explanations in the research article. Even though these explanations were missing, we looked for new, solely graphically defined elements (see *Graphical Custom.*). Criterion *Size Class* is a simple parameter for the number of new elements [1]. Further, the so-called extension styles of an extension were analyzed in order to get a better understanding of the way an intended extension was implemented and expressed. Therefore, ten extension styles were derived from the set of all 30 extensions inductively. Each extension style is assigned to one of the following classes that were adapted from method engineering: Abstract syntax, semantics and concrete syntax. Table 5 presents and describes all styles in detail. Each analyzed extension can have multiple extension styles.

3 BPMN Extension Classification

The conducted literature review resulted in a set of 30 BPMN extensions. Each extension was analyzed with respect to the abovementioned framework. Figure 1 presents the results of the analysis regarding to basic attributes, conformity to the standard and the applied method. Figure 6 presents the results regarding to the syntactical definition of the extensions.

3.1 Basic Extension Attributes

The majority of the considered extensions is related to BPMN version 2.0 (76,6%). Extensions are mainly published in conference proceedings (60,0%) or as research

[1] Size classes were generated by the application of the k-means algorithm over all element counts (k=4; euclidean distance).

Basic Attributes							Standard Conformity				Method			
Authors	Year	Version	Medium	Title	Domain	Purpose	Definition	Abstract Syntax	Concrete Syntax	Semantic Conflicts	Req. Analysis	Semantic Fit Check	Reuse of Artifacts	Process Model
Altuhhov et al.	2013	2.0	J	No title (Security Risk Management)	Risk Management	D	Own Ext Notation	UML	Explicit	No	implicit (ISSRM)	yes	yes (ISSRM)	no
Awad et al.	2009	2.0	O	No title (Ressource Allocation Constraints)	Ressource	E	Own Ext	UML, OCL	No	No	implicit (patterns)	no	no	no
Bocciarelli & D'Ambrogio	2011	1.x	P	PyBPMN	Performance Measurement	A	Own Ext	UML, OCL	No	No	no	no	yes (MARTE)	(yes)
Brambilla et al.	2012	2.0	P	No title (Social BPM)	Social BPM	D	Own Ext	(UML)	By example	No	no	no	no	no
Braun & Esswein	2014	2.0	P	No title (Ressources in Engineering)	Resource	D	Valid Ext	Ext MM	Explicit	No	implicit (ontology)	yes	yes (ResML)	yes: ext. of Stroppi et al.
Braun et al.	2014	2.0	O	BPMN4CP	E-Health	D	Valid Ext	Ext MM	By example	No	explicit	yes	yes (requirements)	yes: ext. of Stroppi et al.
Brucker et al.	2012	2.0	P	SecureBPMN	Risk Management	D	None	No	By example	No	no	no	yes (SecureBPM)	no
Charfi et al.	2010	2.0	J	AO4BPMN	Aspect Modelling	E	None	No	By example	No	no	no	partly (AOP elements)	no
Friedenstab et al.	2012	2.0	P	No title (Business Activity Monitoring)	Performance Measurement	A	Own Ext	UML	Explicit	No	implicit (descriptive)	no	no	no
Gagne & Trudel	2009	1.x	P	Time-BPMN	Time	A	None	No	By example	No	implicit (descriptive)	no	yes (time aspects)	no
Großkopf	2007	2.0	O	No title (Ressource Information Layer)	Resources	E	None	UML	No	No	no	yes	no	no
Kopp et al.	2012	2.0	J	BPMN4TOSCA	Cloud	D	None	No	Explicit	No	explicit	no	yes	no
Korherr & List	2007	1.x	J	No title (Performance Measures)	Performance Measurement	A	Own Ext	UML	By example	No	implicit (descriptive)	no	no	no
Lodhi et al.	2011	2.0	O	No title (Process Evaluation)	Performance Measurement	A	Own Ext Notation	(UML)	By example	(No)	no	no	partly (eval. concepts)	no
Lohmann & Nyolt	2011	2.0	P	No title (Artifact-centric Modeling)	Artifacts	D	None	No	By example	No	no	no	no	no
Magnani & Montesi	2007	1.x	P	No title (Costs)	Performance Measurement	A	None	No	By example	No	no	no	no	no
Magnani & Montesi	2009	1.x	O	BPDMN	Artifacts	A	Own Ext Notation	No	Explicit	Yes	implicit (feature comparison)	no	yes (data flow, ERM)	no
Marcinkowski & Kuciapski	2012	2.0	J	No title (Risk Handling)	Risk Management	D	Own Ext	UML	By example	No	no	no	no	no
Müller-Wickop & Schultz	2013	2.0	P	No title (Process Audits)	Compliance and Audits	D	Valid Ext	Ext MM	Explicit	No	explicit	no	no	(yes) BPMN ext.
Natschläger	2011	2.0	P	Deontic BPMN	Deontic Analysis	A	None	No	No	No	no	no	yes (deontic items)	no
Pillat et al.	2012	2.0	P	BPMNt	Software Development	D	Own Ext	Ext MM	No	No	implicit (description)	no	yes (SPEM 2.0)	(yes)
Rodriguez et al.	2007	1.x	J	No title (Security)	Risk Management	D	Own Ext	UML, OCL	Explicit	No	no	partly	yes (requirements)	no
Saeedi et al.	2010	2.0	P	No title (Serivce Quality Requirements)	Quality Management	D	Valid Ext	Ext MM	By example	No	implicit (description)	no	no	(yes) BPMN ext.
Saleem & Hassan	2012	2.0	J	No title (Security Requirements in SOA)	Risk Management	D	Own Ext	UML	Explicit	No	no	no	no	no
Schleicher et al.	2010	2.0	P	No title (Compliance Requirements)	Compliance and Audits	D	None	No	No	No	no	no	no	no
Stroppi et al.	2011	2.0	P	No title (Ressources)	Resources	E	Valid Ext	Ext MM	By example	No	no	no	no	(yes) BPMN ext.
Sungur et al.	2013	2.0	P	BPMN4WSN	Sensors	D	Own Ext	UML	Explicit	No	explicit	no	no	no
Supulniece et al.	2012	2.0	P	No title (knowledge)	Knowledge	D	None	No	Explicit	Yes	explicit	no	yes (KMDL)	no
Wolter & Schaad	2007	1.x	P	No title (Authorization Constrains)	Authorization	A	Own Ext	UML	By example	No	no	no	no	no
Zor et al.	2011	2.0	P	No title (Manufacturing)	Manufacturing	D	None	No	Explicit	Yes	no	partly	no	no

Fig. 1. Analysis of the extensions regarding their basic attributes, BPMN standard conformity and the applied extension approach or method

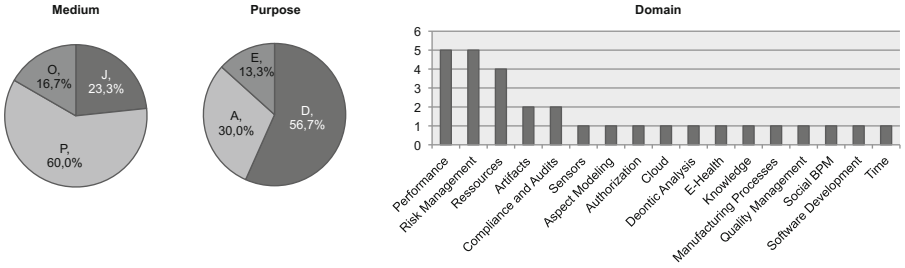

Fig. 2. Distribution of the observed extension attributes regarding the publication medium, the particular purpose and the addressd domain

reports (16,7%). Only every fourth extension is published in a journal, which could be interpreted as a lack of maturity in BPMN extension research (see figure 2). Also, we could not find any advice for cumulated research on single extensions. More than half of the publications reveal a descriptive purpose (56,7%) that aims to describe some domain (e.g., sensor networks [20]). 30% of the extensions aim to enrich BPMN for specific analytical purposes such as process cost [21]. 13,3% focuses run-time or execution-oriented issues like resource allocation constraints [22]. The targeted domains of the extensions are very heterogenous; altogether 17 domains were identified. Five publications address performance measurement [23], [24], [21], [25], [26], another five publications deal with issues related to risk management [27], [28], [29], [30], [31] and four extensions are designed for resource related issues [22], [32], [33], [34].

3.2 Standard Conformity

It is remarkable, that only 16,7% of the extensions are defined by the BPMN extension mechanism (see the first piechart in figure 3). Thus, four out of five extensions are not compliant with the BPMN meta model! These extensions are either defined by a dedicated meta modeling approach (36,7%) using UML or OCL expressions (e.g., [24], [23], [30]). Or these extensions do not have any meta model and are defined solely by new notation elements (10,0%) like [35]. 36,7% of the extensions do not present any definition! It has to be stated, that BPMN extension mechanism was introduced in version 2.0 in January 2011. Thus, actually all eleven extensions published before 2011 could not have any methodical support. However, the consideration of the 19 extensions published after 2011 reveals that only 21% were defined as BPMN extension meta models and still 32% do not provide any structured definition. It became obvious that the majority of extensions is not compliant with the BPMN standard.

Modeling language extensions generally requires the definition of customized or added notation elements (see [7], p. 44). 40% of the analyzed extensions present the extended concrete syntax be describing new graphics explicitly. Other 40% of the articles present new graphical elements implicitly within demonstration models. 20% of the extensions do not define or explicate any kind of

graphical extension. Further, BPMN specification claims to not contradict the semantics of any BPMN element. Within the analysis process, not every part of each meta model was checked due to resource limitations and due to the fact that most of the articles were peer-reviewed before publication. However, we found semantic discrepancies in four extensions: [25] uses Pools and Lanes in order to express performances, although these elements are designated for organizational units, responsibilities or roles. [35] integrates data objects within the sequence flow, although they must not have any direct effect on it. In a similar way, [36] integrates non-flow elements within the sequence flow what is not permitted. [37] specializes gateways to material gateways and use them for material transformations what is not the scope of gateways. Despite these few irregularities, the majority of the extensions do not contain semantical errors.

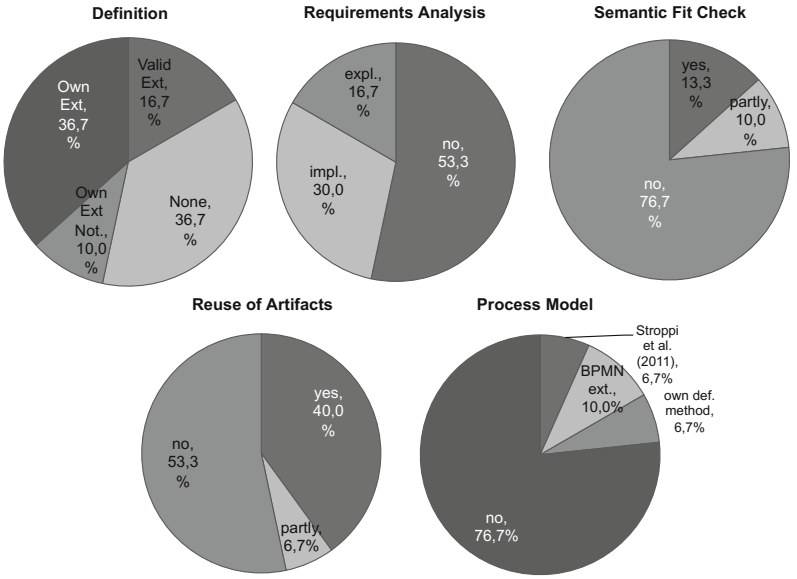

Fig. 3. Analysis of the extensions regarding their meta model definition and methodological aspects

3.3 Applied Method

As already shown in section 3.2, the BPMN extension mechanism is rarely applied. Nearly three out of four do not apply any method. These extensions are developed in an ad hoc manner, what impedes the assessment of the replicability and comprehensibility. 16,7% of the extensions were designed based on the BPMN extension model (five in total), whereby only two applied the process model of [8]: [32] and [38]. [38] extends the process model concerning a semantic equivalence check to ensure the necessity of extension. Another two extensions

were designed based on individually outlined procedures [23], [39]. Regarding the criterion of requirements analysis, approximately one of two articles provide requirements to the extension. One third was stated explicitly (e.g., by a set of requirements R1 to Rn, [20]). The rest of these articles describe requirements implicitly within the introduction or the description of the application context (e.g., [40]). Three of four articles designed the particular extension without any deep consideration of the question, whether each requirement or extension demand needs necessarily an extension concept (see the middle piechart in figure 3). 13,3% conducted a discussion for every concept [27], [33], [32], [38]. Further, nearly half of the extensions make use of existing domain artifacts. For instance, UML profiles [23], [39], domain modeling concepts [27], [32], [36] or requirements [30] are reused within the extension design.

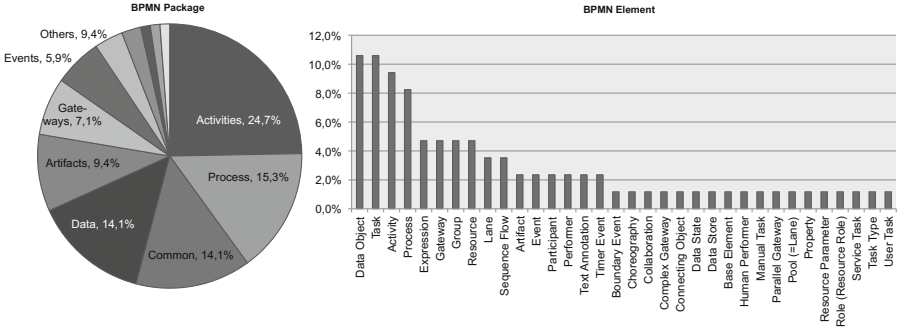

Fig. 4. Distribution of extended BPMN packages and elements

3.4 Domain-specific Extension

Within the extension analysis, only publications with at least one identifiable new element were considered. The number of new elements had a range between one and 35 elements; on an average of nearly eight elements (e.g., [41], [24]). Examples for particular new elements as well as the derived size classes can be found in figure 6. Although the definition of new diagram types is not considered within BPMN, some extensions also provide the definition of new diagrams like a Resource Structure Diagram [34] or a Secure Business Process Diagram [30], [31]. Next to the definition of new elements, BPMN elements are also extended or customized. As figure 4 shows, primarily Data Objects, Tasks, Activities and Processes are extended. This fact is also emphasized by the presentation of the extended BPMN packages: Elements from the Activity package, the Process package, the Common package (e.g., Resource, Sequence Flow or Expression) and the Data package are extended mainly. It could be concluded, that these elements are predestinated for domain-specific extensions. Especially,

Data Objects and Tasks are often specified within extensions (see figure 6, column "specialization"). The extension of standard BPMN elements is mainly realized by new relations (associations in the meta model) or specifications (inheritances). Generally, the new relations are passive. It means that they are not mandatory from the perspective of the extended element but rather optional; extending the dynamic range of the referencing element. New relations between standard elements are on rare occasions [24]. Also, the extension by owned attributes is implemented rarely (e.g., [32], [24], [40]), whereas the specification of (generic) BPMN elements by new domain specific sub classes seems to be a common means (e.g., [42], [43], [44], [29], [45]).

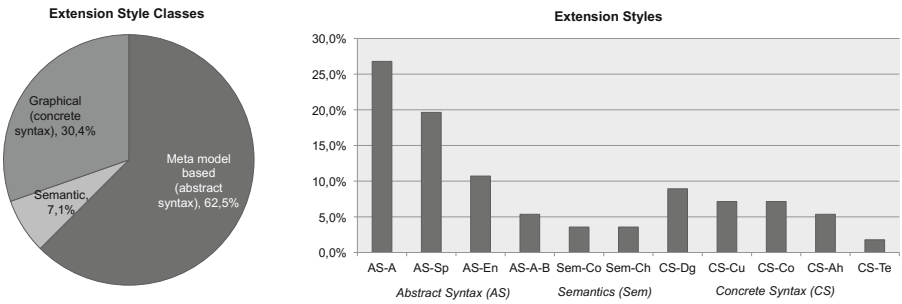

Fig. 5. Distribution of identified extensions styles (56 in total over all extensions)

Consequently, the distribution of the applied extension styles reveals that the *AS-Sp* style (specialization) is one of the most applied extension techniques (see figure 5). In total, 56 style applications could be identified within the extension definitions. 62,5% of them affect the abstract syntax (meta model), 30,4% are related to the graphical notation (concrete syntax) and 7,1% realize some extension by concrete or change some element's semantics. Unsurprisingly, the (more or less unspecific) *AS-A* style is applied the most. The enumeration technique (*AS-En*) for the domain-specific definition of ranges is used in more than 10% of all style applications. Within the area of graphical style, there is no dominating technique. Interestingly, *CS-Co* is not applied often, although BPMN explicitly emphasize this possibility for artifacts elements [7].

4 Implications

Several implications can be derived based on the analysis of existing BPMN extensions. We argue, that the following aspects should be considered in prospective research works on BPMN extensions.

Strict Use of the BPMN Extension Mechanism: As shown, only very few extensions are designed by applying the BPMN extension mechanism. However, such an implementation is indispensable for reasons of standard conformity,

	New elements				Extended or customized elements						
Authors	Elements (sample)	Count	Size Class	New Diagram	Relations	Properties	Speciali-zation	Enhance-ment	Graphical Custom.	Extended Elements	Extension Style
Altuhhov et al.	Vulnerability Points, Lock, Association Flow	3	Tiny	-	Data Store, Data Object, Task	-	-	-	-	3	CS-Cu, CS-Co, AS-A
Awad et al.	Organizational Role, Profile, Case, ...	6	Light	-	Task, Process, Lane		Role	-	-	4	AS-A, CS-Co
Bocciarelli & D'Ambrogio	Ga Workload Event, Arrival Pattern, Pa Qualification, ...	11	Large	-	Collaboration, process (direction of association is not defined)	-	-	-	-	2	AS-A, CS-Te
Brambilla et al.	Social Monitoring, Social Behaviour, URI, ...	14	Large	-	-	-	Task	-	-	1	AS-Sp
Braun & Esswein	Human Resource, Material, Allocation States, ...	17	Large	-	Timer Event, Participant	Data Object	Property, Resource	-	-	5	AS-A, AS-Sp, AS-En, Sem-Co
Braun et al.	Medical Document, CPG Reference, Diagnosis Task, ...	14	Large	-	Activity, Group, Gateway, Process, Data Object	-	Data Object, Task, Parallel Gateway, Complex Gateway, Expression	-	-	9	AS-A, AS-Sp, AS-En, Sem-Co
Brucker et al.	Access Control, Separation of Duty, Binding of Duty, ...	(4)	Tiny		-	-	-	-	-	0	AS-A
Charfi et al.	Pointcuts, Advice	(2)	Tiny		-	-	-	-	-	0	CS-Cu
Friedenstab et al.	Duration, Process Section, Quantitative Limit, ...	35	Heavy	(yes)	Data Object, Data State, Process, Expression, Sequence Flow	Activity, Process	-	-	-	6	AS-A, AS-Sp, AS-En, CS-Dg
Gagne & Trudel	Time Date Expressions, Temporal Constraint Decorators	(2)	Tiny	-	(Temporal Dependencies, Temporal Constraint Attributes)		-	-	-	0	AS-A
Großkopf	Performer Role, Processor Role, Actor	3	Tiny	-	-	Activity	-	-	-	1	AS-A
Kopp et al.	TOSCA Node Management Task, TOSCA Script Task, TOSCA Data Object, ..	(4)	Tiny	-	-	-	Task, Data Object	-	-	2	AS-Sp
Korherr & List	Waiting Time, Working Time, Process Goal, ...	12	Large	-	Pool, Process, Group, Event, Timer Event	-	-	-	-	5	AS-A-B
Lodhi et al.	Metric Values, Colors, Rules/Condition, Probability, Dimensional Attributes, Content/Structure	6	Light	-	Activity, Gateways, Connecting Objects, Artefacts, Lanes	-	-	-	-	5	AS-A, CS-Co, Sem-Ch
Lohmann & Nyolt	Placeholder container	(1)	Tiny	(yes)	-	-	Events, Choreographies (only graphically!)	-	-	2	CS-Cu, (CS-Dg)
Magnani & Montesi 2007	Simple Costs, Cost Intervals, Average Costs	3	Tiny	-	Task, Gateway, Activity	-	-	-	-	3	CS-Ah
Magnani & Montesi 2009	Store, Entity, Relationship, ...	7	Light	-	-	-	-	-	Data objects	1	CS-Co
Marcinkowski & Kuciapski	Risk Factor, Risk Type, Risk Handler	9	Light	-	Sequence Flow, Resource, Participant	-	Artefact, Task	-	-	5	AS-Sp, AS-En
Müller-Wickop & Schultz	Account Entry, Debit And Credit, Account, Balance	4	Tiny	-	-	-	Data Object, Group, Text Annotation	-	-	3	AS-Sp
Natschläger	Deontic Task, O(Task), X(Task), P(Task), ...	(6)	Light	-	-	-	Task	-	-	1	-
Pillat et al.	Tailored Base Element	1	Tiny	-	-	-	-	-	-	0	AS-Sp
Rodriguez et al.	Security Requirement, Integrity, Privacy, ...	13	Large	yes	-	-	Process (Business Process Diagram)	-	-	1	AS-A-B, AS-A, CS-Dg
Saeedi et al.	Reliability, Cost, Response Time, ...	7	Light	-	Activity	-	-	-	-	1	AS-A
Saleem & Hassan	Access Control, Authentication, Authorization, ...	8	Light	yes	-	-	-	-	-	0	AS-Sp, CS-Cu, CS-Dg
Schleicher et al.	Compliance Scope, Compliance Rule, Language Indicator	(3)	Tiny	-	-	-	-	-	-	0	AS-A-B
Stroppi et al.	Resource Privilage, Resource Base, Subsumption, ...	17	Large	yes	Resource, Resource Parameter, Element, Boundary Event, User Task, Human Performer, Expression	-	-	-	-	7	AS-A, AS-En, CS-Dg
Sungur et al.	WSN Task, tWSNOperation, tWSNPerformer, ...	7	Light	-	(Resource Assignment) Expression	-	Performer, Service Task	-	-	3	AS-Sp, AS-A, AS-En
Supulniece et al.	Data Object, Information Object, Knowledge	(3)	Tiny	-	Activity, Task Type, Role (Performer), Data Object	-	-	-	-	4	CS-Ah, (Sem-Ch)
Wolter & Schaad	Authorization Constraint	1	Tiny	-	Manual Task, Group (new relation, Lane	-	Text Annotation	-	-	4	AS-A, AS-Sp
Zor et al.	Parts Container, Machines and Tools Flow Connector, ...	10	Light	-	-	-	Gateway, Task, Resource, Sequence Flow	Activity	-	5	CS-Ah

Fig. 6. Analysis of the extensions regarding new and extended elements

comprehensibility, model exchange and tool support. For instance, model engineers fail in reusing the most BPMN extensions since they do not provide a valid BPMN extension model. Thus, it is necessary to transform the provided dedicated meta model into a BPMN conform model in order to integrate it within a BPMN tool. Also, the communication within the research community is hampered by this shortcoming. In the context of method engineering, it is also necessary to define the concrete syntax of each extension element explicitly to avoid misunderstandings. The semantics of a new element or its relations to BPMN elements should be described in detail in order to support its application.

Integrated Methodological Support Is Necessary: As stated at the beginning of this article, BPMN lacks in term of providing an extension process model. Thus, most of the considered BPMN extensions are not designed rigorously. There seems to be a gap between the domain-specific definition of extension requirements, their conceptualization and the implementation as valid meta model. The last aspect is successfully solved by [8] and few extensions make use of its proclaimed transformation procedure. However, the early phases of extension planning and design are still not guided. Therefore, we see the need for an integrated process model for BPMN extension development that focuses the domain analysis and conceptualization phase. For example, there should be a systematic support for the decision whether any domain concept can be represented within the "semantic scope" of a standard element or not. We suppose, that more than a few BPMN extensions do not exploit the entire expressiveness of BPMN. Besides, research on the integration of domain-specific artifacts within BPMN extensions and DSMLs in general should be intensified, since such artifacts (e.g., ontologies, taxonomies) provide well-defined domain knowledge that could complement domain expert knowledge.

BPMN Language and Extension Design: It became obvious, that specific aspects are often demanded. Especially, a better resource and data object modeling needs to be supported by BPMN, albeit BPMN will not be understood as any kind of a "data-flow language" ([7], p. 22). Referring to enterprise architecture frameworks, an extension of the BPMN regarding several views (e.g., resource perspective) is promising. Hereof, further research should consider the question, how to extend BPMN with new diagrams or views. Currently, such an extension is not designated. Also, based on our analysis of the so-called extension styles, a deeper analysis of extension patterns is necessary in order to provide specific patterns or guidelines for given extension purposes.

5 Conclusion

To the best of our knowledge, this is the first approach addressing the comparison and classification of BPMN extensions in order to present the current state of the art. Therefore, an extensive literature review of BPMN extensions was conducted that results in a set of 30 publications that were subjected an in-depth analysis. For the comparison and classification, a four-part extension analysis framework was designed containing criteria on the extension itself and

the applied procedures. Based on the application of this framework, several implications were derived.

First, authors of BPMN extension should strictly use the BPMN extension mechanism in order to provide a valid extension and enable model exchangeability (currently less than 20% provide valid BPMN extensions). Second, we identified a need for an integrated methodological support of extension development, especially in terms of domain analysis and the comparison of domain elements with BPMN standard elements. Third, we have identified a recognizable need for the support of resource and data oriented modeling aspects within BPMN. In this context, especially the question of extending BPMN (or a modeling language in general) with new diagrams and views should be considered. Regarding to the identified extension styles, it might be promising to develop extension patterns (or at least guidelines) for specific extension needs in order to support the design process.

References

1. Chinosi, M., Trombetta, A.: Bpmn: An introduction to the standard. Computer Standards and Interfaces 34(1), 124–134 (2012)
2. ISO: Iso/iec 19510:2013 - International organization for standardization (iso) (2013)
3. Frank, U.: Outline of a method for designing domain-specific modelling languages. ICB Research Report 42, Universität Duisburg-Essen, Essen (2010)
4. Mohagheghi, P., Haugen, Ø.: Evaluating domain-specific modelling solutions. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010. LNCS, vol. 6413, pp. 212–221. Springer, Heidelberg (2010)
5. Lämmel, R., Ostermann, K.: Software extension and integration with type classes. In: Proceedings of the 5th International Conference on Generative Programming and Component Engineering, pp. 161–170. ACM (2006)
6. Kopp, O., Görlach, K., Karastoyanova, D., Leymann, F., Reiter, M., Schumm, D., Sonntag, M., Strauch, S., Unger, T., Wieland, M., et al.: A classification of bpel extensions. Journal of Systems Integration 2(4), 3–28 (2011)
7. OMG: Business Process Model and Notation (BPMN) - Version 2.0. Object Management Group (OMG) (2011)
8. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extending BPMN 2.0: Method and tool support. In: Dijkman, R., Hofstetter, J., Koehler, J. (eds.) BPMN 2011. LNBIP, vol. 95, pp. 59–73. Springer, Heidelberg (2011)
9. Braun, R., Schlieter, H.: Requirements-based development of bpmn extensions – the case of clinical pathways. In: 1st International Workshop on the Interrelations between Requirements Engineering and Business Process Management (2014)
10. Hevner, A., Chatterjee, S.: Design science research in information systems. In: Design Research in Information Systems. Integrated Series in Information Systems, vol. 22, pp. 9–22. Springer, US (2010)
11. Cooper, H.M.: Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge in Society 1(1), 104–126 (1988)
12. Vom Brocke, J., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., Cleven, A.: Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: ECIS 2009 Proceedings, Paper 161 (2009)

13. WKWI: Wi-orientierungslisten. Wirtschaftsinformatik 50(2), 155–163 (2008)
14. Decker, G., Puhlmann, F.: Extending BPMN for modeling complex choreographies. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 24–40. Springer, Heidelberg (2007)
15. Gao, F., Zaremba, M., Bhiri, S., Derguerch, W.: Extending bpmn 2.0 with sensor and smart device business functions. In: 2011 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 297–302. IEEE (2011)
16. Zor, S., Görlach, K., Leymann, F.: Using bpmn for modeling manufacturing processes. In: Proceedings of 43rd CIRP International Conference on Manufacturing Systems, pp. 515–522 (2010)
17. Pardillo, J.: A systematic review on the definition of UML profiles. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part I. LNCS, vol. 6394, pp. 407–422. Springer, Heidelberg (2010)
18. Natschläger, C.: Deontic BPMN. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part II. LNCS, vol. 6861, pp. 264–278. Springer, Heidelberg (2011)
19. Schleicher, D., Leymann, F., Schumm, D., Weidmann, M.: Compliance scopes: Extending the bpmn 2.0 meta model to specify compliance requirements. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–8. IEEE (2010)
20. Sungur, C.T., Spiess, P., Oertel, N., Kopp, O.: Extending bpmn for wireless sensor networks. In: 2013 IEEE 15th Conference on Business Informatics (CBI), pp. 109–116. IEEE (2013)
21. Korherr, B., List, B.: Extending the epc and the bpmn with business process goals and performance measures. In: ICEIS (3), pp. 287–294 (2007)
22. Awad, A., Grosskopf, A., Meyer, A., Weske, M.: Enabling resource assignment constraints in bpmn. Hasso Plattner Institute, Potsdam (2009)
23. Bocciarelli, P., D'Ambrogio, A.: A bpmn extension for modeling non functional properties of business processes. In: Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, Society for Computer Simulation International, pp. 160–168 (2011)
24. Friedenstab, J., Janiesch, C., Matzner, M., Muller, O.: Extending bpmn for business activity monitoring. In: 2012 45th Hawaii International Conference on System Science (HICSS), pp. 4158–4167. IEEE (2012)
25. Lodhi, A., Küppen, V., Saake, G.: An extension of bpmn meta-model for evaluation of business processes. Scientific Journal of Riga Technical University. Computer Sciences 43(1), 27–34 (2011)
26. Magnani, M., Cucci, F.: BPMN: How much does it cost? An incremental approach. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 80–87. Springer, Heidelberg (2007)
27. Altuhhov, O., Matulevičius, R., Ahmed, N.: An extension of business process model and notation for security risk management. International Journal of Information System Modeling and Design (IJISMD) 4(4), 93–113 (2013)
28. Brucker, A.D., Hang, I., Lückemeyer, G., Ruparel, R.: Securebpmn: Modeling and enforcing access control requirements in business processes. In: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, pp. 123–126. ACM (2012)
29. Marcinkowski, B., Kuciapski, M.: A business process modeling notation extension for risk handling. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 374–381. Springer, Heidelberg (2012)

30. Rodríguez, A., Fernández-Medina, E., Piattini, M.: A bpmn extension for the modeling of security requirements in business processes. IEICE Transactions on Information and Systems 90(4), 745–752 (2007)
31. Saleem, M., Jaafar, J., Hassan, M.: A domain-specific language for modelling security objectives in a business process models of soa applications. AISS 4(1), 353–362 (2012)
32. Braun, R., Esswein, W.: Extending bpmn for modeling resource aspects in the domain of machine tools. WIT Transactions on Engineering Sciences (87), 450–458 (2014)
33. Großkopf, A.: An extended resource information layer for bpmn. Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam (2007)
34. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extended resource perspective support for bpmn and bpel. CIbSE, 56–69 (2012)
35. Magnani, M., Montesi, D.: BPDMN: A conservative extension of bpmn with enhanced data representation capabilities. arXiv preprint arXiv:0907.1978 (2009)
36. Supulniece, I., Businska, L., Kirikova, M.: Towards extending bpmn with the knowledge dimension. Enterprise, Business-Process and Information Systems Modeling, 69–81 (2010)
37. Zor, S., Leymann, F., Schumm, D.: A proposal of bpmn extensions for the manufacturing domain. In: Proceedings of 44th CIRP International Conference on Manufacturing Systems (2011)
38. Braun, R., Schlieter, H., Burwitz, M., Esswein, W.: Bpmn4cp: Design and implementation of a bpmn extension for clinical pathways. Research Report TU Dresden
39. Pillat, R.M., Oliveira, T.C., Fonseca, F.L.: Introducing software process tailoring to bpmn: Bpmnt. In: 2012 International Conference on Software and System Process (ICSSP), pp. 58–62. IEEE (2012)
40. Gagne, D., Trudel, A.: Time-bpmn. In: IEEE Conference on Commerce and Enterprise Computing, CEC 2009, pp. 361–367. IEEE (2009)
41. Charfi, A., Müller, H., Mezini, M.: Aspect-oriented business process modeling with ao4bpmn. Modelling Foundations and Applications, 48–61 (2010)
42. Brambilla, M., Fraternali, P., Vaca Ruiz, C.K.: Combining social web and bpm for improving enterprise performances: the bpm4people approach to social bpm. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 223–226. ACM (2012)
43. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: BPMN4TOSCA: A domain-specific language to model management plans for composite applications. In: Mendling, J., Weidlich, M. (eds.) BPMN 2012. LNBIP, vol. 125, pp. 38–52. Springer, Heidelberg (2012)
44. Lohmann, N., Nyolt, M.: Artifact-centric modeling using BPMN. In: Pallis, G., Jmaiel, M., Charfi, A., Graupner, S., Karabulut, Y., Guinea, S., Rosenberg, F., Sheng, Q.Z., Pautasso, C., Ben Mokhtar, S. (eds.) ICSOC 2011 Workshops. LNCS, vol. 7221, pp. 54–65. Springer, Heidelberg (2012)
45. Wolter, C., Schaad, A.: Modeling of task-based authorization constraints in BPMN. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 64–79. Springer, Heidelberg (2007)