Representation Theory versus Workflow Patterns - The Case of BPMN

Jan Recker[1], Petia Wohed[2], and Michael Rosemann[1]

[1] Queensland University of Technology
126 Margaret Street, Brisbane QLD 4000, Australia
{j.recker, m.rosemann}@qut.edu.au
[2] Stockholm University/The Royal Institute of Technology
Forum 100, 164 40 Kista, Sweden
petia@dsv.su.se

Abstract. Selecting an appropriate process modeling language forms an important task within business process management projects. A wide range of process modeling languages has been developed over the last decades, leading to an obvious need for rigorous theory to assist in the evaluation and comparison of the capabilities of these languages. While academic progress in the area of process modeling language evaluation has been made on at least two premises, Representation Theory and Workflow Patterns, it remains unclear how these frameworks relate to each other. We use a generic framework for language evaluation to establish similarities and differences between these acknowledged reference frameworks and discuss how and to what extent they complement respectively substitute each other. Our line of investigation follows the case of the popular BPMN modeling language, whose evaluation from the perspectives of Representation Theory and Workflow Patterns is reconciled in this paper.

1 Introduction

Improving and managing business processes continues to be on the top of the agenda for chief executives [1]. This strong momentum has, over time, led to the development of a wide range of solutions and approaches for Business Process Management. One prominent example in this context is the increased popularity of business process modeling [2]. Recently, "yet another" process modeling language has entered the BPM domain, the Business Process Modeling Notation (BPMN) [3]. The conformity with emerging Web Services standards, its reasonably intuitive notation and the promise of becoming an official process modeling industry standard, have boosted the popularity of BPMN. The attention that BPMN has been receiving since its first release, however, had at the time of release not been balanced by a critical analysis of its actual and perceived capabilities. Quite contrary indeed, the proliferation of arbitrary approaches to process modeling has led to a need for rigorous theory to assist in the evaluation and comparison of process modeling languages. Van der Aalst

[4] points out that many of the available 'standards' for process and workflow specification lack critical evaluation. Along similar lines, Moody [5] states a concern about lacking evaluation research with respect to the conceptual modeling of the dynamics of information systems. In fact, the large selection of currently available process modeling languages stands in sharp contrast to the paucity of evaluation frameworks that can be used for the task of evaluating and comparing those modeling languages. However, while there is un-fortunately not one single framework that facilitates a comprehensive analysis of all facets of a process modeling language (*e.g.*, expressive power, consistency and correctness of its meta model, perceived intuitiveness of its notation, available tool sup-port), reasonably mature research has emerged over the last decade with a focus on the representational capabilities and expressive power of modeling languages. In academia two examples, Representation Theory [6–8] and the Workflow Patterns Framework [9–11], have emerged as well-established evaluation frameworks in the field of process modeling. What remains unclear, however, is how these frameworks relate to each other. Are they complementary in their approaches and are their results comparable? What types of insights into expressive power and shortcomings of a process modeling language can be obtained from them? These and related questions can be traced back to Moody's [5] argument that a proliferation of different quality measurement proposals is counterproductive to research progress; in fact, the existence of multiple competing proposals is a sign of an immature research field. What is needed is a reconciliation and synthesis of available proposals in order to establish consensus on a common understanding of conceptual modeling quality [5, p. 258]. Taking together the ongoing proliferation of prospective standard languages for process modeling and the need for a reconciliation of quality frameworks, our paper seeks to contribute to the body of knowledge on at least two premises. First, we apply a framework for language evaluation to both Representation Theory and Workflow Patterns Framework in order to establish commonalities and differences between these two quality proposals. As a second contribution we use the example of the most recent and prominent candidate for a process modeling standard, BPMN, as a language that is evaluated by both frameworks; thereby we are able to reconcile the analyses of BPMN and give a comprehensive picture of its capabilities and shortcomings.

We proceed as follows. First we briefly introduce our selected example, BPMN, and discuss studies related to our research (section 2). We then establish a framework for language evaluation and apply it to the frameworks in question (section 3). Section 4 presents and discusses our reconciliation of the frameworks, and also gives a synthesis of the analyses of BPMN. We close in section 5 by summarizing our work and outlining future research opportunities.

2 Background & Related Work

2.1 Overview of the Process

In the remainder of this paper we will refer to previous analyses of the Business Process Modeling Notation (BPMN) as examples for our elaborations. In this section we briefly introduce BPMN in order to give the reader sufficient background for understanding our subsequent argumentations.

BPMN was developed by the Business Process Management Initiative and adopted by OMG for standardization purposes in February 2006 [3]. The development of BPMN stemmed from the demand for a graphical notation that complements the BPEL4WS standard for executable business processes. Although this gives BPMN a technical focus, it has been the intention of the BPMN designers to develop a modeling technique that can be applied for typical business modeling activities as well. The complete BPMN specification defines thirty-eight language constructs plus attributes, grouped into four basic categories of elements, *viz.*, Flow Objects, Connecting Objects, Swimlanes and Artefacts. *Flow Objects*, such as events, activities and gateways, are the most basic elements used to create Business Process Diagrams (BPDs). *Connecting Objects* are used to inter-connect Flow Objects through different types of arrows. *Swimlanes* are used to group activities into separate categories for different functional capabilities or responsibilities (*e.g.*, different roles or organizational departments). Finally, *Artefacts* may be added to a diagram where deemed appropriate in order to display further related information such as processed data or other comments. Refer to the specification [3] for further information on BPMN.

2.2 Related Work

Work related to our study can broadly be differentiated into (a) research on the evaluation of process modeling languages in general and of BPMN in particular, and (b) research on the comparison of evaluation techniques for conceptual models. We briefly recapitulate such related work in this section and will, where appropriate, refer to it in the later sections of this paper.

In the area of evaluation of process modeling languages, only little research has tried to compare process modeling languages based on an established theoretical model. The most prominent example for an evaluation framework that has deductively been derived from established theory is the Bunge-Wand-Weber (BWW) representation model [6–8] that forms the core of Representation Theory. The BWW representation model, which will be discussed in more detail in section 3.1 of this paper, has a strong track record in the area of process modeling, for instance in the evaluation of Petri Nets, EPCs, BPMN, ebXML and others. A comprehensive annotated overview is given in [12].

A second example of a theoretical sound quality proposal is the Workflow Patterns framework [9–11], which will also be considered in more detail in this paper later on. Since its establishment, the framework that has inductively been derived from observable practice in workflow management has been widely used both as a benchmark for analysis and comparison of languages. A comprehensive annotated overview is given in [13].

Besides these two established proposals it is worthwhile to mention the semiotic quality framework [14], which is a well-discussed framework for evaluating the quality of conceptual models in general. However, it has so far only sparingly been applied to the domain of process modeling [15]. Research related directly to the evaluation of BPMN is still limited due to the recency of its release. The semiotic quality framework [14] has been used to evaluate BPMN with respect to the criteria domain appropriateness, participant language knowledge appropriateness, knowledge externalizability appropriateness, comprehensibility appropriateness, and technical actor interpretation appropriateness analytically [16] and empirically [17]. Both studies conclude that BPMN particularly excels in terms of comprehensibility appropriateness due to its construct specializations and type aggregations, is overall well-suited for the domain of business process modeling but achieves rather modest results in domain appropriateness. In preparation for this study, the Workflow Patterns framework was used to evaluate BPMN [13]. The results from this evaluation show that BPMN performs well in terms of capturing the control flow and handling data in a process but is limited in expressing resources and the work distribution of activities among them. Also, in preparation for this study, was BPMN analyzed as per Representation Theory [18]. The analysis proposed, and empirically confirmed, shortcomings related to organizational modeling due to unclear specifications of the Pool and Lane constructs. Also, representational shortcomings were found, amongst others, in the specification of business rules. Both analyses will be further discussed in section 4 of this paper.

Regarding related work on comparing evaluation techniques, Siau and Rossi [19] provide a classification of evaluation approaches for modeling methods and differentiate multiple proposals into analytical and empirical approaches. They discuss analyses based on the BWW model, however, an evaluation based on workflow pat-terns does not fit into their classification scheme. We see a reason for this in the scope of Siau and Rossi's study, which focused modeling *methods* rather than modeling *languages*. Similarly, Recker [20] proposes a comparison framework that comprises the facets *model perception*, *evaluation perception* and *quality perception*, in order to assess the suitability of modeling language evaluation approaches in various research contexts. He argues that the suitability of any evaluation approach is determined by the conformity of its underlying epistemological viewpoints to the overall presuppositions of the research context.

3 Evaluating Process Modeling Languages - A Theoretical Perspective

3.1 Framework for Language Evaluation

Before we compare Representation Theory and the Workflow Patterns framework it is necessary to appreciate the theoretical analysis model that underlies research on language evaluation. The purpose of the current section is to define a framework for language evaluation under which existing approaches can be subsumed.

In order to establish this framework we draw on the generally acknowledge objective of conceptual modeling, which is to build a representation of a selected domain of interest for the purpose of understanding and communication among stakeholders in the process of requirements engineering for Information Systems analysis and design [21]. These stakeholders are confronted with the need to represent the requirements in a conceptual form, *viz.*, an underlying conceptual structure is needed on which conceptual models can be based [22]. As such underlying conceptual structures are dependant on, *inter alia*, modeling purpose and the preferences of the involved modeling participants, they cannot be equated for anyone. They merely denote potentially valid modeling references that hold true in certain but not all modeling contexts. The overall lack of such underlying conceptual structures for conceptual modeling motivated research on reference frameworks for conceptual models in given domains, against which modeling languages can be assessed as to their compliance with the framework, leading to statements about the 'goodness' of the resulting model in light of the selected framework. Fig. 1 explicates these relations.

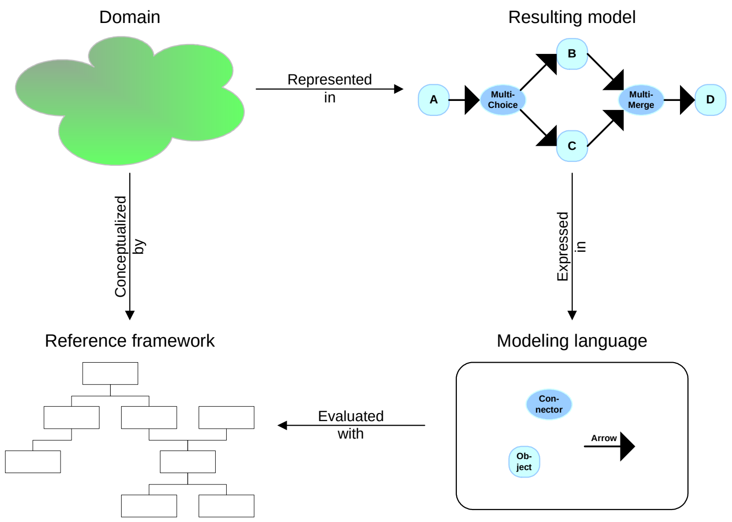


Fig. 1. Relations between domain, reference framework, modeling language and model

According to Fig. 1, a modeling reference framework, such as the BWW representation model or the Workflow Patterns framework, can be used as a heuristic specification of the domain to be modeled. As an example, the Workflow Patterns framework conceptualizes the domain of processes in form of atomic chunks of workflow semantics, differentiated in the perspectives of control flow, data and resources. In order to assess whether a given modeling language is 'good' with respect to its capability to represent relevant aspects of the domain, the reference framework serves as a theoretical benchmark in the evaluation and comparison of available modeling languages. The assumption of this type of research is that capabilities and shortcomings of a conceptual modeling language in light of the reference framework in use ultimately affect the quality of the model produced.

Taking these elaborations into account, the process of evaluating modeling languages against a reference framework consists of a pair wise bi-directional mapping between the constructs specified in the reference framework against the constructs specified in the modeling language. For example, the Workflow Patterns framework assesses which of the specified patterns can be expressed by a given language. The basic assumption is that any deviation from a 1-1 relationship between the corresponding constructs in the reference framework and the modeling language leads to situations of deficiency and/or ambiguity in the use of the language, thereby potentially diminishing the quality of the model produced.

Formally, the relationships between what can be represented (constructs of the modeling language) and what is represented (constructs of the reference framework as a heuristic for the domain being modeled) can be specified as follows (see Fig. 2)[3].

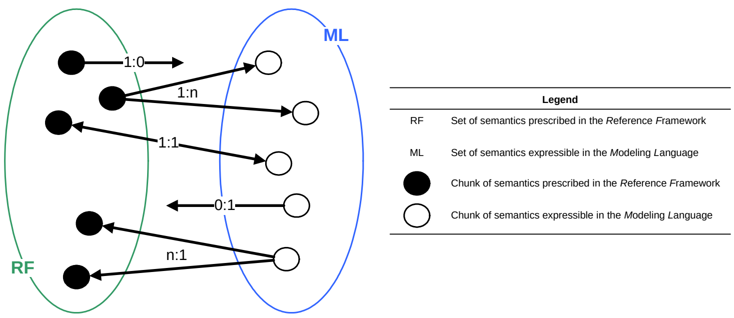


Fig. 2. Framework for language evaluation

- *Equivalence*: The construct prescribed by the reference framework can unequivocally be mapped to one and only one construct of the modeling language (1:1 mapping).
- *Deficiency*: The construct prescribed by the reference framework cannot be mapped to any construct of the modeling language (1:0 mapping).

[3] Note that the framework for language evaluation presented here draws on previous work in related disciplines. Weber [23], for instance uses a similar albeit not identical framework to explain the two situations of ontological completeness and clarity of a language. Guizzardi [24] argues along similar lines in the context of structural specifications. Gurr [25] uses similar mapping relations to analyze diagrammatic communication. We do not claim to supersede the works of these authors but merely build upon their works to explain in general the research type of language evaluation.

– *Indistinguishability*: The construct prescribed by the reference framework can be mapped to more than one construct of the modeling language (1:n mapping).
– *Equivocality*: More than one construct prescribed by the reference framework can be mapped to one and the same construct of the modeling language (n:1 mapping).
– *Overplus*: Not one construct prescribed by the reference framework can be mapped to the construct of the modeling language (0:1 mapping).

Having defined hypothetical relationships that may occur in a pair wise bidirectional mapping between a reference framework and a given modeling language we can now turn to existing frameworks in the research field of process modeling in order to investigate which of these potential constellations are covered in the respective evaluation approach. For the purpose of this study we selected the Bunge-Wand-Weber representation model, which forms the core of Representation Theory, and the Workflow Patterns framework as indications for available reference frameworks in the domain of process modeling. Subsequently, we will briefly introduce both approaches.

3.2 Frameworks for the Evaluation of Process Modeling Languages

The Bunge-Wand-Weber Representation Model The development of the representation model that is known as the Bunge-Wand-Weber model stemmed from the observation that, in their essence, computerized information systems are representations of real world systems. Wand and Weber [6–8] suggest that in order to help define and build information systems that faithfully represent real world systems, models of information systems and thus their underlying modeling language must contain the necessary representations of real world constructs including their properties and interactions. The BWW representation model contains four clusters of constructs that are deemed necessary to faithfully model information systems: things including properties and types of things; states assumed by things; events and transformations occurring on things; and systems structured around things [23, 12]. The BWW model defines a *theory of representation* that has been developed deductively from philosophical research, in particular an ontology defined by Bunge [26].

The BWW model allows for the evaluation of modeling languages with respect to their capabilities to provide complete and clear descriptions of the IS domain being modeled [23]. Referring to the five types of relations specified above (see Fig. 2, the completeness of a description can be measured by the degree of *construct deficit*, *i.e.*, deficiency. The clarity of a description can be measured by the degrees of *construct overload*, *i.e.*, equivocality, *construct redundancy*, *i.e.*, indistinguishability, and *construct excess*, *i.e.*, overplus. Although implicitly being measured by the extent of deficiency, we were not able to locate any previous analysis based on the BWW model that explicitly documented equivalence of a modeling language.

The Workflow Patterns Framework The development of the Workflow Patterns framework was triggered by a bottom-up analysis and comparison of workflow management software. Provided during 2000 and 2001, this analysis included the evaluation of 15 workflow management systems, with focus being given to their underlying modeling languages. The goal was to bring insights into the expressive power of the underlying languages and hence outline similarities and differences between the analyzed systems. During the work 20 *control-flow patterns* [9] were inductively derived. These patterns in the control-flow context denote atomic chunks of behavior capturing some specific process control requirements. The identified patterns span from simple to complex control-flow scenarios and provide a taxonomy for the control-flow perspective of processes.

Recently, the Workflow Patterns framework was extended to also cover pattern constructs for the data and the resource perspectives of workflows. While the control-flow perspective focuses extensively on the ordering of the activities within a process, the data perspective focuses on the data representation and handling in process-aware information systems. The resource perspective further complements the approach with focusing on describing the various ways in which work is distributed amongst and managed by the resources associated with a business process. In 2005, a set of 43 *resource patterns* [10] and a set of 40 *data patterns* [11] were added to the framework. All control-flow, resource and data pattern constructs are grouped into various clusters.

Referring back to the five types of relations specified above (see Fig. 2, evaluations using the Workflow Patterns framework focus on the identification of potential representations within a given modeling language for each of the patterns (*i.e.*, on identification of equivalence). The non-identification of a representation for a pattern denotes a deficiency of the language. The identification of alternative representations of a pattern denotes indistinguishability. Previous analyses based on this framework have not explicitly taken into consideration the constellations of overplus and equivocality. While the performed analysis can be used to partially reveal some equivocality, it is not sufficient to identify and reason about overplus.

4 Reconciling the Evaluation Frameworks - The Case of BPMN

Based on the elaborations in section 3.1 we argue that it is possible to pair wise compare the findings obtained from analyses using Representation Theory and Workflow Patterns by using the framework for language evaluation defined in Fig. 2. In preparation for this study we analyzed BPMN against the BWW model [18] and the Workflow Patterns framework [13]. In the following we reconcile these analyses in order to extract similarities and differences in the reference frameworks. This allows us to address both objectives of this paper, *viz.*, delivering a comprehensive evaluation of the capabilities of BPMN and studying to what extent the two frameworks under observation complement respectively substitute each other.

4.1 Evaluation Frameworks Synthesis

In previous studies we have used the frameworks in questions to evaluate BPMN individually. Due to space restrictions we cannot outline the individual analyses here but refer to our previous studies described in [18] and [13]. We fitted the results of these analyses into Table 1, structured in accordance to the framework for language evaluation (see Fig. 2)[4]. Subsequently we pair wise compare the findings derived from each analysis for each of the five mapping relations.

In conducting the pair wise comparison two researchers first individually cross-evaluated the findings from each analysis, then met to defend their evaluation. A second, joint draft of the pair wise comparison was then presented to, and discussed with, a third member of the research team. By reaching a consensus over the third, joint draft of the comparison we feel that we have displayed sufficient reliability and validity of our evaluation.

Equivalence From Table 1 it can be observed that from a Representation Theory perspective, there is not a single language construct in BPMN that is unambiguously and unequivocally specified. While this finding per se is problematic as the usage of any given construct potentially causes confusion in the interpretation of the resulting model (for empirical support for this proposition refer, for instance, to [18]), the Workflow Patterns framework shows that the atomic constructs provided in BPMN can nevertheless be arranged in a meaningful, unambiguous manner to arrange a series of control-flow, data and resource patterns. This indicates that it may not be sufficient to analyze languages solely on a construct level, but it is moreover required to assess the modeling context in which the language constructs are used to compose "chunks" of model semantics. In this regard, the Workflow Patterns framework appears to be an extension in the level of analysis of Representation Theory as it transcends the construct level by specifically taking into consideration the capability of a language to compose atomic language constructs to sets of preconceived domain semantics such as control-flow patterns.

Deficiency Table 1 strongly suggests a lack of capability of BPMN to model state-related aspects of business processes. Both analyses reveal that BPMN is limited in modeling states assumed by things [18] and state-based patterns [13], respectively. Here, the two frameworks complement each other and together make a strong case for a potential revision and extension of the BPMN specification in order to advance its capability of modeling state-related semantics.

[4] For the Workflow Patterns-based evaluation, note that CP7, CP9 and CP17 have partial representations, *i.e.*, they present solutions that are not general enough to hold for all potential scenarios but may be used in some cases. Also note that, for the cluster equivocality, the differences between the solutions are captured though advanced attribute settings. The attribute settings can indeed be graphically captured through text annotations, however, such text annotations lie in our opinion outside the graphical notation of a language.

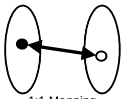
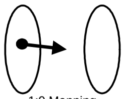
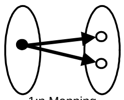
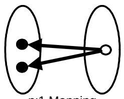
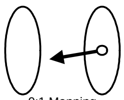
Relation	Workflow Patterns	Representation Theory
 1:1 Mapping *Equivalence*	The following Workflow Patterns can unequivocally be expressed in BPMN: *CP1, CP11-14, CP19; RP11, RP14, RP19, RP36, RP39, RP42; DP1, DP2, DP5, DP10i, DP10ii, DP11i, DP11ii, DP15-18, DP27, DP28, DP31, DP34, DP36, DP38-40*	There is no construct in the BWW model that can unequivocally be mapped to a single BPMN construct.
 1:0 Mapping *Deficiency*	The are no representations in BPMN for the following Workflow Patterns: *CP7, CP9, CP15, CP17, CP18; RP3-10, RP12, RP13, RP15-18, RP20-35, RP37, RP38, RP40, RP41, RP43; DP3, DP4, DP6, DP7, DP8, DP12-14, DP19-26, DP29, DP30, DP32, DP33, DP35, DP37*	There are no representations in BPMN for the following BWW constructs: *State, Stable State, Unstable State, Conceivable State Space, State Law, Lawful State Space, Conceivable Event Space, Lawful Event Space, History, Property* (*in particular, hereditary, emergent, intrinsic, mutual: non-binding, mutual: binding, attributes).*
 1:n Mapping *Indistinguishability*	The following Workflow Patterns have multiple representations in BPMN: *CP2-6, CP10, CP16, CP20; RP1, RP2; DP9*	The following BWW constructs have multiple representations in BPMN: *Thing, Property (in general), Class, Event, External Event, Internal Event, Well-defined Event, Poorly-defined Event, Transformation, Lawful Transformation* (including *Stability Condition, Corrective Action), Acts On, Coupling, System, System Decomposition, System Composition, System Environment, Subsystem, Level Structure.*
 n:1 Mapping *Equivocality*	The following Workflow Patterns have the same graphical representations in BPMN: *CP4 and CP6; CP9, CP12, CP13 and CP14*	The following BPMN constructs represent many BWW constructs: *Lane (Thing, Class, Kind, System, System Decomposition, System Composition, System Environment, Subsystem, Level Structure); Pool (Thing, Class, System, System Decomposition, System Composition, System Environment, Subsystem, Level Structure); Message Flow (Acts On, Coupling); Start Event (Internal Event, External Event); Intermediate Event (Internal Event, External Event); End Event (Internal Event, External Event); Error (Internal Event, External Event); Cancel (Internal Event, External Event); Compensation (Internal Event, External Event);*
 0:1 Mapping *Overplus*	Workflow Patterns analysis does not lead to statements about a possible overplus of patterns, which a language may be able to represent but which are not included in the framework.	The following BPMN constructs do not map to any BWW construct: *Link, Off-Page Connector, Gateway Types, Association Flow, Text Annotation, Group, Activity, Looping, Multiple Instances, Normal Flow, Event (super type), Gateway (super type)*

Table 1. Comparison of analysis results. Extracted from [13] and [18], respectively

Another interesting deficiency of BPMN is the lack of means to describe some of the data patterns. In particular, data interaction to and from multiple instances tasks (DP12 and DP13) cannot comprehensively be described, which is mostly credited to a lack of attributes in the specification of the language constructs. This finding aligns with the BWW-based finding that BPMN lacks mechanisms to describe properties, especially property types that *emerge* or are *mutual* due to couplings of things, or those that characterize a component thing of a composite thing (*hereditary*).

Furthermore, the Workflow Pattern analysis reveals deficiency in BPMN's support for the majority of the resource patterns. This finding can also be supported by the BWW-based analysis that found that the constructs in BPMN dedicated to modeling an organizational perspective, *viz.*, Lane and Pool, are considerably unclear in their specification (see next paragraph). Hence it appears that a language specification containing unclear definitions on a construct level lead to deficiencies in composing these constructs to meaningful sets of constructs.

Indistinguishability The Workflow Pattern-based evaluation reveals that while BPMN is capable of expressing all basic control-flow patterns (CP1-5), it contains multiple representations for them, thereby potentially causing confusion as to which representation for a pattern is most appropriate in a given scenario. This aligns with the finding in [18] that BPMN contains a reasonably high degree of construct overload. Especially, in terms of modeling essential concepts of process modeling, such as things, events and transformation, it appears that BPMN is considerably overloaded. This complements the finding that the modeling of the most basic workflow patterns is doubled and thereby unnecessarily complex.

The BWW-based analysis furthermore reveals that the Lane and Pool constructs are extensively overloaded, allowing for representation of various domain aspects (in the case of the Lane construct for example things, classes of things, systems, kinds of things etc.). This complements the statement from the analysis in [13] that the resource patterns RP1 and RP2 use the same graphical notation, relying mostly on Lanes and Pools, for representing two different patterns.

Equivocality The notion of equivocality reveals an interesting facet in the comparison of the two reference frameworks. The findings from the two frameworks do not seem to match with each other. As an example, the control flow patterns 9, 12, 13 and 14 were found to use the same graphical notation, with the differences between the solutions for these patterns only readable from the attribute settings. These solutions rely on the Multiple Instances construct, which the BWW-based analysis classified as overplus. On the other hand, the BWW-based analysis proposes that the different event types in BPMN are redundant. This finding, however, is not supported by the Workflow Pattern-based analysis. It is interesting to note that our own empirical findings related to the BWW-based analysis have, in fact, led to the conclusion that BPMN's differentiation of event constructs has been perceived as very helpful for modeling by BPMN users [18].

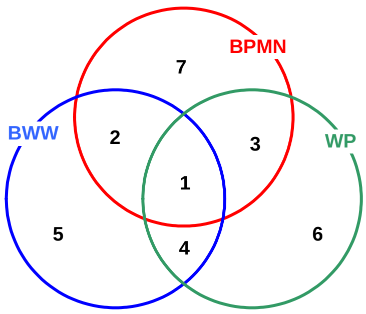
Overplus The perspective of language overplus denotes yet another interesting comparison aspect. It proposes that the Workflow Patterns framework can be used as a means of reasoning for explaining why a particular language contains some constructs that, from a Representation Theory perspective, seem to be unnecessary for capturing domain semantics. In particular, throughout the whole process modeling domain, control flow mechanisms such as logical connectors, selectors, gateways and the like are repeatedly proposed as overplus as they do not map to any construct of the BWW model [12]. However, the Workflow Patterns framework suggests that these constructs nevertheless are central to modeling control-flow, based on the understanding that these mechanisms essentially support the notion of being "in between" states or activities [9].

It must be stated that the Workflow Patterns framework so far has not been used to identify potential overplus of workflow patterns that may be supported in a given language. However, in principle it is possible to apply overplus analysis to the framework for a limited number of language construct involved in a model chunk. It may even be worthwhile investigating how language constructs that the BWW representation model considers as overplus may, in composition, constitute patterns of workflows that have not yet been identified. In this regard the BWW analysis appears to extend the scope of the Workflow Patterns analysis.

4.2 Synopsis

While in the previous section we used the case of BPMN to discuss the complementary and/or substitutive nature of the two reference frameworks under observation, in this section we seek to establish similarities and differences between statements derivable from the analyses of process modeling languages based on different reference framework in a more general fashion.

Fig. 3 presents a simple set model that illustrates potential relationships between two reference frameworks (Representation Theory *BWW* and Workflow Patterns *WP*) and the modeling language under observation (*BPMN*)[5]. Note that in the following we will abstract from the specific relationship types (1:1, 1:0, 0:1, 1:m, m:1) that may occur in a mapping.



Main sets	
BWW	Domain specification as per BWW representation model
WP	Domain specification as per Workflow Patterns framework
BPMN	A BPMN model of the domain

	Subsets						
Property	1	2	3	4	5	6	7
Is described in the BWW model	✓	✓		✓	✓		
Is not described in the BWW model			✓			✓	✓
Is described in the WP model	✓		✓	✓		✓	
Is not described in the WP model		✓			✓		✓
Can be expressed in BPMN	✓	✓	✓				✓
Cannot be expressed in BPMN				✓	✓	✓	

Fig. 3. Set model showing relationships between reference frameworks and modeling language

From Fig. 3 it can be observed that seven hypothetical constellations may in principle occur.

- A set of constructs is provided by both of the reference frameworks and it is found that the modeling language is able to express this set of constructs (subset 1).

[5] We use these indications merely to illustrate our point. The approach itself is in principle applicable to any given combination of two (or even more) reference frameworks and a modeling language.

- A set of constructs is provided by only one of the reference frameworks and it is found that the modeling language is able to express this set of constructs (subsets 2 and 3, respectively).
- A set of constructs is provided by both of the reference frameworks and it is found that the modeling language is not able to express this set of constructs (subset 4).
- A set of constructs is provided by only one of the reference frameworks and it is found that the modeling language is not able to express this set of constructs (subsets 5 and 6, respectively).
- A set of constructs is not provided by any of the reference frameworks but it is found that the modeling language is able to express this set of constructs (subset 7).

Besides the fact that the set model given in Fig. 3 allows for the specification of a ranking of constellations that may occur in the evaluation of modeling languages (*e.g.*, a mapping to subset 1 is of higher relevance than a mapping to subset 3), it also allows us to conclude about the comparison and assessment of modeling languages and reference frameworks in general. As shown, language evaluation by means of reference frameworks has two facets. On the one side, reference frameworks provide a filtering lens that facilitates insights into potential issues with a modeling language. On the other side, any evaluation is restricted to that lens, only exploring potential issues of a language in light of the selected framework. A comparative assessment of such reference frameworks using the case of a single language then can have multiple facets.

It can be used to strengthen the findings obtained from an individual evaluation by identifying complementary statements derived from the analyses. For instance, the finding that BPMN lacks support for the majority of control-flow patterns in the cluster state-based patterns (CP16-18) aligns with the finding that BPMN lacks means for representing states assumed by things (subset 1 in Fig. 3). It can, on the other hand, also be used to identify facets of a given reference framework that extend the scope of another, thereby increasing the focus of an evaluation and overcoming the restricting filter of a single framework. As an example, while the BWW-based evaluation of BPMN shows that BPMN does not contain a single construct that is unambiguously equivalent to any construct of the BWW model, the Workflow Patterns-based analysis reveals that the (potentially ambiguous) atomic BPMN constructs can be arranged to a set of constructs that, as a set, unequivocally equals a number of workflow patterns (subset 3 in Fig. 3). Or, the BWW-based evaluation classifies BPMN connector types as an overplus unnecessary to model IS domains. The Workflow Patterns-based analysis on the other hand suggests that the connector types are in fact essential for the description of control-flow convergence and divergence. However, as subset 7 in Fig. 3 indicates, there may be aspects of a modeling language that are not found to map to any aspect of any of the reference framework used. This scenario can lead to two findings. On first sight, such aspects of a modeling language may in fact unnecessary, ambiguous and/or potentially confusing for modeling the given domain and their usage should therefore be avoided or at least better specified. On the other hand, such a finding may also contribute to the further development of the selected theoretical bases as it indicates that the reference frameworks in use potentially lack relevance or scope for the given domain and thus should be refined or extended. For instance, in an earlier study we discussed the potential lack of relevance of the BWW model for the domain of process modeling [12]. In the case of the Workflow Patterns framework it can by no means be guaranteed that the identified set of patterns is complete.

This brief discussion indicates a need for researchers to carefully observe and scrutinize the findings they derive from their evaluations with respect to the extent to which their findings are rooted in an actual shortcoming of the artifact being evaluated or in a limitation of the selected theoretical reference framework(s) used for the evaluation.

5 Contributions & Future Research

This paper presented the first comprehensive study that compares evaluation frameworks for process modeling languages based on an analysis of the principles of language evaluation.

We do not consider this research complete. In particular, we look to further extend our assessment of evaluation frameworks to incorporate other levels of analysis such as the ones identified in [20]. In particular, we seek to use the principles of presupposition analysis in order to establish differences between evaluation approaches that are imposed by underlying paradigms, for instance in terms of methodology (inductive vs. deductive) or epistemology (constructionist vs. realist). Also, we seek to further populate our set model given in Fig.r̃effigthree by comparatively assessing the findings from the evaluations of other process modeling languages such as BPEL4WS (evaluated in [27] and [28], respectively).

In spite of some limitations of our study, *e.g.*, we have not obtained an empirical perspective towards our evaluation and we have not fully taken into consideration the differences in terms of analysis granularity (atomic notation elements versus compositions of notation elements), we see first evidence for the usefulness of our approach. Our research hopefully motivates practitioners and researchers to converge (rather than diverge) their use of theoretical bases for process modeling. A combination of the principles of both Representation Theory (for the specification of the language constructs) and the Workflow Patterns framework (for the specification of the relationships of language constructs to form meaningful composites) may ultimately lead to the design of process modeling languages that not only provide complete and clear descriptions of real-world domains but also provide sophisticated support for advanced workflow concepts.

We further see potential of generalizing our research to related domains. While our comparative assessment was restricted to process modeling languages and reference frameworks for process modeling languages, we spent considerable effort on defining a generic analysis level that allows for wider uptake. For instance, our research might motivate other researchers to conduct a similar study on reference frameworks for data or object-oriented modeling languages.

Acknowledgement. The authors would like to express their gratitude towards the fruitful contributions of Marta Indulska and Peter Green from the University of Queensland to the evaluation of BPMN by means of Representation Theory.

References

1. Group, G.: Delivering ITs Contribution: The 2005 CIO Agenda. Gartner EXP Premier Reports January2005, Gartner, Inc (2005)
2. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do Practitioners Use Conceptual Modeling in Practice? Data & Knowledge Engineering **58** (2006) 358–380
3. BPMI.org, OMG: Business Process Modeling Notation Specification. Final Adopted Specification (2006)
4. van der Aalst, W.M.P.: Don't Go with the Flow: Web Services Composition Standards Exposed. IEEE Intelligent Systems **18** (2003) 72–76
5. Moody, D.L.: Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. Data & Knowledge Engineering **15** (2005) 243–276
6. Wand, Y., Weber, R.: An Ontological Model of an Information System. IEEE Transactions on Software Engineering **16** (1990) 1282–1292
7. Wand, Y., Weber, R.: On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Journal of Information Systems **3** (1993) 217–237
8. Wand, Y., Weber, R.: On the Deep Structure of Information Systems. Information Systems Journal **5** (1995) 203–223
9. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases **14** (2003) 5–51
10. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. In Pastor, O., Falcao e Cunha, J., eds.: Advanced Information Systems Engineering - CAiSE 2005. Volume 3520 of Lecture Notes in Computer Science. Springer, Porto, Portugal (2005) 216–232
11. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns: Identification, Representation and Tool Support. In Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O., eds.: Conceptual Modeling - ER 2005. Volume 3716 of Lecture Notes in Computer Science. Springer, Klagenfurt, Austria (2005) 353–368
12. Rosemann, M., Recker, J., Indulska, M., Green, P.: A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. In Dubois, E., Pohl, K., eds.: Advanced Information Systems Engineering - CAiSE 2006. Volume 4001 of Lecture Notes in Computer Science. Springer, Luxembourg, Grand-Duchy of Luxembourg (2006) 447–461
13. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: On the Suitability of BPMN for Business Process Modelling. In: 4th International Conference on Business Process Management, Vienna, Austria, Springer (2006) forthcoming

14. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software **11** (1994) 42–49
15. Krogstie, J., Sindre, G., Jørgensen, H.D.: Process Models Representing Knowledge for Action: a Revised Quality Framework. European Journal of Information Systems **15** (2006) 91–102
16. Wahl, T., Sindre, G.: An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. In Castro, J., Teniente, E., eds.: CAiSE'05 Workshops. Volume 1. FEUP, Porto, Portugal (2005) 533–544
17. Nysetvold, A.G., Krogstie, J.: Assessing Business Process Modeling Languages Using a Generic Quality Framework. In Castro, J., Teniente, E., eds.: CAiSE'05 Workshops. Volume 1. FEUP, Porto (2005) 545–556
18. Recker, J., Indulska, M., Rosemann, M., Green, P.: How Good is BPMN Really? Insights from Theory and Practice. In Ljungberg, J., Andersson, M., eds.: 14th European Conference on Information Systems, Goeteborg, Sweden (2006)
19. Siau, K., Rossi, M.: Evaluation of Information Modeling Methods – A Review. In Dolk, D., ed.: 31st Hawaii International Conference on System Sciences, Big Island, Hawaii, Computer Society Press (1998) 314–322
20. Recker, J.: Evaluation of Conceptual Modeling Languages. An Epistemological Discussion. In Romano, N.C., ed.: 11th Americas Conference on Information Systems, Omaha, Nebraska, Association for Information Systems (2005) 329–337
21. Siau, K.: Informational and Computational Equivalence in Comparing Information Modeling Methods. Journal of Database Management **15** (2004) 73–86
22. Floyd, C.: A Comparative Evaluation of System Development Methods. In Olle, T.W., Sol, H.G., Verrijn-Stuart, A.A., eds.: Information System Design Methodologies: Improving the Practice. North-Holland, Amsterdam, The Netherlands (1986) 19–54
23. Weber, R.: Ontological Foundations of Information Systems. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia (1997)
24. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Volume 015 of Telematica Instituut Fundamental Research Series. Telematica Instituut, Enschede, The Netherlands (2005)
25. Gurr, C.A.: Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues. Journal of Visual Languages and Computing **10** (1999) 317–342
26. Bunge, M.A.: Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World. Kluwer Academic Publishers, Dordrecht, The Netherlands (1977)
27. Green, P., Rosemann, M., Indulska, M., Manning, C.: Candidate Interoperability Standards: An Ontological Overlap Analysis. Technical report, University of Queensland (2004)
28. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Analysis of Web Services Composition Languages: The Case of BPEL4WS. In Song, I.Y., Liddle, S.W., Ling, T.W., Scheuermann, P., eds.: Conceptual Modeling - ER 2003. Volume 2813 of Lecture Notes in Computer Science. Springer, Chicago, Illinois (2003) 200–215