Communications of the Association for Information Systems

Volume 20 Article 48

12-14-2007

Ontology- Versus Pattern-Based Evaluation of Process Modeling Languages: A Comparison

Jan Recker
Queensland University of Technology, j.recker@qut.edu.au

Michael Rosemann
Queensland University of Technology, m.rosemann@qut.edu.au

John Krogstie
Norwegian University of Science and Technology

Recommended Citation

Recker, Jan; Rosemann, Michael; and Krogstie, John (2007) "Ontology- Versus Pattern-Based Evaluation of Process Modeling Languages: A Comparison," *Communications of the Association for Information Systems*: Vol. 20 , Article 48.
DOI: 10.17705/1CAIS.02048
Available at: https://aisel.aisnet.org/cais/vol20/iss1/48

Communications of the Association for Information Systems

ONTOLOGY- VERSUS PATTERN-BASED EVALUATION OF PROCESS MODELING LANGUAGES: A COMPARISON

Jan Recker
Business Process Management Cluster
Queensland University of Technology
j.recker@qut.edu.au

Michael Rosemann
Business Process Management Cluster
Queensland University of Technology

John Krogstie
Department of Computer and Information Science
Norwegian University of Science and Technology

ABSTRACT

Selecting an appropriate process modeling language forms an important task for organizations engaging in business process management initiatives. A plethora of process modeling languages has been developed over the last decades, leading to a need for rigorous theory to assist in the evaluation and comparison of the capabilities of these languages. While substantial academic progress in the area of process modeling language evaluation has been made in at least two areas, using an ontology-based theory of representation or the framework of 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 corroborate 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 comparatively assessed in this paper. We also show which tenets of modeling quality these frameworks address and that further research is needed, especially in the area of evaluating the pragmatic quality of modeling.

Keywords: process modeling, Bunge-Wand-Weber representation model, workflow patterns, SEQUAL, model quality

I. INTRODUCTION

The increased popularity of process modeling in IS and BPM practice over the past few years [Davies et al. 2006] has put quite a burden on organizations seeking to engage in process management initiatives. In order to reply to the increasing market demand for business and technical analysts equipped with process modeling skills, a range of interesting questions have to be answered by academia and practice: (1) Which process modeling language should be taught in tertiary educational institutions in order to account for the market demand of graduates being skilled in process modeling? (2) Which process modeling language should a vendor of a BPM tool support, or should a vendor even create yet another language—and what are the implications of making such a decision? (3) Which process modeling language should an organization strive to adopt and implement? These questions have massive economic impact. Amongst others, setting on the "false" process modeling language may lead to significant expenditures on tool licensing and training, and the ultimate failure of the BPM initiative.

As of today, the process modeling discipline has been coined by fragmentation in the choice of languages used for teaching, tools, and practice. The range of languages available spans simple flowcharting techniques, languages initially used as part of requirements engineering such as UML, dedicated business-oriented modeling languages such as event-driven process chains and also formalized and academically studied languages such as Petri nets and their dialects. Consequently, a competitive market is providing a large selection of languages and tools for process modeling, significant demand has been created for means to evaluate and compare the available set of languages and almost every educational institute offers process modeling courses focusing on different languages.

The overall proliferation of process modeling languages has led to an increased need for rigorous theory to assist in the evaluation and comparison of these languages. Van der Aalst [2003] points out that many of the available "standards" for process and workflow specification lack critical evaluation. Along similar lines, Moody [2005] states a concern about lacking evaluation research in the field of conceptual modeling of the dynamics (i.e., the involved processes) of information systems and related phenomena.

In fact, the large selection of currently available process modeling languages and the ongoing efforts in developing these languages stand in sharp contrast to the paucity of evaluation frameworks that can be used for the task of evaluating and comparing those modeling languages in a rigorous manner. There is unfortunately 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 and resulting models, available tool support). However, reasonably mature research has emerged over the last decade with a focus on the representational capabilities and expressive power of process modeling languages. Two examples, the **ontology-based theory of representation** [Wand and Weber1990, 1993, 1995; Weber 1997] and the **workflow patterns framework** [van der Aalst et al*.* 2003, 2005a; Russell et al*.* 2005b, 2006a, 2006b] 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? Are their results comparable? What types of insights into expressive power and shortcomings of a process modeling language can be obtained from them? Does the joint application of both frameworks cover all relevant criteria of a complete evaluation? These and related questions can be traced back to Moody's [2005] argument that the observable proliferation of different quality measurement proposals in the field of conceptual modeling is in fact counterproductive to research progress; indeed, the existence of multiple competing proposals is rather an indicator for 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 modeling quality [Moody, 2005, p. 258].

Taking together the ongoing proliferation of prospective 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:

1. We introduce a generic framework for language evaluation and apply it to both representation theory and workflow patterns framework in order to establish commonalities and differences between these two quality proposals.
2. We use the example of the most recent and prominent candidate for an industry standard language for process modeling, BPMN [BPMI.org and OMG 2006], as a language that is

evaluated by both frameworks. Thereby we are able to show how to integrate the analyses of BPMN and give a comprehensive picture of its capabilities and shortcomings.

We proceed as follows. First we briefly introduce our selected unit of analysis, BPMN, and discuss studies related to our research (Section II). We then establish a generic framework for language evaluation and apply it to the frameworks in question (Section III). Section IV briefly describes how the individual analyses of BPMN were carried out and then presents our assessment of the two frameworks, and finally compares the individual analyses of BPMN. We close in Section V by summarizing our work, identifying contributions and implications for theory and practice, discussing the limitations of our work, and outlining future research opportunities.

II. BACKGROUND AND RELATED WORK

INTRODUCTION TO BPMN

In this section, we briefly introduce BPMN in order to give the reader sufficient background for understanding our subsequent argumentations.

BPMN has over the last years been propelled as the most prominent candidate for an industry standard in process modeling, similar to the example of the UML notation in software engineering. BPMN was originally developed by the Business Process Management Initiative BPMI.org. Its specification 1.0 was released in May 2004 and adopted by OMG for standardization purposes in February 2006 [BPMI.org and OMG 2006]. The development of BPMN was based on the revision of other notations, including UML, IDEF, ebXML, RosettaNet, LOVeM and EPCs, and stemmed from the demand for a graphical language that complements the BPEL 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 language that can be applied for typical business modeling activities as well. The complete BPMN specification defines thirty-eight distinct 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 interconnect 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. Figure 1 gives an example of a BPMN model that shows a payment process in which customers can pay via cash, check or credit card. Refer to OMG’s specification [BPMI.org and OMG 2006] for further information on BPMN.

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 frameworks for conceptual models. We briefly recapitulate the related work in this section and how it contrasts to the work presented in this paper. Where appropriate, we will refer to selected related work in the later sections of this paper.

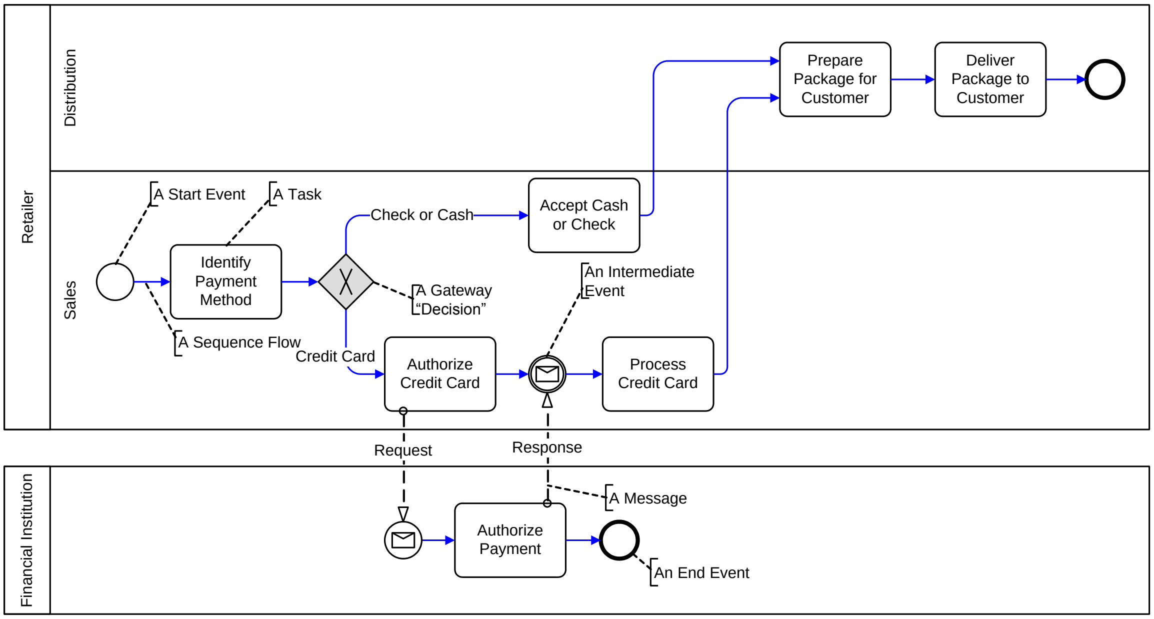


Figure 1. BPMN Model of a Payment Process

Over the last years, at least two promising proposals for a quality framework for process modeling languages have emerged, viz., the Wand and Weber's [1990, 1993, 1995] theory of representation (in short: the BWW theory) and the workflow patterns framework [van der Aalst et al. 2003; Russell et al. 2006a]. Both proposals will be discussed in detail in Section III of this paper.

Besides these two established proposals it is required to mention the semiotic quality framework [Lindland et al. 1994], which is a well-discussed framework for evaluating the quality of conceptual modeling in general. However, it has so far only sparingly been applied to the domain of process modeling (e.g., [Krogstie et al. 2006b]). The framework is based on linguistic and semiotic concepts (such as syntax, semantics and pragmatics) that enable the assertion of quality at different levels:

- *Syntax* relates the model to the modeling language by describing relations among language constructs without considering their meaning.
- *Semantics* relates the model to the domain by considering relations among statements and their meaning.
- *Pragmatics r*elates the model to audience participation by considering not only syntax and semantics, but also how the audience (anyone involved in modeling) will interpret and apply them.

The ontology- and pattern-based evaluation frameworks discussed in this paper focus on the expressiveness of a process modeling language. In the work of Lindland et al., this aspect belongs to the question of how to support the achievement of semantic quality and is denoted as domain appropriateness,[1] which, in the general framework, is specified on a level of high

1 This deals with how suitable a language is for use within different domains. If there are no statements in the domain that cannot be expressed in the language, then the language has good domain appropriateness. In addition you should not be able to express statements that are not in the domain [Krogstie et al*.* 2006b].

abstraction. As indicated in earlier work on the semiotic quality framework, e.g., [Krogstie and Jørgensen 2003; Wahl and Sindre 2006], using an ontology-based evaluation approach such as the BWW theory (or a similar reference system such as the workflow patterns framework) is one of several possible ways of devising concrete criteria for domain appropriateness.

Having established that ontology- and pattern-based evaluation reference systems for process modeling languages operate on a semantic level of model quality, Lindland et al.'s framework can identify areas of quality that are not being addressed by any of these two frameworks:

Neither of the two frameworks explicitly addresses aspects of the syntactical quality of process modeling languages, i.e., the goodness of the formal laws that constitute the grammar rules by which models are being created. This is neither surprising nor of concern. A number of authors have provided sufficient means for assessing syntax-related aspects of process modeling, e.g., [ter Hofstede and van der Weide 1992; van der Aalst 1999; Kiepuszewski et al. 2003].

The pragmatic criterion is concerned with the compliance of the model to the aims and purposes for which the model was created. This dimension is concerned with assessing the value of the process model for helping its audience to better cope with their problems of, for example, introducing process-aligned organizational structures, designing executable workflow specifications or solving process improvement tasks.

Lindland et al. distinguish in their framework between technical actor interpretation and social actor interpretation [see also Krogstie et al. 2006b]. Social actor interpretation concerns how the model is being received by a (human) audience while technical actor interpretation concerns how the model is being received by an information system. In its essence, these two facets of pragmatic quality address the two major purposes of process modeling [Dehnert and van der Aalst 2004]:

- Intuitive business process models are created for the sake of providing a basis for communication between relevant stakeholders, for instance, for scoping process improvement projects or capturing and discussing business requirements. As such, they must be understandable, extendable, should be intuitive and interpretable to facilitate discussion and agreement.
- Formal business process models are created for the sake of process automation, which requires them to be machine-readable. They are used as input to process enactment systems and hence must be unambiguous, should not contain any uncertainties and should also feature implementation information.

Following this differentiation it becomes clear how the pattern- and ontology-based frameworks relate to each other. The workflow patterns framework has been developed to delineate the fundamental requirements that arise during business process modeling for the selection, design and development of workflow systems [van der Aalst et al. 2003]. Hence, business process modeling languages are evaluated in light of one pragmatic aspect, to facilitate the specification of executable workflow input to process enactment systems. In Lindland et al.'s framework, this purpose addresses the pragmatic quality aspect "technical actor interpretation." The ontology-based BWW theory addresses a different pragmatic aspect of modeling: the goodness of a representation of real-world domains for the purpose of enabling communication between involved stakeholders (such as developers and users, business analysts and system designers etc.) and documenting business requirements [Siau 2004]. Hence, business process modeling languages are evaluated with respect to the quality of the representation of aspects of real-world domains and how well these representations enable domain understanding [Gemino and Wand 2005]. As such, the important aspect is here that process models be understandable to stakeholders, analysts, and designers. As such, the corresponding pragmatic aspect in Lindland et al.'s framework would be "social actor interpretation."

Yet another perspective on process modeling quality is provided by Hepp and Roman [2007] who discuss the various traits of process modeling (e.g., model sources, modeling motivations, modeling requirements etc.) that need to be taken in consideration. Their work suggests a set of ontologies to define the fundamental notions relevant to process modeling, such as orchestration, organization and resources, function, data, strategy, business logics as well as provision and consumption. Their work indicates that in the area of process modeling, several dimensions exist that are at current only poorly supported by available languages, and also only insufficiently incorporated in evaluation frameworks. In light of their SBPM framework, the work presented in this paper concerns evaluation frameworks that focus on the dimensions of orchestration, data, function as well as organization and resources.

Though the SBPM and the semiotic quality framework provide good examples of quality management proposals for process modeling, it remains unclear how other frameworks could be used, in isolation or combination, to address aspects of process modeling quality that the arbiters of the semiotic quality or the SBPM framework feel insufficiently addressed. The work presented in this paper addresses this gap of knowledge by discussing explicitly how the two most prominent evaluation frameworks (representation theory and the workflow patterns framework) compare to each other.

Our work presents the first contribution towards a critical, comparative appraisal of the workflow patterns framework and the BWW theory and also presents the first work that comparatively assesses different modeling quality proposals by using a specific unit of analysis, viz., a particular process modeling language.

III. EVALUATING PROCESS MODELING LANGUAGES

A GENERIC 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 language evaluation research. The purpose of the current section is to define a framework for language evaluation under which existing approaches can be subsumed. This will allow us to comparatively assess the two selected frameworks.

In order to establish this framework we refer back to one of the generally acknowledged objectives of process modeling, which is to build a (predominantly graphical) representation of a selected set of domain operations for the purpose of understanding and communication among stakeholders in the process of requirements engineering for process-aware information systems [Dumas et al. 2005].[2] Process modeling languages are used to compose graphical models that convey information about a domain or system in such a form that it not only enables easy interpretation, but moreover denotes a useful means for communication and understanding.

The stakeholders involved are typically 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 [Wyssusek 2006]. As such underlying conceptual structures are dependant on, inter alia, modeler, model audience and modeling purpose, they cannot be equated for all involved stakeholders, but merely denote *potentially* valid modeling references that hold true in some 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

---

2 We acknowledge that also other purposes exist for conceptual modeling, such as providing input to systems design, model execution (e.g., in connection to automated workflow) or documenting user requirements for future reference. Yet, we argue that it is foremost the objective of enabling communication amongst relevant stakeholders that applies to process modeling, which is the reason we focus on this purpose in our elaborations.

selected framework. The underlying assumption here is that modeling languages should be similar to the conceptualization of the domain of interest in the form of the modeling reference framework so as to facilitate adequate communication with the resulting model. Figure 2 explicates these relations.

According to Figure 2, a modeling reference framework, such as the BWW representation model or the workflow patterns framework, can be used as a universal, general 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, resources, and exception handling. 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 in use 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 [Frank 1999]. The question that arises here is that if there are more than one of those universal reference systems for conceptual modeling (e.g., ontology-based systems versus pattern-based systems), how is one to decide which system is better than others in conveying a good representation by any modeling language [Lyytinen 2006]?

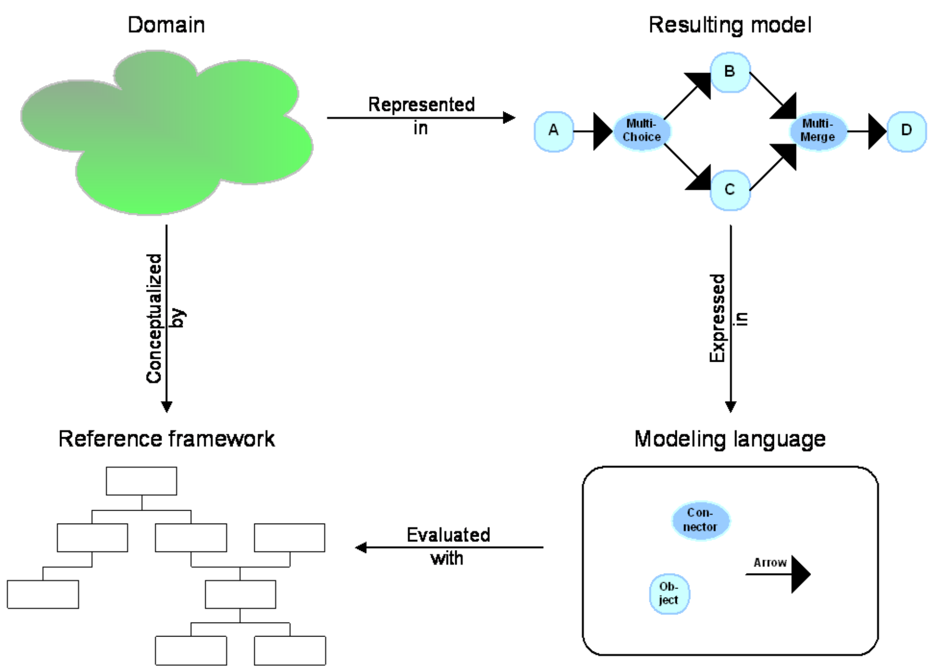


Figure 2. Relations between Domain, Reference Framework, Modeling Language and Model

The process of evaluating modeling languages against a reference framework consists of a pair-wise bi-directional mapping between the concepts specified in the reference framework against the symbolic constructs specified in the modeling language. For example, the workflow patterns framework assesses which of the specified patterns (with a pattern being a set of meaningfully composed constructs) can be expressed with a given language. The basic assumption is usually 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. This assumption rests on the observation that if the selected reference framework for modeling denotes a valid conceptualization of the domain of interest, then a modeling language should neither express fewer aspects than conveyed in the reference framework, nor more aspects, nor the given domain aspects in an ambiguous or redundant way.

Following this argumentation, formally, the relationships between what can be represented (the set of semantics, i.e., the constructs, of the modeling language) and what is represented (the set of semantics, i.e., the concepts, of the reference framework as a heuristic for the domain being modeled) can be specified in a generic framework for language evaluation that differentiates five types of relationships that may occur in the bi-directional evaluation of modeling languages against reference frameworks (see Figure 3).

- *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).
- *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).

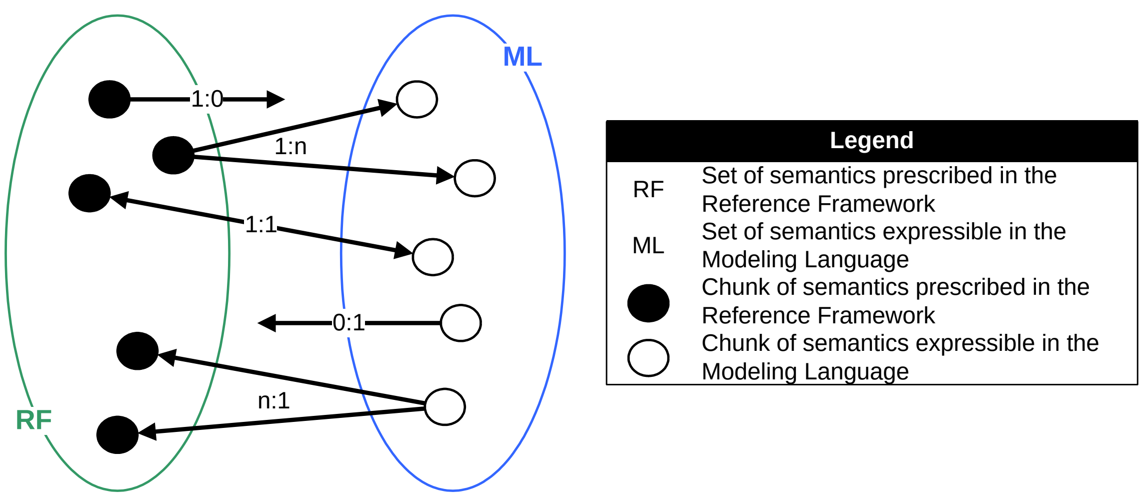


Figure 3. Framework for Language Evaluation

The framework for language evaluation presented in Figure 3 draws on previous work in related disciplines. Weber [1997] for instance uses a similar albeit not identical framework to explain the two situations of ontological completeness and clarity of a language, Guizzardi [2005] argues in a similar fashion in the context of structural specifications, and Gurr [1999] uses similar mapping relations to analyze diagrammatic communication. It should be noted that these three authors use their frameworks for language evaluation while we propose to use the framework depicted in Figure 3 on a *meta level*, i.e., to evaluate the evaluation framework themselves. Hence, we build upon their work to explain *in general* the research type of language evaluation.

Having defined hypothetical relationships that may occur in a pair-wise bi-directional 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 that forms the core of representation theory, and the workflow patterns framework as indications for available reference frameworks in the domain of process modeling.

Our selection of the Bunge-Wand-Weber representation model was motivated by the maturity of the theory and the widespread adoption of this model not only in conceptual modeling research [Weber and Zhang 1996; Opdahl and Henderson-Sellers 2002; Shanks et al. 2003; Gemino and Wand 2005] but also in the area of process modeling, for instance in the evaluation of Petri Nets [Recker and Indulska 2007], EPCs [Green and Rosemann 2000], ebXML [Green et al. 2005], BPEL [Green et al. 2007] and others. A comprehensive annotated overview is given in [Rosemann et al. 2006]. Our selection can further be justified in referral to the large number of empirical tests on basis of this model that were undertaken in the past, e.g., [Bodart et al. 2001; Green and Rosemann 2001; Gemino and Wand 2005; Bowen et al. 2006].

Similar to the case of the BWW representation model, the workflow patterns framework has been widely used both as a benchmark for analysis and comparison of process modeling languages (e.g., UML 2.0 Activity Diagrams [Russell et al. 2006c]), Web services composition languages (e.g., BPEL [Wohed *et al.* 2003b]) and languages for enterprise application integration (e.g., BML [Wohed *et al.* 2003a]). A comprehensive annotated overview is given on www.workflowpatterns.com. Our choice of the workflow patterns framework as a second analysis framework in our study was motivated by several factors. First, it is a well accepted framework that has been widely used both for the selection of workflow management systems (e.g., by UWV, the Dutch Justice Department, ArboNed, etc.) as well as for vendors' self-evaluations of process modeling products (e.g., COSA, FLOWer, Staffware, IBM, etc.). Second, this framework has proven impact in the industry. It has triggered extensions to process modeling systems (e.g., FLOWer 3.0, Staffware Process Suite, Pectra Technology Inc.'s tool) and inspired their development (e.g., OpenWFE, Zebra, Alphaflow).

EXISTING FRAMEWORKS FOR EVALUATING PROCESS MODELING LANGUAGES

The Bunge-Wand-Weber Representation Model

The development of the *representation theory* 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 [1990, 1993, 1995] suggest that ontology may help define and build information systems that faithfully represent real world systems. Ontology is a well-established theoretical domain within philosophy that deals with identifying and understanding elements of the real world [Bunge 2003]. Wand and Weber adopted an ontology defined by Bunge [1977] and from this derived a theory of representation for the Information Systems discipline that became widely known as the Bunge-Wand-Weber (BWW) representation model. Following Wand and Weber's arguments, models of information systems and thus their underlying modeling language should 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 and thus represent information systems: things including properties and types of things; states assumed by things; events and transformations occurring on things; and systems structured around things [Rosemann and Green 2002].

Wand and Weber's work based on Bunge's theory is not the only case of ontology-based research on conceptual modeling. The approaches of Milton and Kazmierczak [2004] and Guizzardi [2005] are closest to the ideas of Wand and Weber. These upper-level ontologies have been built for similar purposes and appear to be equally expressive [Davies *et al.* 2005] but have not yet achieved the popularity and dissemination of the BWW model. As our related work section shows, the BWW model has in several instances also been shown to deliver fruitful insights into the capabilities and shortcomings of process modeling languages, e.g., [Rosemann *et al.* 2006].

Generally speaking, 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. Referring to the five types of relations specified previously, the completeness of a description can be measured by the degree of *construct deficit*, *i.e.*, deficiency (see Figure 3). The clarity of a description can be measured by the degrees of *construct overload*, *i.e.*, equivocality (see Figure 3), *construct redundancy*, *i.e.*, indistinguishability (see Figure 3), and *construct excess*, *i.e.*, overplus (see Figure 3). 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 (see Figure 3) of a modeling language.

The Workflow Patterns Framework

In contrast to ontology-based research on process modeling languages, a second reference system for process modeling emerged over the last years, which built upon the use of patterns as they have been used in architecture or software engineering. 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 initial investigation 20 *control-flow patterns* [van der Aalst *et al.* 2003] were 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 constructs (e.g., parallel split) to complex control-flow scenarios (e.g., multiple instances without synchronization) and provide a taxonomy for the control-flow perspective of processes.

In 2005, the workflow patterns work was extended to also analyze constructs for the data [Russell *et al.* 2005a] and the resource perspectives of workflows [Russell *et al.* 2005b]. 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. The resource perspective further complements the approach by describing the various ways in which work is distributed amongst and managed by the resources associated with a business process. During the same year also the area of workflow exception handling was investigated, which resulted in the identification of a set of *exception handling patterns* [Russell *et al.* 2006b] systematizing the various mechanisms for dealing with exceptions occurring in the control-flow, the data or the resource perspectives.[3]

Referring back to the five types of relations specified in Figure 3, evaluations (such as the ones reported in the related work section) using the workflow patterns framework traditionally focus on the identification of potential representations within a given modeling language for each of the patterns (*i.e.*, the 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 could be used to partially reveal some equivocality, it has so far not been used to identify and reason about overplus.[4]

---

[3] Note that in 2006 the work on the Workflow Patterns has progressed with a revision and formalization of the original control-flow patterns [Russell *et al.* 2006a]. The set of the 20 control-flow patters was extended to 43 and every pattern has been formally represented in colored Petri Nets notation. In this paper, however, we refer to the original set of workflow patterns.

[4] Usually, one-to-many correspondences between patterns and primitives in a modeling language exist, which in turn leads to multiple potential representations of a pattern.

IV. COMPARING THE EVALUATION FRAMEWORKS

Based on the elaborations in Section III we argue that it is possible to pair-wise compare the findings from representation theory and workflow patterns analyses using the framework for language evaluation defined in Figure 3. We will in the following use the example of an evaluation of the BPMN language in order to extract similarities and differences in the reference frameworks. This allows us to address all the objectives of this paper, viz., delivering a comprehensive evaluation of the capabilities of BPMN, studying to what extent the two frameworks under observation complement respectively substitute each other, and identifying the areas of modeling quality in which both frameworks require extension and/or revision.

EVALUATION FRAMEWORKS ASSESSMENT

In preparation for this study we have used the two frameworks in question to evaluate BPMN individually. In the interest of brevity we omit an in-depth discussion of the individual analyses and refer to the description of our previous work in [Recker *et al.*, 2006] and [Wohed *et al.*, 2006b].

Each individual analysis followed an established research process to display reliability and validity of the evaluation.

Analysis on basis of the BWW representation model

Our evaluation of BPMN against the BWW representation model followed the procedural model presented by Rosemann et al. [2004]. Their procedural model was developed specifically to countervail potential flaws and ambiguity in this type of analytical research and addresses concerns such as lack of understandability, lack of comparability, lack of completeness, lack of objectivity, lack of guidance and others. More precisely, our analysis was conducted in three steps. First, two researchers separately read the BPMN specification and mapped each of the single BPMN constructs against BWW constructs in order to create individual first analysis drafts.[5] Second, the researchers met to discuss and defend their mapping results. Third, the jointly agreed second draft was discussed and refined in several meetings with the entire research team. By reaching a consensus over the final mapping result we feel that we achieved a maximum of possible objectivity and rigor in this type of research.

Adopting this methodology has also allowed the derivation of agreement statistics between the individual researchers. In order to display inter-judge reliability in the mappings, a raw percentage agreement [Moore and Benbasat, 1991] and Cohen's Kappa [Cohen, 1960] were used to measure the agreement between the mapping researchers. Cohen's Kappa is accepted to be a better measure than a raw percentage agreement calculation, since it also accounts for chance agreement between the researchers. Raw percentage agreement for the representation mapping of BPMN was calculated to be 68.8 percent in the first round and 87.2 percent in the second

---

Analyzing all the possible combinations of primitives (which may be an infinite number) would certainly be insightful but is virtually impossible without automation of the process.

5 At this stage it should be noted that we were restricted in our evaluation to 1:1 mappings between constructs in BPMN and constructs in the BWW representation model. Whilst in general representation theory would allow for the comparison of BWW model constructs to a combination of several language constructs (1:n mappings) or even vice versa—similar to evaluations of the workflow patterns framework type, representational analyses typically are restricted to 1:1 comparisons. All of the previous studies of process modeling languages based on the BWW representation model are restricted to 1:1 mappings [Rosemann *et al.*, forthcoming]. We are aware that this posits a limitation to our study. It would indeed be interesting and challenging to examine how ontologically meaningful clusters of BPMN constructs could be formed. Yet, for brevity reasons we cannot consider the potentially unlimited variety of construct compositions in BPMN in our study.

round while Cohen's Kappa was calculated to be .616 in the first round and .832 in the second round, both of which exceeds generally recommended Kappa levels of .6 [Moore and Benbasat 1991]. In the third round, the mapping was being discussed and refined until a 100 percent agreement across the complete research team was obtained.[6]

Analysis on Basis of the Workflow Patterns Framework

Regarding the workflow patterns analysis, typically, the analysis of a process modeling language against the workflow patterns framework involves an (automatic) comparison of the formal semantics of the language in an execution environment against the workflow patterns as formally defined in a mathematically valid specification language such as Coloured Petri Nets notation. Unfortunately, due to the recency of its release BPMN does neither yet have commonly agreed-upon formal semantics nor an execution environment. Hence, the analysis of BPMN against the workflow patterns framework was performed in a manner similar to the process outlined previously. First, individual analyses of BPMN against the workflow patterns framework were created by members of the research team. These individual were then combined and finally defended and revised before the complete background team until a consensus was obtained. In performing this work, the encountered ambiguities as well as the assumptions made to overcome these were documented in tabular form.[7]

Results and Comparison

We fitted the results of these analyses into Table 1, structured in accordance to the framework for language evaluation (see Figure 3).[8] Subsequently we pair-wise compare the findings derived from each analysis for each of the five mapping relations.

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

---

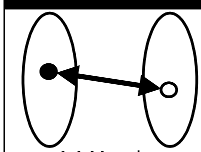
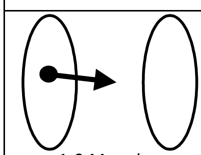
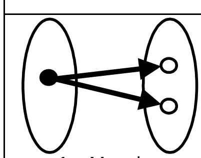
6 Consider this example: in the first, individual mapping round, one researcher classified the BPMN construct "Data Object" as excess. This was reasoned in referral to the BPMN specification [BPMI.org and OMG 2006], which states that the use of this artifact does not affect the other parts of the domain representation contained in the model. Hence it was argued that the Data Object construct does not carry real-world semantics. The other researcher, however, afforded Data Object a mapping to the BWW representation model construct Thing, based on the observation that a Data Object is used to depict information objects, both physical and electronic, and accordingly represents real-world objects such as documents or data records. After discussion and study of specification documents, in the second mapping round both researchers individually revised their mappings. One researcher maintained his mapping of Data Object to Thing while the other mapped it to"Class." This was justified by the observation that a Data Object actually does not model a specific document or data record (such as invoice 47-11) but instead only types of objects (e.g., invoice, policy, customer master record). These two alternative mapping suggestions were presented to, and discussed with, the entire research team that together studied the specification of the constructs and, eventually, agreed to afford the Data Object a mapping to Class. This process was carried out for all other construct mappings.

7 This documentation is available at www.BPMCenter.org or in [Russell *et al.* 2006a, pp. 113-115; Wohed *et al.* 2006b].

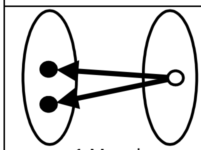
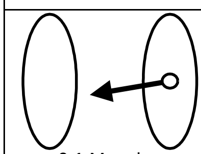
8 Note that in the Workflow Patterns column in Table 1, the acronyms (e.g., CP1, RP14, DP2) refer to the numbers that were given to the different patterns. CP refers to control flow patterns, RP to resource patterns and DP to data patterns.

the interpretation of the resulting model [Recker *et al.*, 2006], 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 is not 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 matter, the workflow patterns framework appears to be an extension in the level of analysis offered by representation theory. 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.

Table 1. Mapping Results

Mapping 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 single 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:[9] *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,*

9 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 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 the language.

Mapping Relation	Workflow Patterns	Representation Theory
		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 at least two 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)*

Deficiency

Table 1 strongly suggests a lack of capabilities in BPMN to model state-related aspects of business processes. Both analyses reveal that BPMN is limited if not incapable of modeling states assumed by things and state-based patterns, 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 BPMN in its capability of modeling state-related semantics.

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 to a large extent credited to the lack of attributes in the specification of the language constructs. This finding aligns with the BWW 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 a deficiency in BPMN's support for the majority of the resource patterns. This finding can also be supported by the BWW-based analysis as it was 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 that BPMN contains a relatively high degree of construct redundancy. Especially, in terms of modeling essential concepts of process modeling, such as things, events and transformation, it appears that BPMN contains a relatively large number of redundant constructs (different forms of activity and event constructs in particular)—which complements the finding that the modeling of the most basic workflow patterns is doubled and thereby unnecessarily complex.

Equivocality

The notion of equivocality reveals an interest facet in the comparison of the two reference frameworks in that the findings from each framework 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. From the graphical model itself, it is thus impossible to identify which distinct process pattern exactly is being represented. This in turn may result in model end user confusion due to unclear semantics.

The BWW analysis reveals that the Lane and Pool constructs as well as a number of event types are extensively overloaded. These constructs allow for the representation of various domain aspects, in the case of the Lane construct for example things, classes of things, systems, kinds of things etc.

These findings are not supported by the workflow pattern-based analysis. The patterns CP4, CP6, CP9, CP12, CP13 and CP14 that were found to have equivocal representations in BPMN do not rely on the Event, Pool or Lane constructs. Here it would appear that the findings from the two analyses contradict each other.

Overplus

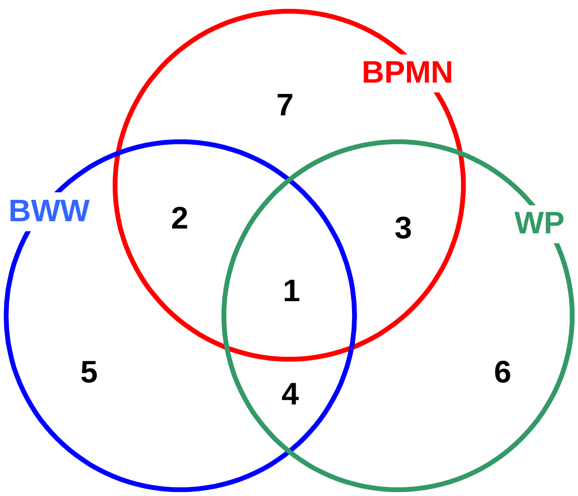
The perspective of language overplus denotes an aspect similar to the case of equivalence in that it proposes that the workflow patterns framework can be used as a means of reasoning for explaining why a particular language contains some constructs that, from a representation theory perspective, seem 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. However, the workflow patterns framework suggests that these constructs nevertheless are central to control-flow modeling based on the understanding that these mechanisms essentially support the notion of being “in between” states or activities [van der Aalst *et al.* 2003].

Aside from this particular aspect, it must be stated that the workflow patterns framework so far has not been used to identify a 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 and 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. This could potentially put an end to the discussion of so-called excess constructs that are frequently found in process modeling languages, see [Rosemann *et al.* 2006]. It may also be an interesting research suggestion to investigate how BWW-based process modeling language primitives may be formed to meaningful sets of workflow patterns.

DISCUSSION

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. In essence, we use the case of the BPMN evaluation to derive conclusions about the nature of the evaluation frameworks themselves.

Figure 4 presents a simple set model that illustrates theoretically possible relationships between two reference frameworks (representation theory *BWW* and workflow patterns *WP*) and the modeling language under observation (*BPMN*). Note here 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 (refer to Figure 3). Note that we use the indications BWW, WP and BPMN 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.



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				✓	✓	✓	

Figure 4. Set Model Showing Relationships between Reference Frameworks and Modeling Language

From Figure 4 it can be observed that seven constellations may in principle occur:

- A set of concepts[10] is provided by both of the reference frameworks and it is found that the modeling language is able to express this set of concepts (subset 1).
- A set of concepts is provided by only one of the reference frameworks and it is found that the modeling language is able to express this set of concepts (subsets 2 and 3, respectively).
- A set of concepts is provided by both of the reference frameworks and it is found that the modeling language is not able to express this set of concepts (subset 4).

---

10 The reference frameworks may in fact prescribe the set of semantics as a set of atomic constructs (as in the case of the BWW representation theory) or as a set of composite constructs (as in the case of the workflow patterns framework). Thus, we refer here to a set of concepts to abstract from the level of granularity employed by any framework.

- A set of concepts 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 concepts (subsets 5 and 6, respectively).

- A set of concepts is not provided by any of the reference frameworks but it is found that the modeling language is able to express this set of concepts (subset 7).

Besides the fact that the basic model given in Figure 4 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 a quality indicator for the language under evaluation whereas a mapping to subset 4 points to a potentially significant issue). It also allows us to conclude about the comparison and assessment of modeling languages and reference frameworks in general.

As has been shown in our evaluation of BPMN, 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 exactly that lens, hence only exploring potential issues of a language *in light* of the selected framework. A comparative assessment of such reference frameworks using the example of a given 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 Figure 4).

- It can be used to identify facets of a given reference framework that extends the scope of another, thereby increasing the focus of an evaluation and overcoming the restricted filter of one given 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) BPMN constructs can nevertheless be arranged to a meaningful set of constructs that, as a set, unequivocally equal a number of workflow patterns (subset 3 in Figure 4). Or, the BWW-based evaluation classifies BPMN connector types as an overplus, i.e., unnecessary to model IS domains; however, the workflow patterns-based analysis suggests that the same connector types, in combination with other constructs, could in fact be meaningful for the description of control flow convergence and divergence. Table 2 gives a summary of where, in the case of BPMN, findings from the representation theory evaluation and the workflow patterns evaluation corroborate, extend or contradict the other. The summary follows the introduced five relationship types as shown in Figure 3.

Table 2 suggests that BWW analysis and workflow pattern analysis are mostly complementary in nature. The findings appear to support each other in most of the cases. If not, differences in the findings were often found to be explained by the divergent range of inquiry, i.e., the scope of the investigation. Consequently, it would appear that the combination of atomic construct level analysis (as per BWW representation model) with a composite construct level analysis (as per workflow patterns framework) is most fruitful for separating “true” deficiencies in a process modeling language from only seemingly valid findings.

The case of contradiction between the findings (in the case of equivocality) poses an interesting proposition to process modeling research, namely whether one or both of the framework are over- or under-engineered. Our suggestion would be to use empirical insights into the actual practice of process modeling as a starting point for further investigation and potential extension or revision of the frameworks. We would like to invite interested colleagues to join in this endeavor.

Table 2. Comparison of Evaluation Findings

Mapping relationship	Key finding	Framework comparison
Equivalence	Only the workflow pattern evaluation identified equivalence.	**Extension** Since representation theory and workflow pattern work on different levels of granularity (atomic language constructs versus construct compositions), it would appear that the workflow pattern framework extends the evaluative capacity of representation theory by indicating how seemingly ambiguous atomic language constructs in BPMN can be arranged to compositions that are meaningful in terms of specifying certain process patterns.
Deficiency	The workflow patterns framework identified deficiencies in BPMN in regards to state-based, data and resource patterns. The BWW-based evaluation suggests deficiencies in modeling properties, states and aspects of systems of things.	**Corroboration** Both representation theory and workflow pattern analysis indicate deficiencies in BPMN. The analysis on a language construct level (by means of the BWW representation model) here suggests that deficiencies in the nature of the language constructs can lead to problems when arranging the atomic constructs to meaningful compositions. A deficiency on construct level thus leads to a diminished capacity to depict all process patterns that may potentially be of relevance.
Indistinguishability	The workflow pattern analysis shows an unnecessary complex representation of basic patterns. The BWW-based analysis shows a redundancy in basic notions such as thing, transformation and event.	**Corroboration** The analysis on a construct level identifies a high level of redundancy in the set of BPMN constructs available. A large number of BPMN constructs share the same representational capacity. This finding supports the findings from the workflow pattern analysis that shows how a number of patterns can be represented in multiple ways, thereby inducing unnecessary complexity in the modeling exercise and potentially causing confusion to the model end user.
Equivocality	Some of the patterns are equivocally modeled in BPMN. However, these patterns do not make use of any equivocal language construct in BPMN, such as Event, Pool or Lane.	**Contradiction** The findings from both analyses do not align. The BWW-based analysis predicts equivocality of some language constructs. However, none of these constructs is involved in the equivocal representation of some patterns. This suggests a lack of relevance or adequacy of one of the frameworks and indicates a need for improving these theoretical bases. The proposition would be to obtain empirical insights into the nature and use of the conflicting constructs and composites and, based on these insights, work on

Mapping relationship	Key finding	Framework comparison
		a revision or extension of the evaluation framework.
Overplus	The BWW-based analysis indicates some superfluous language constructs. These constructs, however, are shown in the workflow pattern analysis to be relevant to the depiction of certain patterns.	**Extension** From an atomic perspective, some constructs (such as control flow mechanisms) appear superfluous. Yet, an analysis on composite level gives a justification for their existence in that it shows how they can be arranged in meaningful compositions of process patterns. This way, the workflow pattern analysis extends the range of representation theory by expanding the scope of analysis to a level of less granularity.

It should further be noted that in addition to our elaborations earlier, there are also other constellations that need to be considered. Subset 7 in Figure 4 indicates that 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:

1. The identified aspects of a modeling language are in fact unnecessary/ambiguous/potentially confusing for modeling the given domain and their usage should therefore be avoided or at least better specified.
2. Such a finding can also contribute to the further development of the selected theoretical bases as it might indicate that the reference frameworks in use might lack relevance or coverage for the given domain and should thus be refined or extended.

For instance, in the case of the workflow patterns framework it can by no means be guaranteed that the identified set of patterns is complete. This 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.

V. CONTRIBUTIONS AND CONCLUSIONS

CONTRIBUTIONS

This paper presents the first comprehensive study that compares the most popular evaluation frameworks for process modeling languages based on a generic framework of the principles of language evaluation. We showed that very fruitful insights on language evaluation and, ultimately, language use can be generated if evaluation reference frameworks are being applied in a complementary rather than substitute manner. We also reported on the first attempt to classify existing theoretical frameworks for process modeling language evaluation by using a generic framework for model quality management.

The contributions of this work relate to both process modeling practice and theory:

Implications for Practice

Although methodological or theoretical/analytical argumentations such as ours often appear far-stretched rather than directly applicable to IS practice, there are arguably observable practical merits. First and foremost we have shown how additional insights into the use of, and potential problems with, a process modeling language can be obtained if multiple frameworks for language evaluation are being applied. Especially in light of the wide range of process modeling languages that have already been evaluated by the two frameworks considered (see Section III of this paper), it can be assumed that organizations will have easy access to benefits at considerable low costs.

These findings are beneficial for organizations currently selecting process modeling languages, a step that is of crucial importance to any business process modeling initiative. Especially as more and more organizations turn to BPM, often in concert with changing to a Service Oriented Architecture (SOA), the choice of modeling approach can have large organizational consequences for a number of years. The evaluation reported in [Nysetvold and Krogstie 2006], for instance, was used as a basis for a choice of modeling language and environment across an enterprise in transition to SOA. Thus it can obviously be cost efficient to perform a rather rigorous evaluation process prior to such change. Second, we have been able to show that the question of process modeling purpose is crucial to the selection of an appropriate reference framework. Practitioners should thus carefully consider the general objective of their process modeling efforts when evaluating and selecting an appropriate process modeling language and, in effect, the comparison and evaluation criteria they employ in such decision making processes.

Implications for Research

We deem our work a fruitful starting point for further research investigations into the nature, and use, of process modeling languages. We have shown that language development and deployment not only should consider semantics of language constructs and semantics of construct compositions but moreover the pragmatics of using the language in real-life modeling scenarios. We see a number of interesting and stimulating research challenges stemming from our work.

First, the workflow patterns framework has, as reported, been derived inductively from observable practice while representation theory builds upon a strong theoretical foundation. We believe that an ontological foundation of the workflow patterns framework could lead to more rigor in the workflow engineering discipline and would also benefit the investigation into the nature of language patterns.

Second, representation theory often is being applied to language evaluation for investigation the semantics of atomic constructs. As the workflow patterns framework shows, another interesting aspect of study is the composition of atomic constructs to meaningful patterns of semantics. It would be interesting and beneficial to compose ontologically well-founded generic patterns of model semantics from representation theory and then to investigate how they related to other patterns, such as, for instance, workflow patterns.

Third, in the area of language and method engineering, we deem it fruitful to investigate whether process modeling languages that are built in light of several reference frameworks would outperform those that have been designed before the background of one framework only (examples for the latter include the work presented in [Gehlert *et al.* 2005] who based their language on the principles of representation theory, and [van der Aalst and ter Hofstede 2005], who based their language on the principles of workflow patterns).

LIMITATIONS

Our study suffers from several limitations. First, as noted in the introduction, our comparative assessment does not consider all aspects relevant to quality of models and modeling languages. The discussion of quality management proposals such as the semiotic quality framework [Lindland *et al.* 1994] or Hepp and Roman's [2007] semantic business process management framework in Section II of this paper highlights aspects that our analysis misses, including:

- The pragmatic quality of process modeling in a wider sense, e.g., [Krogstie *et al.* 2006b, Recker, 2007a], including not only the comprehension but also the effect the models produced have on modelers (e.g.,learning of the domain), and on the domain itself (i.e., process improvement) due to the way process modeling was conducted.
- The overall value [Krogstie *et al.* 2006a] or success [Bandara and Rosemann 2005] of process modeling, both project internal, but also organizational on a longer term.
- The user acceptance of process modeling languages, e.g. [Recker 2007b], and its impact on long-term viability of the process modeling initiative.
- The quality of the overall process modeling process, [Moody 2005].
- The aspects of BPM strategy, business logics, provision and consumption, as noted by Hepp and Roman [2007].

Second, a limitation is acknowledged related to the conduct of our analyses of BPMN by means of the BWW theory and the workflow patterns framework. In absence of automatic analysis tools, the process of language evaluation is by definition open to subjective interpretation. We did our best to mitigate subjectivity in our analysis, for instance by forming teams and having multiple rounds of coding, as reported in Section IV. While, for instance, the obtained Kappa values indicate reliability of our analyses and while we also documented all assumptions and rationales of our analysis (refer to [Recker *et al.* 2005; Recker *et al.* 2006] and [Wohed *et al.* 2006a; Wohed *et al.* 2006b]), we cannot guarantee beyond doubt the objectivity of our analyses, which is a typical limitation in this type of research [Rosemann *et al.* 2004].

Third, in the comparative assessment of the BWW theory and the workflow patterns framework using the case of the BPMN language, we noted in Section IV that a noted deficiency in light of the framework not necessarily implies a shortcoming of the language but may also reveal shortcomings of the scope of the quality frameworks. Accordingly, findings from a conceptual, analytical study such as the one presented in this paper, should always be approached with caution in absence of empirical validation. It would be a most insightful and stimulating challenge to operationalize some of the conjectures we reported in an empirical study to obtain more insights on the validity of our claims.

OUTLOOK

We do not consider our discussion to be complete. We look to further extend our assessment of evaluation frameworks to incorporate other levels of analysis such as the ones reported in our limitations section. Also, we seek to further populate our set model given in Figure 4 by comparatively assessing the findings from the evaluations of other process modeling languages such as BPEL (evaluated in [Green *et al.* 2007] and [Wohed *et al.* 2003b], respectively). This will allow us to provide some evidence for the generalizability of our results and the usefulness of our discussion in general.

In spite of some of the noted limitations of our study, most notably that we have not obtained an empirical perspective on either BPMN or the reference frameworks, we see first evidence of the usefulness of our approach. Our research is a first step towards more sophisticated process modeling languages that should be designed in light of not only one theoretical framework but rather in adherence to 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). Thereby we envisage the design of process modeling languages that not only provide complete and clear descriptions of real-world domains, but that can also be used to provide sophisticated support for workflow technologies and which may hence serve the two major purposes of process modeling at the same time.

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

ACKNOWLEDGEMENTS

We would like to express our enormous gratitude towards the fruitful collaboration with the workflow patterns team in our study. In particular, we would like to thank Dr. Petia Wohed for assisting us in our research and sharing with us her evaluation and insights into the workflow pattern analysis of BPMN. We would further like to thank our colleagues Dr. Marta Indulska and Dr. Peter Green for their assistance in the representation theory evaluation of BPMN.

Editor's Note: This article was received on March 28, 2007. It was with the authors five months for two revisions.

REFERENCES

Bandara, W. and M. Rosemann. (2005). "What Are the Secrets of Successful Process Modeling? Insights from an Australian Case Study," *Systèmes d'Information et Management* (10) 3, pp. 47-68.

Bodart, F., A. Patel, M. Sim, and R. Weber. (2001). "Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests," *Information Systems Research* (12) 4, pp. 384-405.

Bowen, P. L., R. A. O'Farrell, and F. Rohde. (2006). "Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End User Query Performance?" *Journal of the Association for Information Systems* (7) 8, pp. 514-544.

BPMI.org and OMG. (2006). "Business Process Modeling Notation Specification. Final Adopted Specification," Object Management Group, http://www.bpmn.org (February 20, 2006).

Bunge, M. A. (1977). *Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bunge, M. A. (2003). *Philosophical Dictionary*. New York, New York: Prometheus Books.

Cohen, J. (1960). "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement* (20) 1, pp. 37-46.

Davies, I., P. Green, S. Milton, and M. Rosemann. (2005). "Analysing and Comparing Ontologies with Meta Models," in J. Krogstie, T. Halpin, and K. Siau (Eds.) *Information Modeling Methods and Methodologies*, Hershey, Pennsylvania: Idea Group, pp. 1-16.

Davies, I., P. Green, M. Rosemann, M. Indulska et al. (2006). "How Do Practitioners Use Conceptual Modeling in Practice? " *Data & Knowledge Engineering* (58) 3, pp. 358-380.

Dehnert, J. and W. M. P. van der Aalst. (2004). "Bridging the Gap between Business Models and Workflow Specifications," *International Journal of Cooperative Information Systems* (13) 3, pp. 289-332.

Dumas, M., W. M. P. van der Aalst, and A. H. M. ter Hofstede (eds.). (2005). *Process Aware Information Systems: Bridging People and Software Through Process Technology*, Hoboken, New Jersey: John Wiley & Sons.

Frank, U. (1999). "Conceptual Modelling as the Core of the Information Systems Discipline - Perspectives and Epistemological Challenges," *5th America's Conference on Information Systems, Milwaukee, Wisconsin, 1999*, pp. 695-698.

Gehlert, A., U. Buckmann, and W. Esswein. (2005). "Ontology-Based Method Engineering," *11th Americas Conference on Information Systems, Omaha, Nebraska, 2005*, pp. 2824-2833.

Gemino, A. and Y. Wand. (2005). "Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties," *Data & Knowledge Engineering* (55) 3, pp. 301-326.

Green, P. and M. Rosemann. (2000). "Integrated Process Modeling. An Ontological Evaluation," *Information Systems* (25) 2, pp. 73-87.

Green, P. and M. Rosemann. (2001). "Ontological Analysis of Integrated Process Models: Testing Hypotheses," *Australasian Journal of Information Systems* (9) 1, pp. 30-38.

Green, P., M. Rosemann, and M. Indulska. (2005). "Ontological Evaluation of Enterprise Systems Interoperability Using ebXML," *IEEE Transactions on Knowledge and Data Engineering* (17) 5, pp. 713-725.

Green, P., M. Rosemann, M. Indulska, and C. Manning. (2007). "Candidate Interoperability Standards: An Ontological Overlap Analysis," *Data & Knowledge Engineering* (62) 2, pp. 274-291.

Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models*. Vol. 015. Enschede, The Netherlands: Telematica Instituut.

Gurr, C. A. (1999). "Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues," *Journal of Visual Languages and Computing* (10) 4, pp. 317-342.

Hepp, M. and D. Roman. (2007). "An Ontology Framework for Semantic Business Process Management," *8th International Conference Wirtschaftsinformatik, Karlsruhe, Germany, 2007*, pp. 423-440 2.

Kiepuszewski, B., A. H. M. ter Hofstede, and W. M. P. van der Aalst. (2003). "Fundamentals of Control Flow in Workflows," *Acta Informatica* (39) 3, pp. 143-209.

Krogstie, J., V. Dalberg, and S. M. Jensen. (2006a). "Increasing the Value of Process Modelling," in Y. Manolopoulos, J. Filipe, P. Constantopoulos, and J. Cordeiro (Eds.) *Proceedings of the 8th International Conference on Enterprise Information Systems: Databases and Information Systems Integration*, Paphos, Cyprus, pp. 70-77.

Krogstie, J. and H. D. Jørgensen. (2003). "Quality of Interactive Models," in, vol. 2784 M. Genero, F. Grandi, W.-J. van den Heuvel, J. Krogstie et al. (Eds.) *Advanced Conceptual Modeling Techniques - ER 2002 Workshops*, Tampere, Finland: Springer, pp. 351-363.

Krogstie, J., G. Sindre, and H. D. Jørgensen. (2006b). "Process Models Representing Knowledge for Action: A Revised Quality Framework," *European Journal of Information Systems* (15) 1, pp. 91-102.

Lindland, O. I., G. Sindre, and A. Sølvberg. (1994). "Understanding Quality in Conceptual Modeling," *IEEE Software* (11) 2, pp. 42-49.

Lyytinen, K. (2006). ""Ontological Foundations of Conceptual Modeling" by Boris Wyssysek - A Critical Response," *Scandinavian Journal of Information Systems* (18) 1, pp. 81-84.

Milton, S. and E. Kazmierczak. (2004). "An Ontology of Data Modelling Languages: A Study Using a Common-Sense Realistic Ontology," *Journal of Database Management* (15) 2, pp. 19-38.

Moody, D. L. (2005). "Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions," *Data & Knowledge Engineering* (15) 3, pp. 243-276.

Moore, G. C. and I. Benbasat. (1991). "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2) 3, pp. 192-222.

Nysetvold, A. G. and J. Krogstie. (2006). “Assessing Business Process Modeling Languages Using a Generic Quality Framework,” in K. Siau (Ed.) *Advanced Topics in Database Research Vol. 5*, Hershey, Pennsylvania: Idea Group, pp. 79-93.

Opdahl, A. L. and B. Henderson-Sellers. (2002). "Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model," *Software and Systems Modeling* (1) 1, pp. 43-67.

Recker, J. (2007a). "A Socio-Pragmatic Constructionist Framework for Understanding Quality in Process Modelling," *Australasian Journal of Information Systems* (14) 2, pp. 43-63.

Recker, J. (2007b). “Why Do We Keep Using A Process Modelling Technique?” *18th Australasian Conference on Information Systems, Toowoomba, Australia, 2007b*.

Recker, J. and M. Indulska. (2007). "An Ontology-Based Evaluation of Process Modeling with Petri Nets," *Journal of Interoperability in Business Information Systems* (2) 1, pp. 45-64.

Recker, J., M. Indulska, M. Rosemann, and P. Green. (2005). “Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN.” *16th Australasian Conference on Information Systems, Sydney, Australia, 2005*.

Recker, J., M. Indulska, M. Rosemann, and P. Green. (2006). “How Good is BPMN Really? Insights from Theory and Practice.” *14th European Conference on Information Systems, Goeteborg, Sweden, 2006*, pp. 1582-1593.

Rosemann, M. and P. Green. (2002). "Developing a Meta Model for the Bunge-Wand-Weber Ontological Constructs," *Information Systems* (27) 2, pp. 75-91.

Rosemann, M., P. Green, and M. Indulska. (2004). “A Reference Methodology for Conducting Ontological Analyses,” in, vol. 3288 H. Lu, W. Chu, P. Atzeni, S. Zhou et al. (Eds.) *Conceptual Modeling – ER 2004*, Shanghai, China: Springer, pp. 110-121.

Rosemann, M., P. Green, M. Indulska, and J. Recker. (forthcoming). "Using Ontology for the Representational Analysis of Process Modeling Techniques," *International Journal of Business Process Integration and Management*,in press.

Rosemann, M., J. Recker, M. Indulska, and P. Green. (2006). “A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars,” in, vol. 4001 E. Dubois and K. Pohl (Eds.) *Advanced Information Systems Engineering - CAiSE 2006*, Luxembourg, Grand-Duchy of Luxembourg: Springer, pp. 447-461.

Russell, N., A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst (2005a) “Workflow Data Patterns: Identification, Representation and Tool Support,” in, vol. 3716 L. M. L. Delcambre, C. Kop, H. C. Mayr, J. Mylopoulos et al. (Eds.) *Conceptual Modeling - ER 2005*, Klagenfurt, Austria: Springer, pp. 353-368.

Russell, N., A. H. M. ter Hofstede, W. M. P. van der Aalst, and N. A. Mulyar. (2006a). "Workflow Control-Flow Patterns: A Revised View", *BPM Center Report BPM-06-22*, BPMcenter.org,

Russell, N., W. M. P. van der Aalst, and A. H. M. ter Hofstede. (2006b). “Workflow Exception Patterns,” in, vol. 4001 E. Dubois and K. Pohl (Eds.) *Advanced Information Systems Engineering - CAiSE 2006*, Luxembourg, Grand-Duchy of Luxembourg: Springer, pp. 288-302.

Russell, N., W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond. (2005b). “Workflow Resource Patterns: Identification, Representation and Tool Support,” in, vol. 3520 Ó. Pastor and J. Falcão e Cunha (Eds.) *Advanced Information Systems Engineering - CAiSE 2005*, Porto, Portugal: Springer, pp. 216-232.

Russell, N., W. M. P. van der Aalst, A. H. M. ter Hofstede, and P. Wohed. (2006c). “On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling,” in, vol. 53 M. Stumptner, S. Hartmann, and Y. Kiyoki (Eds.) *Conceptual Modelling 2006: 3rd Asia-Pacific Conference on Conceptual Modelling*, Hobart, Australia: Australian Computer Society, pp. 95-104.

Shanks, G., E. Tansley, and R. Weber. (2003). "Using Ontology to Validate Conceptual Models," *Communications of the ACM* (46) 10, pp. 85-89.

Siau, K. (2004). "Informational and Computational Equivalence in Comparing Information Modeling Methods," *Journal of Database Management* (15) 1, pp. 73-86.

ter Hofstede, A. H. M. and T. P. van der Weide. (1992). "Formalisation of Techniques: Chopping down the Methodology Jungle," *Information and Software Technology* (34) 1, pp. 57-65.

van der Aalst, W. M. P. (1999). "Formalization and Verification of Event-driven Process Chains," *Information and Software Technology* (41) 10, pp. 639-650.

van der Aalst, W. M. P. (2003). "Don’t Go with the Flow: Web Services Composition Standards Exposed," *IEEE Intelligent Systems* (18) 1, pp. 72-76.

van der Aalst, W. M. P. and A. H. M. ter Hofstede. (2005). "YAWL: Yet Another Workflow Language," *Information Systems* (30) 4, pp. 245-275.

van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. (2003). "Workflow Patterns," *Distributed and Parallel Databases* (14) 1, pp. 5-51.

Wahl, T. and G. Sindre. (2006). “An Analytical Evaluation of BPMN Using a Semiotic Quality Framework,” in K. Siau (Ed.) *Advanced Topics in Database Research Vol. 5*, Hershey, Pennsylvania: Idea Group, pp. 102-113.

Wand, Y. and R. Weber. (1990). "An Ontological Model of an Information System," *IEEE Transactions on Software Engineering* (16) 11, pp. 1282-1292.

Wand, Y. and R. Weber. (1993). "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars," *Journal of Information Systems* (3) 4, pp. 217-237.

Wand, Y. and R. Weber. (1995). "On the Deep Structure of Information Systems," *Information Systems Journal* (5) 3, pp. 203-223.

Weber, R. (1997). *Ontological Foundations of Information Systems*. Melbourne, Australia: Coopers & Lybrand and the Accounting Association of Australia and New Zealand.

Weber, R. and Y. Zhang. (1996). "An Analytical Evaluation of NIAM's Grammar for Conceptual Schema Diagrams," *Information Systems Journal* (6) 2, pp. 147-170.

Wohed, P., E. Perjons, M. Dumas, and A. H. M. ter Hofstede. (2003a). “Pattern Based Analysis of EAI Languages - The Case of the Business Modeling Language,” in *Proceedings of the 5th International Conference on Enterprise Information Systems. Vol. 3*, Angers, France: Escola Superior de Tecnologia do Instituto Politecnico de Setubal, pp. 174-184.

Wohed, P., W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. (2003b). “Analysis of Web Services Composition Languages: The Case of BPEL4WS,” in, vol. 2813 I.-Y. Song, S. W. Liddle, T. W. Ling, and P. Scheuermann (Eds.) *Conceptual Modeling - ER 2003*, Chicago, Illinois: Springer, pp. 200-215.

Wohed, P., W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. (2006a). "Pattern-based Analysis of BPMN - An extensive evaluation of the Control-flow, the Data and the Resource Perspectives (revised version)", *BPM Center Report BPM-06-17*, BPMcenter.org,

Wohed, P., W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede et al. (2006b). "On the Suitability of BPMN for Business Process Modelling." *Business Process Management - BPM 2006, Vienna, Austria, 2006b*, pp. 161-176. Lecture Notes in Computer Science 4102.

Wyssusek, B. (2006). "On Ontological Foundations of Conceptual Modelling," *Scandinavian Journal of Information Systems* (18) 1, pp. 63-80.

ABOUT THE AUTHORS

Jan Recker is a Ph.D candidate at the Business Process Management Cluster at the Faculty of Information Technology, Queensland University of Technology Brisbane, Australia. He received his BScIS and MScIS from the University of Muenster, Germany in 2004. His research interests include Business Process Modeling, Representation Theory, Standards Adoption and Process Flexibility. Jan has published more than fourty refereed scholarly papers on these topics, including articles in journals such as *Australasian Journal of Information Systems*, *Journal of Interoperability in Business Information Systems*, *International Journal of Business Process Integration and Management* and *Applied Ontology*.

Michael Rosemann is a professor for Information Systems and Co-Leader of the Business Process Management Cluster at Queensland University of Technology, Brisbane. He received his MBA and Ph.D in Information Systems from the University of Muenster, Germany. His main areas of interest are business process management, process modeling, enterprise systems and ontologies. He published more than 120 refereed papers including publications in journals such as *MIS Quarterly, Information Systems, IEEE Transactions on Knowledge and Data Engineering, European Journal of Information Systems, Decision Support Systems and Information Systems Frontiers*. Michael is the author and editor of six books, on the editorial board of six journals and a member of the ARC (Australian Research Council) College of Experts.

John Krogstie has a Ph.D (1995) and a MSc (1991) in Information Systems, both from the Norwegian University of Science and Technology (NTNU). He is a professor in Information Systems at IDI, NTNU, Trondheim, Norway. He is also a senior advisor at SINTEF. He was employed as a manager in Accenture 1991-2000. John Krogstie is the Norwegian representative for IFIP TC8 and vice-chair of IFIP WG 8.1 on information systems design and evaluation, where he is the initiator and leader of the task group for Mobile Information Systems. He was recently general chair of CAiSE'07, and has published around 80 refereed papers in journals, books and archival proceedings since 1991.

Communications of the Association for Information Systems

ISSN: 1529-3181