A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars

Michael Rosemann[1], Jan Recker[1], Marta Indulska[2], and Peter Green[2]

[1] Faculty of Information Technology
Queensland University of Technology
Brisbane, Australia
{m.rosemann, j.recker}@qut.edu.au
[2] UQ Business School
University of Queensland
Ipswich, Australia
{m.indulska, p.green}@business.uq.edu.au

Abstract. A plethora of process modeling techniques has been proposed over the years. One way of evaluating and comparing the scope and completeness of techniques is by way of representational analysis. The purpose of this paper is to examine how process modeling techniques have developed over the last four decades. The basis of the comparison is the Bunge-Wand-Weber representation model, a benchmark used for the analysis of grammars that purport to model the real world and the interactions within it. This paper presents a comparison of representational analyses of several popular process modeling techniques and has two main outcomes. First, it provides insights, within the boundaries of a representational analysis, into the extent to which process modeling techniques have developed over time. Second, the findings also indicate areas in which the underlying theory seems to be over-engineered or lacking in specialization.

1 Introduction

While the general objectives and methods of Business Process Management (BPM) are not new, BPM has only recently received a significant amount of attention and is now perceived to be a main business priority [1]. However, the actual modeling of business processes still presents major challenges for organizations. As graphical presentations of current or future business processes, business process models serve two main purposes. First, intuitive business process models are used for scoping the project, and capturing and discussing business requirements and process improvement initiatives with subject matter experts. A prominent example of such a business modeling technique is the Event-driven Process Chain (EPC). Second, business process models are used for process automation, which requires their conversion into executable languages. These automated techniques have higher requirements in terms of expressive power. Examples include Petri nets or the Business Process Modeling Notation (BPMN), a new Business Process Execution Language for Web Services (BPEL4WS)-conform notation.

Overall, a high number of process modeling techniques have been proposed since Carl Petri published his initial ideas on Petri nets in 1962 [2], and process modeling has become one of the most popular reasons for conceptual modeling [3]. Clearly, a theoretical basis is required to assist in the evaluation and comparison of available process modeling techniques. Given the existence of such theory, it would not only be possible to evaluate these techniques, but also to determine if the discipline of process modeling is building on previous knowledge, and if new techniques denote an actual improvement. A promising candidate of such theories, the Bunge-Wand-Weber (BWW) representation model, uses the principles of *representational analysis* for an investigation of a modeling technique's strength and weaknesses. The BWW representation model denotes a widespread means for evaluating conceptual modeling grammars for information systems analysis and design. We will employ this model as a benchmark and filter through which we will assess comparatively the most popular process modeling techniques. Thus, our research is motivated in several ways:

1. to provide theoretical guidance in the evaluation and comparison of available process modeling techniques;
2. to propose a measure of development of process modeling over time;
3. to highlight representations that process modeling languages do not appear to address; and
4. to add to the development of the BWW theoretical models.

The *aim of this paper* then is to study the development of process modeling techniques over time. As a measurement for the development of these techniques we selected *ontological completeness,* defined as the coverage of constructs as proposed by the Bunge-Wand-Weber representation model. We are very much aware that ontological completeness is not the only relevant criterion for the evaluation of the capabilities of a modeling technique. Thus, the focus on the set of BWW constructs leads to a specific scope in the evaluation. The study of modeling technique development is based on a review of previous published BWW analyses of process modeling techniques. In order to report on a reasonably complete set of modeling techniques, we also conducted our own analysis of two additional prominent modeling techniques, *viz.,* Petri nets and BPMN. Overall, this paper considers twelve common process modeling techniques and extracts the similarities and differences in terms of the ontological completeness of these techniques. The consolidated findings point to common shortcomings of modeling techniques, but also they highlight the main differentiating features. As part of this work, the BWW representation model is also evaluated in terms of appropriateness of its specification within the business process modeling domain.

This paper is structured as follows. The next section provides an overview of the Bunge-Wand-Weber set of models and its previous applications in the area of process modeling, including our analyses of Petri nets and BPMN. Section 3 presents and discusses the findings of the comparison of process modeling techniques from the viewpoint of ontological completeness. Also, it reports on potential issues of the BWW set of models with respect to their application to the area of process modeling. The paper concludes in section 4 with a discussion of results, limitations, and future research.

2 Related Work & Background

2.1 Representation Theory in Information Systems

Over the last few decades many conceptual modeling techniques, used to define requirements for building information systems, have emerged with limited theoretical foundation underlying their conception or development [4]. Concerned that this situation would result in the development of information systems that were unable to capture completely important aspects of the real world, Wand and Weber [5-7] developed and refined a set of models for the evaluation of the representational capability of the modeling techniques and the scripts prepared using such techniques. These models are based on an ontology defined by Bunge [8] and are referred to as the Bunge-Wand-Weber (BWW) models. Generally, *ontology* studies the nature of the world and attempts to organize and describe what exists in reality, in terms of the properties of, the structure of, and the interactions between real-world things [9]. As computerized information systems are representations of real world systems, Wand and Weber suggest that a theory of representation based on ontology can be used to help define and build information systems that contain the necessary representations of real world constructs including their properties and interactions. The *BWW representation model* is one of three theoretical models defined by Wand and Weber [7] that make up the Representation Theory. Its application to information systems foundations has been referred to by a number of researchers [10]. Some minor alterations have been carried out over the years by Wand and Weber [6, 7] and Weber [11], but the current key constructs of the BWW model can be grouped into the following clusters: things including properties and types of things; states assumed by things; events and transformations occurring on things; and systems structured around things (see Appendix 1 for a complete list).

Weber [11] suggests that the BWW representation model can be used to analyze a particular modeling technique to make predictions on the modeling strengths and weaknesses of the technique, in particular its capabilities to provide *complete* and *clear* representations of the domain being modeled. He clarifies two main evaluation criteria that may be studied according to the BWW model: *Ontological Completeness* and *Ontological Clarity*. The focus of our study is ontological completeness only, *i.e.,* the analysis of the extent to which a process modeling technique covers completely the set of constructs proposed in the BWW representation model.

Among other theories that have been proposed as a basis for representational analysis of conceptual modeling in information systems, the approaches of Chisholm [12] and Guizzardi *et al.* [13] are to be regarded as closest to the ideas of Wand and Weber. These upper-level ontologies have been built for similar purposes and seem to be equally expressive [14] but have not yet achieved the popularity and dissemination of the BWW models.

2.2 Previous Representational Analyses of Process Modeling Techniques

Only limited research efforts have been made to compare process modeling techniques based on an established theoretical model, refer, for instance, to [15]. However, these proposals neither appear to have been widely adopted in practice nor do they have an established track record. On the contrary, the BWW representation model has been used in over twenty-five research projects for the evaluation of different modeling techniques (see [10] for an overview), including data models, object-oriented models and reference models. It also has a track record in the area of process modeling, with contributions coming from various researchers. In this section, we briefly summarize these studies that focused on process modeling techniques.

Keen and Lakos [16] determined essential features for a process modeling scheme by using the BWW representation model to evaluate the degree of ontological completeness of six process modeling techniques in a historical sequence. Empirical studies to validate the results have not been conducted. The process modeling techniques examined include the ANSI flowchart notation, the ISO Conceptual Schema Model (ISO/TC97) [17], the Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise (MERISE) [18], the Data Flow Diagram (DFD) notation [19], the Integrated Definition Method 3 Process Description Capture Method (IDEF3) [20], and the Language for Object-Oriented Petri nets (LOOPN++) [21]. From their analysis, Keen and Lakos concluded that, in general, the BWW representation model facilitates the interpretation and comparison of process modeling techniques. They propose the BWW constructs of *system*, *system composition*, *system structure*, *system environment*, *transformation*, and *coupling* to be essential process modeling technique requirements. As our analysis will show, however, these findings are not entirely reflected in the leading process modeling techniques we consider.

Green and Rosemann [22] used the BWW representation model to analyze the Event-Driven Process Chain (EPC) notation [23], focusing on both ontological completeness and clarity. Their findings have been empirically validated through interviews and surveys [24]. Confirmed shortcomings were found in the EPC notation with regard to the representation of real world objects, in the definition of business rules, and in the thorough demarcation of the analyzed system.

Green *et al.* [25] also examined the Electronic Business using eXtensible Markup Language Business Process Specification Schema (ebXML BPSS) v1.01 [26] in terms of ontological completeness and clarity. While the empirical validation of results has not yet been performed, the analysis shows a relatively high degree of ontological completeness.

Green *et al.* [27] examined the ontological completeness of four leading standards for enterprise system interoperability, including BPEL4WS v1.1 [28], Business Process Modeling Language v1.0 (BPML) [29], Web Service Choreography Interface v1.0 (WSCI) [30], and ebXML v1.1 [26]. In addition, a minimal ontological overlap (MOO) analysis [7, 11] has been conducted in order to determine the set of modeling standards with a minimum number of overlapping constructs but with maximal ontological completeness (MOC), *i.e.*, maximum expressiveness, between the selected standards. The study identified two sets of standards that, when used together, allow for the most expressive power with the least overlap of constructs, *viz.*, ebXML and

BPEL4WS, and, ebXML and WSCI. The results of the analysis remain to be tested empirically.

While there has been further work that concentrates on the representational analysis of dynamic modeling techniques (see, for example, [31, 32]), these particular techniques are not considered in our research. For example, modeling techniques relying on an object-oriented paradigm (like UML, OML, OPM, or LOOPN++) have not been included in this study. These techniques, applied in software engineering rather than process management contexts, have different or extended requirements in terms of representation capabilities and are, therefore, limited in comparability to ‘pure’ process modeling notations. We believe that the inclusion of such techniques would limit the comparability of the results to process modeling languages that focus on control flow.

2.3 Representational Analysis of Petri nets and BPMN

While the previous representational analyses of process modeling techniques covered the main techniques, we felt that the field should be further extended by at least two more prominent techniques, *viz*., Petri nets and BPMN.

We conducted our own representational analysis of *Petri nets* in its original and most basic form [2], as we perceive it to be the intellectual birthplace of more rigorous and disciplined process modeling. Petri nets are composed of places, transitions, tokens, and arcs together with an initial state called the initial marking. As places and arcs may be assigned a certain weight of tokens, the notation allows for quite extensive modeling purposes. Special attention is, for example, paid to its capability of business process simulation. Additionally, due to the underlying strict formal foundation, Petri nets provide the capabilities for mathematical analyses and means to be directly executed [33]. Due to this rigorous, yet flexible, specification we found that although the notation originally merely consists of seven constructs, its ontological completeness is quite high. While this apparent flexibility in the interpretation of the Petri net constructs resulted in more than the seven expected mappings, Petri nets still lack ontological completeness. For example, there is no support for the modeling of systems structured around things. Hence, it is problematic to define thoroughly and demarcate the modeled system, a deficit that in turn causes understandability problems in terms of the scope as well as subparts and interrelationships of system elements. Even though our study is based on the notion of ontological completeness, it is important to point out that the same flexibility that affords Petri nets a higher ontological completeness, also results in extensive construct overload [11]. For example, a place construct in a Petri net can be used to represent a *thing*, *class,* or *state*. Such flexibility, while seemingly an advantage, can result in models that are harder to interpret. This weakness can result in ambiguity of the models as extra-model knowledge is required to understand what is meant when a particular construct is used in a model, *e.g.,* whether a place in a given model represents a thing, a class of things, or a state of a thing.

The *Business Process Modeling Notation (BPMN)* [34] is a recently proposed standard that 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 BPMN specification defines thirty-eight distinct language constructs plus attributes, grouped in four basic categories of elements, *viz.*, *Flow Objects*, *Connecting Objects*, *Swimlanes* and *Artefacts* [34]. For example, nine distinct event types and three different event dimensions are included.

As the focus of this paper is on the comparison of different process modeling techniques, we only provide a reduced summary of the outcomes of this analysis. A more complete analysis is discussed in detail in [35]. Our analysis shows that the specification provides a relatively high degree of ontological completeness. However, BPMN is not ontologically complete. For example, states assumed by things cannot be modeled with the BPMN notation. This situation can result in a lack of focus in terms of state and transformation law foundations for capturing business rules. Also, systems structured around things are under-represented. For example, as there is no representation of system structure, problems will arise when information needs to be obtained about the dependencies within a modeled system.

3 Comparison of Representational Analyses

3.1 Research Design and Overview

We reviewed and compared analyses of twelve process modeling techniques with the focus being on the ontological completeness of these techniques. As we are aware that many available process modeling techniques have been designed for distinct purposes, we placed special emphasis on ensuring comparability of the analyses. In order to ensure a reasonably holistic overview of this area, our analysis covered a wide selection of modeling techniques for different purposes, ranging from mere illustration methods (*e.g.*, Flowcharts) to integrated techniques (*e.g.*, EPC), and also covering more recent techniques capable of both process description and execution (*e.g.*, ebXML and BPEL4WS).

As the prior analyses were independently conducted by different research groups and since the representational analyses referred to varied research purposes, effort was put into making the individual analyses comparable. We did neither question nor review the mapping results as proposed by the different research groups. So, our study consolidates existing analyses instead of revising or extending previous research work. The results of our comparison are shown in Table 1, where each tick indicates that the specified BWW construct can be represented by the analyzed technique.

However, due to varying sets of BWW representation model constructs included in the analyses, we had to generalize some constructs of the BWW model in order to stabilize the comparison of the evaluations:

- As some analyses did not entirely differentiate between property types, these types were generalized here to the super-type ‘property’. Therefore, if a mapping was found for a sub-type of ‘property’, then the mapping was recorded as belonging to the super-type ‘property’.

- Similarly, as some analyses did not consider the constructs of *stability condition* and *corrective action* in the context of the *lawful transformation* construct, we generalized mappings of these to a mapping of the *lawful transformation* construct.
- As the construct *process* [22] was not specified in the BWW representation model as defined in [6, 7, 11] we did not consider it in our comparison.

Table 1. Comparison of representational analyses of process modeling techniques

BWW Construct / *Language* / *Version* / *Year*	Petri net	ANSI Flow-charts	DFD	ISO TC87	Merise	EPC	IDEF3	ebXML 1.01	BPML 1.0	WSCI 1.0	BPEL4WS 1.1	BPMN 1.0	*Construct Coverage*
	1962	1970	1979	1982	1992	1992	1995	2001	2002	2002	2003	2004	
Things including Properties and Types of Things													
THING	✓			✓	✓		✓					✓	5 / 12
CLASS	✓							✓	✓	✓	✓	✓	6 / 12
KIND												✓	1 / 12
PROPERTY			✓			✓	✓	✓	✓	✓	✓	✓	8 / 12
States assumed by Things													
STATE	✓					✓	✓	✓	✓	✓	✓		7 / 12
CONCEIVABLE STATE SPACE								✓					1 / 12
STATE LAW	✓			✓	✓	✓		✓					5 / 12
LAWFUL STATE SPACE	✓							✓					2 / 12
STABLE STATE						✓		✓					2 / 12
UNSTABLE STATE	✓							✓					2 / 12
HISTORY								✓					1 / 12
Events and Transformations occurring on Things													
EVENT	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	10 / 12
CONCEIVABLE EVENT SPACE								✓					1 / 12
LAWFUL EVENT SPACE								✓					1 / 12
EXTERNAL EVENT				✓	✓	✓		✓	✓	✓	✓	✓	8 / 12
INTERNAL EVENT	✓			✓	✓	✓		✓	✓	✓	✓	✓	9 / 12
WELL-DEFINED EVENT	✓					✓		✓	✓	✓	✓	✓	7 / 12
POORLY DEFINED EVENT								✓	✓	✓	✓	✓	5 / 12
TRANSFORMATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	12 / 12
LAWFUL TRANSFORMATION	✓			✓	✓	✓		✓	✓	✓	✓	✓	9 / 12
ACTS ON	✓							✓		✓	✓	✓	5 / 12
COUPLING		✓	✓		✓		✓	✓		✓	✓	✓	8 / 12
Systems structured around Things													
SYSTEM			✓		✓		✓	✓		✓	✓	✓	7 / 12
SYSTEM COMPOSITION			✓		✓		✓			✓	✓	✓	6 / 12
SYSTEM ENVIRONMENT			✓									✓	2 / 12
SYSTEM STRUCTURE					✓		✓			✓	✓		4 / 12
SUBSYSTEM								✓				✓	2 / 12
SYSTEM DECOMPOSITION			✓				✓					✓	3 / 12
LEVEL STRUCTURE			✓			✓	✓					✓	4 / 12
Representational Coverage	12 / 29	2 / 29	8 / 29	7 / 29	11 / 29	11 / 29	11 / 29	22 / 29	10 / 29	15 / 29	15 / 29	19 / 29	

3.2 Issues in Process Modeling Techniques: Findings and Propositions

The notion of ontological completeness of a particular process modeling technique serves as an indication of its *representational capabilities*, being the extent to which the techniques are able to provide complete descriptions of a real-world domain.

The consolidation of previous representational analyses with our analyses of Petri nets and BPMN leads to several interesting results. A longitudinal study of the ontological completeness shows an obvious increase in the coverage of BWW constructs that can be interpreted as a sign of increasing representational development over time. Fig. 1 visualizes this trend over time, as measured by the number of BWW constructs covered by each technique. We can see that, while the original Petri net specification did not provide exceptionally good representational coverage[1] (41%) as defined by the BWW representation model, it still performed better than more recent grammars such as DFD (28%) or IDEF3 (38%). A noticeable spike in Fig. 1 depicts the high level of development (in terms of ontological completeness) of the ebXML standard (76%). It is interesting to note that ebXML is specified in UML, with a semi-formal construct definition and description, whereas BPEL4WS, WSCI, and BPMN, for example, have textual specifications supplemented by diagrams of examples. As such, the ebXML specification is less subjective in its possible interpretations.

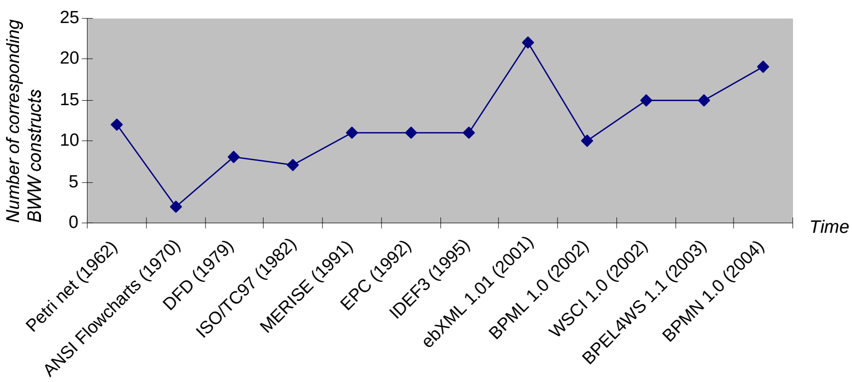

Fig. 1. Comparison of representation mapping analyses

BPMN also appears to perform very well (66%) and hence appears to be quite mature in terms of representation capabilities. This higher level of ontological completeness can perhaps partly be explained by the fact that previous approaches, including EPC and Petri nets, influenced the development of BPMN [34]. Also, this finding is not only supported by the number of identified mappings to BWW constructs, but also by the specialization of the constructs. For example, BPMN has

[1] The degree of representational coverage (DrC) is here calculated as the number of BWW constructs found to be represented by language constructs #L divided by the number of constructs defined in the BWW representation model #M = 29. Note here that each BWW construct has the same weight.

sub-types of *event* and *transformation* that allow a more rigorous and expressive model to be defined. However, this strength can potentially also be its weakness as the varied sub-types of *transformation* and *event* will require thorough understanding by the user in order to appropriately represent the right types of transformations and events, respectively.

It appears that techniques that focus on describing process flow from a business perspective (for instance DFD and IDEF3) are less ontologically complete than those that have to cater for more syntactical rigor due to their focus on executability or translatability into executable languages (such as BPEL4WS or ebXML).

In terms of the coverage of BWW constructs, Table 2 shows some occurrences of mappings of BWW representation model constructs within the considered analyses of process modeling techniques.

Table 2. Analysis of construct occurrences

Most supported BWW constructs		Least supported BWW constructs	
Construct	*Occurrence ratio (%)*	*Construct*	*Occurrence ratio (%)*
TRANSFORMATION	100	KIND	8
EVENT	83	HISTORY	8
LAWFUL TRANSFORMATION	75	CONCEIVABLE STATE SPACE	8
INTERNAL EVENT	75	CONCEIVABLE EVENT SPACE	8
EXTERNAL EVENT	67	LAWFUL EVENT SPACE	8
COUPLING	67	LAWFUL STATE SPACE	17
PROPERTY	67	STABLE STATE	17
STATE	58	UNSTABLE STATE	17
SYSTEM	58	SYSTEM ENVIRONMENT	17
WELL-DEFINED EVENT	58	SUBSYSTEM	17

As can be expected in a BPM domain, each of the analyzed techniques has the ability to represent the BWW construct *transformation* – one of the core concepts in process modeling [36]. Seventy-five percent of these techniques also allow differentiation between all possible transformations and a lawful transformation that is allowed under the business rules in a given case. It is also interesting to note that while *transformation* has full support, neither *event* nor *state* have the same occurrence, with *state* being represented in under sixty percent of the modeling techniques. This situation is surprising, given the importance of events and states in process modeling [36].

There is divided support for the cluster *things including properties and types of things.* Closer inspection of Table 1 shows that while earlier process modeling techniques provided a construct for a specific *thing* (overall support: 42%), more recent standards have representation capabilities for *class* (overall support: 50%) rather than *thing.* Therefore, it would appear that, in general, there has been a move to model classes of things rather than actual things, *i.e.*, instances. It is also interesting to note that only BPMN is able to cover all aspects of things, including properties and types of things (see Table 1). In this respect, BPMN appears to denote a considerable improvement compared to other techniques.

Throughout the BPM domain, a lack of support for business rule definitions can be observed (see also [22, 35]). Because *conceivable* and *lawful event spaces* as well as *state spaces* are under-represented – none of these constructs has support of more than seventeen percent – state and transformation modeling is unclear for the modeler who may encounter confusion when determining which set of events and states can occur in the system and which events and states are possible but should not be allowed. A closer look at Table 1 reveals that most techniques achieve a very low degree of representational coverage in the cluster of *states assumed by things*, except for ebXML (100% in this cluster) and – interestingly – Petri nets (52% in this cluster). This situation suggests that the modeling of business rules is heavily dependant on rigorous state and state law specification. The mathematical specification of Petri nets seems to be advantageous in this aspect.

Also, there appears to be inconsistent support for *systems structured around things*. From the list of seven BWW constructs grouped in this cluster, five are represented in under thirty-four per cent of the modeling techniques. Thus, appropriate structuring and differentiation of modeled things, such as business partners, is not well supported, a fact we find quite problematic, especially in light of collaborative business processes and interoperability. Table 1 suggests that DFD, IDEF3 and BPMN models perform best in representing systems structured around things. All these grammars have in common dedicated constructs for decomposing process models into interlinked subsets.

3.3 Focusing the Underlying Theory

A representational analysis of modeling techniques has two facets. On the one side, it provides a filtering lens that facilitates insights into potential issues with a modeling technique. On the other side, it can also contribute to the further development of the selected theoretical basis. In fact, our findings from the longitudinal analysis of process modeling techniques align with some of the previous criticisms of BWW representation model-based analyses [37, 38].

The fact that even the most developed process modeling technique (ebXML) supports only 76% of the BWW constructs suggests that the selected theory of representation might be too demanding. With regard to this potential *lack of relevance* of the BWW representation model, we suggest the development and use of a specialized BWW model for the domain of business process modeling. The current BWW representation model needs to be investigated in order to determine areas that need further specialization, extension, deletion, or renaming. For example, *events* and *transformations* occurring on *things* may require further specialization. BPMN distinguishes between nine event types, representing a differentiation scheme that is not covered by the BWW constructs of *event* and its subtypes. The same situation can be seen in standards such as ebXML, BPEL4WS, BPML, and WSCI. A similar situation holds for the *transformation* construct that we often found to be susceptible to construct redundancy. For example, in BPML there are ten language constructs representing different types of transformations. A similar situation exists in standards like BPEL4WS and ebXML. This situation implies that, just as 'properties' in the

BWW representation model are specialized, perhaps transformations should also be specialized for the domain of BPM.

It is interesting also to note that throughout the analyses of process modeling techniques, control flow mechanisms such as logical connectors, selectors, gateways and the like are regarded as construct excess as they do not map to any construct of the BWW model. However, these constructs are deemed to be essential to the BPM domain (for empirical evidence supporting this proposition refer, for example, to [35]). Consequently, we are considering how the BWW model might be extended to better reflect such control flow concepts important to the BPM domain.

Taking a methodological viewpoint to the BWW representation model-based analysis, we found the *lack of objectivity* issue persisting. Significant effort had to be applied in objectifying the different analyses in terms of finding a comparable set of BWW constructs. This situation highlights the need for the use of meta-models in conducting analyses. A meta-model allows for a clearer description of the source representation model constructs as well as less subjective evaluation of the target grammar, partially through pattern matching, assuming the meta-model and the grammar are specified in the same notation. A BWW meta-model has been developed [39] for such a purpose and its use for evaluating grammars with meta model-based specifications has been promoted and discussed in [14, 37].

4 Conclusions, Limitations & Future Research

This paper presents the first comprehensive longitudinal study comparing previous representational analyses of process modeling techniques. The innovative comparative study also includes our outcomes of the initial representational analyses of Petri nets and the new proposed modeling standard, BPMN. The findings clearly show signs of a developing modeling discipline, measured by an increased ontological completeness of process modeling techniques over time. The results also identify the common core constructs of process modeling techniques (for example, *transformation, properties, events*) as well as their key differentiators (for example, *subsystem, system environment, lawful state space*). Furthermore, the findings provide valuable insights for the future application of the BWW representation model as a benchmark for such analyses of modeling techniques. As shown in Table 2, there are some constructs of the BWW representation model that are supported by only one technique of the chosen twelve, for example the constructs *kind* and *history*. While this might indicate an area for improvement in the representation power of process modeling techniques, it might also indicate that, perhaps, the particular BWW construct is not necessary for modeling in the domain of BPM. Such issues require further empirical testing (currently under way) in order to determine whether the theory of representation requires pruning and specialization or whether the techniques require refinement and extensions in order to be able to model what is represented by the BWW construct. Such research might also motivate other researchers to conduct a similar study for data or object-oriented modeling techniques.

Furthermore, the outcomes will be of interest to the developers and users of process modeling techniques. Developers of process modeling techniques should be motivated to examine representational analyses of currently used process modeling tools in order to build upon these grammars and counteract any weaknesses in the newly developed techniques or technique extensions. On the other hand, users of process modeling techniques might be motivated to use ontological completeness as one potential evaluation criterion for the selection of a more appropriate modeling technique.

We identify some limitations in our research. Most notably, we based our study on previous representational analyses that have been conducted by different researchers. Therefore, there may exist issues related to the comparability of the analyses due to the impact of the subjective interpretations of the researcher [37]. Second, we constrained the considered representational analyses to analyses based on the BWW representation model that in turn limits the generalization of the results and also the number of techniques we were able to consider. We believe, however, that the selected set of techniques is representative of the most popular techniques in the BPM field. It also enables us to focus our work and to avoid the necessity to translate between different theoretical bases. Third, we focused on ontological completeness, thereby giving a one-dimensional view of modeling technique development over time.

In our future research, we will extend this analysis to include also ontological clarity as an evaluation criterion. We will then use the outcomes of this study and the extended study to develop a process modeling-specific version of the BWW representation model. This work will be divided into four steps. First, based on the BWW representation model, we will eliminate those constructs that seem to be of no or limited relevance for process modeling. Second, some BWW constructs may need to be renamed so that they better reflect common terminology in the domain of process modeling (for example, *activity* instead of *transformation*). Third, we will extend the BWW representation model by specializing those constructs that are perceived as having too high a level of granularity. Fourth, we may, in exceptional cases, introduce new constructs.

In a related stream of research, the outcome of this research is also used to continue work on a weighted scoring model for the interpretation of the levels of criticality of the results of representational analyses [14, 37].

References

1. Gartner Group: Delivering IT's Contribution: The 2005 CIO Agenda. Gartner, Inc, Stamford, Connecticut (2005)
2. Petri, C.A.: Fundamentals of a Theory of Asynchronous Information Flow. In: Popplewell, C.M. (ed.): IFIP Congress 62: Information Processing. North-Holland, Munich, Germany (1962) 386-390
3. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do Practitioners Use Conceptual Modeling in Practice? Data & Knowledge Engineering (In Press)
4. 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
5. Wand, Y., Weber, R.: An Ontological Model of an Information System. IEEE Transactions on Software Engineering 16 (1990) 1282-1292

6. Wand, Y., Weber, R.: On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Journal of Information Systems 3 (1993) 217-237
7. Wand, Y., Weber, R.: On the Deep Structure of Information Systems. Information Systems Journal 5 (1995) 203-223
8. Bunge, M.A.: Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World. Kluwer Academic Publishers, Dordrecht, The Netherlands (1977)
9. Shanks, G., Tansley, E., Weber, R.: Using Ontology to Validate Conceptual Models. Communications of the ACM 46 (2003) 85-89
10. Green, P., Rosemann, M.: Applying Ontologies to Business and Systems Modeling Techniques and Perspectives: Lessons Learned. Journal of Database Management 15 (2004) 105-117
11. Weber, R.: Ontological Foundations of Information Systems. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia (1997)
12. Chisholm, R.M.: A Realistic Theory of Categories: An Essay on Ontology. Cambridge University Press, Cambridge, Massachusetts (1996)
13. Guizzardi, G., Herre, H., Wagner, G.: On the General Ontological Foundations of Conceptual Modeling. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.): Conceptual Modeling - ER 2002. Lecture Notes in Computer Science, Vol. 2503. Springer, Tampere, Florida (2002) 65-78
14. Davies, I., Green, P., Milton, S., Rosemann, M.: Analysing and Comparing Ontologies with Meta Models. In: Krogstie, J., Halpin, T., Siau, K. (eds.): Information Modeling Methods and Methodologies. Idea Group, Hershey, Pennsylvania (2005) 1-16
15. Söderström, E., Andersson, B., Johannesson, P., Perjons, E., Wangler, B.: Towards a Framework For Comparing Process Modelling Languages. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.): 14th International Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science, Vol. 2348. Springer, Toronto, Canada (2002) 600-611
16. Keen, C.D., Lakos, C.: Analysis of the Design Constructs Required in Process Modelling. In: Purvis, M. (ed.): Proceedings of the International Conference on Software Engineering: Education and Practice. IEEE Computer Society, Dunedin, Ireland (1996) 434-441
17. van Griethuysen, J.J.: Concepts and Terminology for the Conceptual Schema and the Information Base. ISO/TC97/SC5 Report N695. International Organization for Standardization, Geneva, Italy (1982)
18. Tardieu, H.: Issues for Dynamic Modelling through Recent Development in European Methods. In: Sol, H.G., Crosslin, R.L. (eds.): Dynamic Modelling of Information Systems II. North-Holland, Amsterdam, The Netherlands (1992) 3-23
19. Gane, C., Sarson, T.: Structured Systems Analysis: Tools and Techniques. Prentice-Hall, Englewood Cliffs, California (1979)
20. Mayer, R.J., Menzel, C.P., Painter, M.K., de Witte, P.S., Blinn, T., Perakath, B.: Information Integration For Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Interim Technical Report AL-TR-1995-XXXX. Logistics Research Division, College Station, Texas (1995)
21. Keen, C.D., Lakos, C.: Information Systems Modelling using LOOPN++, an Object Petri Net Scheme. In: Sol, H.G., Verbraeck, A., Bots, P.W.G. (eds.): Proceedings of the 4th International Working Conference on Dynamic Modelling and Information Systems. Delft University Press, Noordwijkerhout, The Netherlands (1994) 31-52
22. Green, P., Rosemann, M.: Integrated Process Modeling. An Ontological Evaluation. Information Systems 25 (2000) 73-87
23. Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)" (in German: Semantic Modeling of Processes based on Event-driven Process Chains). Working Paper 89. Institut für Wirtschaftsinformatik der Universität Saarbrücken, Saarbrücken, Germany (1992)

24. Green, P., Rosemann, M.: Ontological Analysis of Integrated Process Models: Testing Hypotheses. The Australian Journal of Information Systems 9 (2001) 30-38
25. Green, P., Rosemann, M., Indulska, M.: Ontological Evaluation of Enterprise Systems Interoperability Using ebXML. IEEE Transactions on Knowledge and Data Engineering 17 (2005) 713-725
26. OASIS: ebXML Business Process Specification Schema Version 1.01. UN/CEFACT and OASIS (2001), available at: http://www.ebxml.org/specs/ebBPSS.pdf
27. Green, P., Rosemann, M., Indulska, M., Manning, C.: Candidate Interoperability Standards: An Ontological Overlap Analysis. Technical Report University of Queensland, Brisbane, Australia (2004)
28. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services. Version 1.1. BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG and Siebel Systems (2003), available at: http://xml.coverpages.org/BPELv11-May052003Final.pdf
29. Arkin, A.: Business Process Modeling Language. BPMI.org, (2002), available at: http://www.bpmi.org/
30. Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacsi-Nagy, P., Trickovic, I., Zimek, S.: Web Service Choreography Interface (WSCI) 1.0. BEA Systems, Intalio, SAP, Sun Microsystems (2002), available at: http://www.w3.org/TR/wsci/
31. Opdahl, A.L., Henderson-Sellers, B.: Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model. Software and Systems Modeling 1 (2002) 43-67
32. Soffer, P., Golany, B., Dori, D., Wand, Y.: Modelling Off-the-Shelf Information System Requirements. An Ontological Approach. Requirements Engineering 6 (2001) 183-199
33. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77 (1989) 541-580
34. BPMI.org, OMG: Business Process Modeling Notation Specification. Final Adopted Specification. Object Management Group (2006), available at: http://www.bpmn.org
35. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In: Campbell, B., Underwood, J., Bunker, D. (eds.): Proceedings of the 16th Australasian Conference on Information Systems. Australasian Chapter of the Association for Information Systems, Sydney, Australia (2005)
36. Soffer, P., Wand, Y.: On the Notion of Soft-Goals in Business Process Modeling. Business Process Management Journal 11 (2005) 663-679
37. Rosemann, M., Green, P., Indulska, M.: A Reference Methodology for Conducting Ontological Analyses. In: Lu, H., Chu, W., Atzeni, P., Zhou, S., Ling, T.W. (eds.): Conceptual Modeling – ER 2004. Lecture Notes in Computer Science, Vol. 3288. Springer, Shanghai, China (2004) 110-121
38. Rosemann, M., Green, P., Indulska, M.: A Procedural Model for Ontological Analyses. In: Hart, D., Gregor, S. (eds.): Information Systems Foundations: Constructing and Criticising. ANU E Press, Canberra, Australia (2005) 153-163
39. Rosemann, M., Green, P.: Developing a Meta Model for the Bunge-Wand-Weber Ontological Constructs. Information Systems 27 (2002) 75-91

Appendix

Appendix 1. Constructs in the BWW representation model, assigned to cluster groups. Adapted from [6, 11] with minor modifications.

BWW Construct	*Cluster*	*Description and Explanation*
THING	Things including properties and types of things	A thing is the elementary unit in the BWW ontological model. The real world is made up of things. Two or more things (composite or simple) can be **associated** into a **composite** thing.
PROPERTY in general in particular hereditary emergent intrinsic non-binding mutual binding mutual Attributes		Things possess properties. A property is modeled via a function that maps the thing into some value. For example, the attribute "weight" represents a property that all humans possess. In this regard, weight is an attribute standing for a property **in general**. If we focus on the weight of a specific individual, we would be concerned with a property **in particular**. A property of a composite thing that belongs to a component thing is called a **hereditary** property. Otherwise it is called an **emergent** property. Some properties are inherent properties of individual things. Such properties are called **intrinsic**. Other properties are properties of pairs or many things. Such properties are called **mutual**. **Non-binding mutual** properties are those properties shared by two or more things that do not "make a difference" to the things involved; e.g. order relations or equivalence relations. By contrast, **binding mutual** properties are those properties shared by two or more things that do "make a difference" to the things involved. **Attributes** are the names that we use to represent properties of things.
CLASS		A class is a set of things that can be defined via their possessing a single property.
KIND		A kind is a set of things that can be defined only via their possessing two or more common properties.
STATE	States assumed by things	The vector of values for all property functions of a thing is the state of the thing.
CONCEIVABLE STATE SPACE		The set of all states that the thing might ever assume is the conceivable state space of the thing.
LAWFUL STATE SPACE		The lawful state space is the set of states of a thing that comply with the state laws of the thing.
STATE LAW		A state law restricts the values of the properties of a thing to a subset that is deemed lawful because of natural laws or human laws.
STABLE STATE		A stable state is a state in which a thing, subsystem, or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event).
UNSTABLE STATE		An unstable state is a state that will be changed into another state by virtue of the action of transformations in the system.
HISTORY		The chronologically-ordered states that a thing traverses in time are the history of the thing.
EVENT	Events and transformations occurring on things	A change in state of a thing is an event.
CONCEIVABLE EVENT SPACE		The event space of a thing is the set of all possible events that can occur in the thing.
LAWFUL EVENT SPACE		The lawful event space is the set of all events in a thing that are lawful.
EXTERNAL EVENT		An external event is an event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment on the thing, subsystem, or system.
INTERNAL EVENT		An internal event is an event that arises in a thing, subsystem, or system by virtue of lawful transformations in the thing, subsystem, or system.
WELL-DEFINED EVENT		A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known.
POORLY DEFINED EVENT		A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known.
TRANSFORMATION		A transformation is a mapping from one state to another state.
LAWFUL TRANSFORMATION stability condition corrective action		A lawful transformation defines which events in a thing are lawful. The **stability condition** specifies the states that are allowable under the transformation law. The **corrective action** specifies how the values of the property functions must change to provide a state acceptable under the transformation law.
ACTS ON		A thing acts on another thing if its existence affects the history of the other thing.
COUPLING binding mutual property		Two things are said to be coupled (or interact) if one thing acts on the other. Furthermore, those two things are said to share a **binding mutual property** (or relation).
SYSTEM	Systems structured around things	A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets.
SYSTEM COMPOSITION		The things in the system are its composition.
SYSTEM ENVIRONMENT		Things that are not in the system but interact with things in the system are called the environment of the system.
SYSTEM STRUCTURE		The set of couplings that exist among things within the system, and among things in the environment of the system and things in the system is called the structure.
SUBSYSTEM		A subsystem is a system whose composition and structure are subsets of the composition and structure of another system.
SYSTEM DECOMPOSITION		A decomposition of a system is a set of subsystems such that every component in the system is either one of the subsystems in the decomposition or is included in the composition of one of the subsystems in the decomposition.
LEVEL STRUCTURE		A level structure defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.