

# Process Modeling – A Maturing Discipline?

Michael Rosemann<sup>1</sup>, Jan Recker<sup>1</sup>, Marta Indulska<sup>2</sup>, Peter Green<sup>2</sup>

<sup>1</sup> Centre for Information Technology Information  
Queensland University of Technology  
Brisbane, Australia

{m.rosemann, j.recker}@qut.edu.au

<sup>2</sup> 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 since Petri nets were introduced in 1962. One way of evaluating and comparing the scope and completeness of these techniques is by way of ontological analysis, a method which has gained popularity in the last decade. The purpose of this paper is to gauge the level of maturity of process modeling techniques over the last four decades of development. The basis of the comparison is the Bunge-Wand-Weber set of models, a popular benchmark used for the analysis of grammars that purport to model the real world and the interactions within it. This paper presents the outcomes of a comparison of ontological analyses of the most popular process modeling techniques. This research has two main outcomes. First, it provides insights, within the boundaries of an ontological analysis, into the extent to which process modeling techniques are maturing over time. Second, the findings also indicate areas in which the underlying reference ontology seems to be over-engineered. These latter findings will be the basis of the development of a Business Process Modeling-specific ontology that is the core of our future research in this area.

## 1 Introduction

Business Process Management (BPM) is now widely accepted as an approach that is foundational for many organizational and IT-driven projects. While the general objectives and methods of BPM are not new, BPM has only recently received a significant amount of attention and is now perceived to be the main business priority [1]. However, the actual modeling of business processes still presents major challenges for organizations and practitioners. 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 business modeling techniques is Event-driven Process Chains (EPC). Other business modeling techniques have evolved from requirements engineering techniques such as IDEF or even UML [2]. Second,

business process models are also used during process automation which requires them to be translated 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 [3], and process modeling has become one of the most popular reasons for conceptual modeling [4]. Consequently, a competitive market is providing a plethora of tools and methods. Gartner Research, for example, identified thirty-six vendors of process modeling solutions [5]. Clearly, there is a need for theory to assist the evaluation and comparison of available process modeling techniques. Given the existence of such theory, it is not only possible to evaluate these techniques, but also to determine if the discipline of process modeling overall is actually building on previous knowledge and if new proposed techniques really denote an improvement.

Our research in this area is motivated in several ways:

1. to provide theoretical guidance in the evaluation and comparison of available techniques;
2. to propose a measure of maturity of BPM over time;
3. to highlight representations that BPM 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 evaluation of these techniques we selected ontological completeness defined as the coverage of constructs as proposed by the Bunge-Wand-Weber (BWW) ontology. The study of modeling technique maturity is based on our review of previous BWW analyses of process modeling techniques. In order to report on a reasonably complete set of modeling techniques, we conducted our own analysis of two additional prominent modeling techniques, *viz.* Petri nets and BPMN. The importance of including an analysis of Petri nets in our study stems from the influence of the technique on a number of other modeling techniques. BPMN, on the other hand, was chosen as it denotes a most recently proposed and emerging standard for process modeling. Overall, this paper considers twelve common process modeling techniques and extracts the similarities and differences between 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 ontology is also evaluated in terms of appropriateness of its specification within the BPM domain.

This paper is structured as follows. The next section provides an overview of the Bunge-Wand-Weber ontology and its previous applications in the evaluation of process modeling techniques. We complement the existing work by conducting additional ontological analysis of Petri nets and BPMN as two prominent examples for process modeling. Section 3 reports on, and discusses, the findings of the comparison of process modeling techniques from the viewpoint of their ontological completeness. The paper ends in section 4 with conclusions, limitations, and directions for our future research.

## 2 Related Work & Background

### 2.1 Ontologies in Information Systems

Over the last few decades numerous conceptual modeling techniques, used to define requirements for building information systems, have emerged with no consistent theoretical foundation underlying their conception or development. Concerned that this situation would result in the development of information systems that were unable to completely capture important aspects of the real world, Wand and Weber [6-10] developed and refined a set of models for the evaluation of modeling techniques and the scripts prepared using such techniques. These models are based on an ontology defined by Bunge [11] and are referred to as the Bunge-Wand-Weber (BWW) models. *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 [12]. As computerized information systems are representations of real world systems, Wand and Weber suggest that ontology can be used to help define and build information systems that contain the necessary representations of real world constructs. The BWW representation model is one of three theoretical models defined by Wand and Weber [8] that make up the BWW models. Its application to information systems foundations has been referred to by a number of researchers [13] and is now often referred to as simply the BWW model. Some minor alterations have been carried out over the years by Wand and Weber [9, 10] and Weber [14], 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 of ontological constructs and clusters).

Weber [14] suggests that the BWW model can be used to analyze a particular modeling technique to make predictions on the modeling strengths and weaknesses of the technique. He clarifies two main evaluation criteria that may be studied according to the BWW model: *Ontological Completeness* and *Ontological Clarity* [14]. 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 ontological constructs as proposed in the BWW representation model.

### 2.2 Previous Ontological Analyses of Process Modeling Techniques

The BWW representation model has been used in over twenty-five research projects for the evaluation of different modeling techniques (see [24] for an overview). In this section, we briefly summarize those studies that involved the analysis of a process modeling technique.

Keen and Lakos [15] determined essential features for a process modeling scheme by evaluating six process modeling techniques in a historical sequence through the help of the BWW representation model. Their evaluation is restricted to the assessment of the ontological completeness of each technique. 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) [16], the Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise (MERISE) [17], the Data Flow Diagram (DFD) notation, the Integrated Definition Method 3 Process Description Capture Method (IDEF3) [18], and the Language for Object-Oriented Petri nets (LOOPN++) [19]. From their analysis, Keen and Lakos concluded that, in general, the BWW ontology facilitates the interpretation and comparison of process modeling techniques. They propose the ontological constructs of *system*, *system composition*, *system structure*, *system environment*, *transformation*, and *coupling* to be the 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 in this work.

Green and Rosemann [20] analyzed the Event-Driven Process Chain (EPC) notation with the help of the BWW ontology [21], focusing on both ontological completeness and clarity. Green and Rosemann's [22] findings have been empirically validated through interviews and surveys. 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.* [24] also examined the ebXML (Electronic Business using eXtensible Markup Language) Business Process Specification Schema version 1.1 [23] in terms of its ontological completeness and clarity. An empirical validation of the propositions is in its early stages. As ebXML denotes an executable interoperability standard for business process specification, it provides a wider range of language constructs for implementation requirements. This situation is clearly indicated through its relative high degree of ontological completeness.

Green *et al.* [25] compared different modeling standards for enterprise system interoperability, including Business Process Execution Language for Web Services v1.1 (BPEL4WS) [26], Business Process Modeling Language v1.0 (BPML) [27], Web Service Choreography Interface v1.0 (WSCI) [28], and ebXML version 1.1 [23]. All these standards that proclaim to allow for specification and execution of intra- and inter-organizational business processes, have been analyzed in terms of their ontological completeness. In addition, a minimal ontological overlap (MOO) analysis [10, 14] 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. At the present point in time, this analysis too, has not yet been validated through empirical tests. Steps towards empirical testing are currently underway.

While there has been further work that concentrates on the ontological evaluation of process modeling techniques, these particular process modeling techniques are not included in our research (see, for example, [29-31]). We have not considered those process modeling techniques that have different or extended requirements regarding their expressiveness. 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 have different requirements in terms of expressive power and are, therefore, limited in comparability to ‘pure’ process modeling notations. We believe that the inclusion of such techniques would severely limit the comparability of the results to process modeling languages, which focus on the control flow.

### 2.3 Ontological Analysis of Petri nets and BPMN

While the previous ontological analyses of process modeling techniques (see section 2.2) 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 ontological analysis of *Petri nets* in its original and most basic form [3], 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 extensive mathematical analyses and means to be directly executed [32]. Due to this rigorous, yet flexible, specification we found that although the notation originally merely consists of seven constructs, its ontological completeness can nevertheless be seen as quite high. While this apparent flexibility in 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 thoroughly define and demarcate the modeled system, which in turn causes potential 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 also important to point out that the same flexibility that affords Petri nets a higher ontological completeness, also results in extensive construct overload [14] that can potentially result in ambiguity of the models as extra-model knowledge is required to fully understand the modeling context of a particular construct in a given model situation.

The *Business Process Modeling Notation (BPMN)* [33] is a recently proposed standard that stemmed from the demand for a graphical notation that complements the BPEL4WS standard for executable business processes. It is the intention of the BPMN designers, however, to develop a modeling technique that can be applied for typical business modeling activities as well. 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.<sup>1</sup>

The BPMN specification defines thirty-eight distinct language constructs plus attributes. For example, nine distinct event types and three different event dimensions are included. Our analysis shows that the specification has a relatively high degree of ontological completeness. However, BPMN is not ontologically complete. For

---

<sup>1</sup> A detailed explanation of the ontological analyses of Petri nets and BPMN, including the reasoning behind the individual classifications, is available from the authors on request.

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 Ontological Analyses

#### 3.1 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 there are many process modeling techniques available that 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 notations capable of description and execution (e.g. ebXML, BPEL4WS).

As the prior analyses were independently conducted by four different research groups (see section 2.2) and as the ontological analyses referred to varied research purposes, significant 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 interpretation-wise. However, due to varying sets of BWW representation model constructs included in the analysis, we had to generalize the following specialized constructs of the BWW model:

- As some research groups did not entirely differentiate between property types we generalized all property-related sub-types to the super-type ‘property’. Therefore, if a mapping was found between a process modeling construct and a sub-type of ‘property’, then the mapping was recorded as belonging to the super-type ‘property’.
- Similarly, as some research groups did not consider the constructs of *stability condition* and *corrective action* in the context of the *lawful transformation* construct, we again generalized any mapping of these two constructs to a mapping of the *lawful transformation* construct.
- As the ontological construct *process* used in [21, 24] has not been specified in the representation model as defined in [8-10, 14] we did not consider it in our comparison.

The results of our comparison are illustrated in Table 1. In this table, each tick indicates that the specified ontological construct can be represented by the analyzed technique.

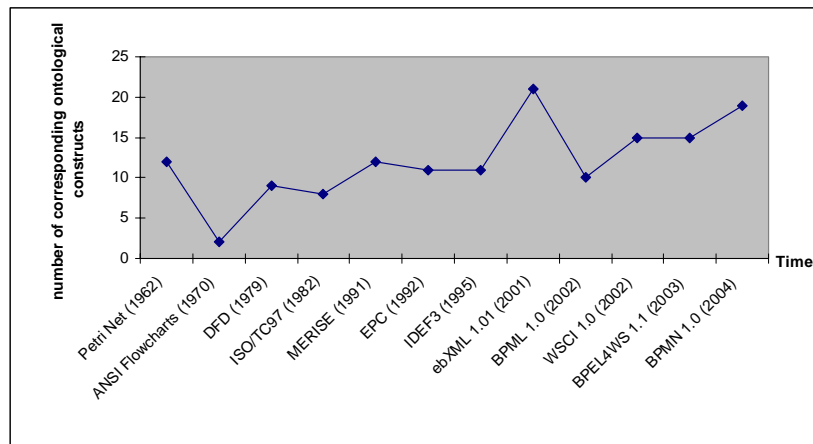
**Table 1.** Comparison of ontological analyses of process modeling techniques

Language	Petri Net	ANSI Flow-charts	DFD	ISO TC87	Merise	EPC	IDEF <sup>3</sup>	ebXML 1.1	BPML 1.0	WSCI 1.0	BPEL4WS 1.1	BPMN 1.0
Version	1962	1970	1979	1982	1992	1992	1995	2001	2002	2002	2003	2004
Year	1962	1970	1979	1982	1992	1992	1995	2001	2002	2002	2003	2004
Ontological Construct												
THING	✓		✓	✓	✓		✓					✓
PROPERTY			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CLASS	✓							✓	✓	✓	✓	✓
KIND												✓
STATE	✓					✓	✓	✓	✓	✓	✓	
CONCEIVABLE STATE SPACE								✓				
STATE LAW	✓			✓	✓	✓		✓				
LAWFUL STATE SPACE	✓							✓				
EVENT	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
CONCEIVABLE EVENT SPACE								✓				
LAWFUL EVENT SPACE								✓				
TRANSFORMATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LAWFUL TRANSFORMATION	✓			✓	✓	✓		✓	✓	✓	✓	✓
HISTORY								✓				
ACTS ON	✓							✓		✓	✓	✓
COUPLING		✓	✓		✓		✓	✓		✓	✓	✓
SYSTEM			✓		✓		✓	✓		✓	✓	✓
SYSTEM COMPOSITION			✓		✓		✓			✓	✓	✓
SYSTEM ENVIRONMENT			✓									✓
SYSTEM STRUCTURE					✓		✓			✓	✓	
SUBSYSTEM								✓				✓
SYSTEM DECOMPOSITION			✓				✓					✓
LEVEL STRUCTURE			✓			✓	✓					✓
STABLE STATE						✓		✓				
UNSTABLE STATE	✓							✓				
EXTERNAL EVENT				✓	✓	✓			✓	✓	✓	✓
INTERNAL EVENT	✓			✓	✓	✓		✓	✓	✓	✓	✓
WELL-DEFINED EVENT	✓					✓		✓	✓	✓	✓	✓
POORLY-DEFINED EVENT								✓	✓	✓	✓	✓

### 3.2 Issues in Process Modeling Techniques: Some Findings

The consolidation of previous ontological analyses with our own analyses of Petri nets and BPMN leads to several interesting results. A longitudinal study of the

ontological completeness of the analyzed techniques shows an obvious increase in the coverage of the ontological constructs. Fig. 1 visualizes this trend over time as measured by the number of ontological constructs covered by each analyzed technique, listed in chronological order (the line connecting the individual points does not have any (statistical) meaning in the diagram below, it is used for illustration purposes only). From this visualization we can see that the original Petri net specification already provided reasonably good coverage of the BWW representation model constructs. In general, over time, the maturity continuously increased. A noticeable spike in Fig. 1 depicts the high level of maturity of the ebXML standard. It is also worthwhile to note that the most recent standard, BPMN, performs very well from the viewpoint of ontological completeness. This higher level of ontological completeness can perhaps partly be explained by the fact that previous approaches including, Event-driven Process Chains and Petri nets, heavily influenced the development of the BPMN specification.



**Fig. 1.** Comparison of representation mapping analyses

Given the results obtained from the inspection and comparison of the analyses of the chosen process modeling techniques, we are able to extract a number of interesting observations.

The ontological analysis of Petri nets confirms that it is a very flexible, but potentially confusing notation that can be applied to many areas and situations. This flexibility, and potential confusion, is evidenced by the high level of construct overload that occurs when one grammatical construct can be mapped to a number of ontological constructs. Such a situation indicates that the constructs in the original Petri net specification can be interpreted in a number of ways. For example, a *place* construct in a Petri net can be used to represent a *thing*, *class*, or *state*. Such flexibility, however, while seemingly an advantage, can also result in a model that is not easily interpreted by a user. This weakness is attributed to the required extra-model knowledge that may be necessary in order to understand whether a *place* in a given model represents a thing, a class of things, or a state.

ebXML appears to be the most ontologically complete process specification to date. It is interesting to note that ebXML is specified in UML, with a semi-formal construct definition and description. This situation contrasts to the latest techniques. For example, BPEL4WS, WSCI, and BPMN are specified in a textual manner with additional diagrams of examples. As such, the ebXML specification is less subjective in its possible interpretations.

The BPMN standard appears to be quite mature. This fact is not only evidenced by the number of identified ontological mappings, but also by the specialization of the constructs. For example, BPMN has sub-types of *event* and *transformation* that allow a more rigorous model to be defined. This strength can potentially also be its weakness as the varied sub-types of *event* and *transformation* will require thorough understanding by the user in order to represent the right types of events/transformations appropriately.

In general terms, it appears that techniques that focus on describing process flows from a semantic, (business) perspective (e.g., DFD, IDEF<sub>3</sub>) are not as ontologically complete as their counterparts that provide more syntactical rigor due to their focus on executability (e.g., ebXML, BPMN).

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

**Table 2.** Analysis of construct occurrences

Most supported ontological constructs		Least supported ontological constructs	
Construct	Occurrence ratio (%)	Construct	Occurrence ratio (%)
TRANSFORMATION	100,00%	HISTORY	8%
PROPERTY	83%	LAWFUL EVENT SPACE	8%
EVENT	83%	CONCEIVABLE EVENT SPACE	8%
LAWFUL TRANSFORMATION	75%	CONCEIVABLE STATE SPACE	8%
INTERNAL EVENT	75%	KIND	8%
COUPLING	67%	UNSTABLE STATE	17%
STATE	58%	STABLE STATE	17%
SYSTEM	58%	SUBSYSTEM	17%
EXTERNAL EVENT	58%	SYSTEM ENVIRONMENT	17%
WELL-DEFINED EVENT	58%	LAWFUL STATE SPACE	17%

As can be expected in a BPM domain, each of the analyzed techniques has the ability to represent the ontological construct *transformation* – one of the core concepts in process modeling. 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 interesting to note however, 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.

Also interesting is the divided support for ontological constructs *thing*, *class*, and *acts-on*. Each of these constructs is represented in fifty percent of the analyzed process modeling techniques. Closer inspection of Table 1 shows that while earlier process modeling techniques provided a construct for a specific *thing*, the recent standards have representation capability for *class* 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.

Throughout the BPM domain, a lack in the support of business rule definitions can be observed. As *conceivable* and *lawful event spaces* as well as *states* are under-represented – none of these constructs has support of more than seventeen percent – state and transformation modeling is unclear for the modeler who thus may encounter confusion when determining which set of events and states can be expected to occur in the system and which events and states can occur but should not be allowed.

Also, there appears to be inconsistent support for systems structured around things. From the list of seven ontological constructs grouped in this cluster, five have been found to be represented in under thirty-five 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 the light of collaborative business processes and interoperability.

### 3.3 Potential Issues with the Underlying Ontology

Our findings from the longitudinal analysis of process modeling techniques are aligned with some of the criticisms of ontological analyses [34, 35].

Regarding *lack of relevance* and *understandability* of the ontology, we suggest the development and utilization of a more specialized ontology for the domain of BPM. The current BWW representation model needs to be investigated in order to determine areas that need further specialization, extension, deletion, or renaming. The following findings from this research support our argument.

Regarding *events* and *transformations* occurring on *things*, we believe that this cluster needs further specialization. For instance, BPMN distinguishes between nine event types, representing a differentiation scheme that is not covered by the ontological construct of *event*. The same can be seen in standards such as ebXML, BPEL4WS, BPML, and WSCI for example. The same 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 a transformation. This situation implies that, just as ‘properties’ in the BWW representation model are specialized, perhaps transformations should be as well for the domain of BPM. A similar situation exists in standards like BPEL4WS and ebXML, for example.

Also, it is interesting to note that throughout all 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 ontology. However, these constructs are agreed to be essential to the BPM

domain. Consequently, we are considering how the ontology might be extended to better reflect such control flow concepts important to the BPM domain.

Taking a methodological viewpoint to ontological analyses, we found the lack of objectivity issue to persist. This situation highlights the need for the use of meta-models in conducting analyses. A meta-model allows for a clearer description of the ontology 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 [36] for such a purpose and its use has been promoted and discussed in [35, 37].

#### 4 Conclusions, Limitations, Future Research

This paper presents the first comprehensive study that longitudinally compares previous ontological analyses of process modeling techniques. The innovative comparative study also includes the outcomes of the initial ontological analyses of Petri nets and, importantly, the new proposed modeling standard BPMN. The *findings* clearly show signs of a maturing discipline, as measured by an increased ontological completeness of process modeling techniques over time. The outcomes are of interest not only to the developers of such process modeling techniques, but also to users of these techniques. Developers of process modeling techniques should be motivated to examine ontological analyses of currently used process modeling tools in order to build upon these 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 criteria for the selection of the most appropriate modeling technique. The results also help to 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 model as a benchmark for such analyses of modeling techniques. As shown in Table 1, there are some ontological constructs of the BWW representation model that are supported by only one technique of the chosen twelve. For example, the ontological construct *kind* can only be represented by BPMN, and the ontological constructs of *history, conceivable state space* and *conceivable event space* can only be represented by ebXML. While this might indicate an improvement of the representation power of process modeling techniques, it might also indicate that, perhaps, the particular ontological construct is not necessary for modeling in the domain of BPM. Such issues require further empirical testing in order to determine whether the ontology requires pruning and specialization or whether other techniques should also be able to model what is represented by the ontological construct. Such research might also motivate other researchers to conduct a similar study for data or object-oriented modeling techniques.

We identify four *limitations* in our research. First, we base our study on previous and independent ontological analyses. These studies have been conducted by different researchers and the analyses results have not been validated by us. The actual process

of conducting an ontological analysis, however, is exposed to the impact of the subjective interpretations of the researcher [35]. As such, there are issues related to the comparability of the analyses. Second, we constrained the considered ontological analyses to analyses based on the BWW models. While this limits the generalization of the results, it increases the focus of the work and avoids the necessity to translate between different ontologies. Third, our own ontological analyses of Petri nets and BPMN have not yet been empirically tested, although this work is currently in process. Fourth, we limit our research to those process modeling techniques that have been previously analyzed from the viewpoint of the BWW models. While we believe that the selected set of techniques is representative of the most popular techniques, it can by no means claim to be complete.

In our future research, we will utilize the outcomes of this study to develop a more process-specific version of the BWW representation model. This work will be divided into four steps. First, based on the original BWW representation model, we will eliminate those ontological constructs that seem to be of no or limited relevance in the context of process modeling. Potential candidates might be *kind*, *lawful event space*, and, *conceivable state space*, for example. However, the list of superfluous constructs will be verified through empirical testing. Second, we may find the need to rename certain ontological constructs so that they better reflect common terminology in the domain of process modeling (for example, an *activity* construct instead of a *transformation* construct). Third, we will extend the BWW representation model by specializing those ontological constructs that are perceived as having too high a level of granularity. Fourth, we may, in exceptional cases, introduce new constructs. Moreover, we will seek, in a different stream of research, empirical insights for the identified ontological shortcomings of BPMN. This work is currently underway and will include interviews with BPMN users as well as discussions with the developers of BPMN.

## References

1. Gartner Group: Delivering IT's Contribution: The 2005 CIO Agenda. Gartner, Inc, Stamford (2005)
2. Penker, M., Eriksson, H.-E.: Business Modeling With UML: Business Patterns at Work. John Wiley & Sons, New York (2000)
3. Petri, C.A.: Fundamentals of a Theory of Asynchronous Information Flow. In: Proceedings of the 1962 IFIP Congress. North-Holland, Amsterdam (1962) 386-390
4. Davies, I., Green, P., Rosemann, M., Gallo, S.: Conceptual Modelling - What and Why in Current Practice. In: Lu, H., Chu, W., Atzeni, P., Zhou, S., Ling, T.W. (ed.): Proceedings of the 23rd International Conference on Conceptual Modelling. Lecture Notes in Computer Science, Vol. 3288. Springer, Shanghai (2004) 30-42
5. Gartner Group: Magic Quadrant for Business Process Analysis. Gartner, Inc, Stamford (2004)
6. Wand, Y., Weber, R.: An Ontological Evaluation of Systems Analysis and Design Methods. In: Falkenberg, E.D., Lindgreen, P. (ed.): Information System Concepts: An In-depth Analysis. Proceedings of the IFIP TC 8/WG 8.1 Working Conference on Information System Concepts. North Holland, Amsterdam (1989) 79-107

7. Wand, Y., Weber, R.: An Ontological Model of an Information System. *IEEE Transactions on Software Engineering* 16 (1990) 1282-1292
8. Wand, Y., Weber, R.: Towards a Theory of the Deep Structure of Information Systems. In: De Gross, J.I., Alavi, M., Oppelland, H. (ed.). *ACM Press, Copenhagen* (1990) 61-71
9. Wand, Y., Weber, R.: On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems* 3 (1993) 217-237
10. Wand, Y., Weber, R.: On the Deep Structure of Information Systems. *Information Systems Journal* 5 (1995) 203-223
11. Bunge, M.A.: *Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World*. Kluwer Academic Publishers, Dordrecht (1977)
12. Shanks, G., Tansley, E., Weber, R.: Using Ontology To Validate Conceptual Models. *Communications of ACM* 46 (2003) 85-89
13. Green, P., Rosemann, M.: Applying Ontologies to Business and Systems Modeling Techniques and Perspectives: Lessons Learned. *Journal of Database Management* 15 (2004) 105-117
14. Weber, R.: *Ontological Foundations of Information Systems*. Coopers & Lybrand, Melbourne (1997)
15. Keen, C., Lakos, C.: Analysis of the Design Constructs Required in Process Modelling. In: *Proceedings of the International Conference on Software Engineering: Education and Practice*. IEEE Computer Society, Dunedin (1996) 434-441
16. van Griethuysen, J.J.: *Concepts and Terminology for the Conceptual Schema and the Information Base*. International Organization for Standardization, Geneva (1982)
17. Tardieu, H.: Issues for Dynamic Modelling through Recent Development in European Methods. In: Sol, H.G., Crosslin, R.L. (ed.): *Proceedings of the Second International Working Conference: Dynamic Modelling of Information Systems II*. Elsevier, Washington D.C. (1991)
18. 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*. Armstrong Laboratory, Human Resources Directorate, Logistics Research Division, College Station (1995)
19. Keen, C., Lakos, C.: Information Systems Modelling using LOOPN++, an Object Petri Net Scheme. In: Sol, H.G., Verbraeck, A., Bots, P.W.G. (ed.): *Proceedings of the 4th International Working Conference on Dynamic Modelling and Information Systems*. Delft University Press, Noordwijkerhout (1994) 31-52
20. Keller, G., Nüttgens, M., Scheer, A.-W.: *Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)"*. Institut für Wirtschaftsinformatik der Universität Saarbrücken, Saarbrücken (1992)
21. Green, P., Rosemann, M.: Integrated Process Modeling. An Ontological Evaluation. *Information Systems* 25 (2000) 73-87
22. Green, P., Rosemann, M.: Perceived Ontological Weaknesses of Process Modelling Techniques: Further Evidence. In: Wrycza, S. (ed.): *Proceedings of the 10th European Conference on Information Systems*. Uniwersytetu Gdanskiego, Gdansk (2002) 312-321
23. OASIS: ebXML Business Process Specification Schema Version 1.01. UN/CEFACT and OASIS (2001), available at: <http://www.ebxml.org/specs/ebBPSS.pdf>
24. 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
25. Green, P., Rosemann, M., Indulska, M., Manning, C.: *Candidate Interoperability Standards: An Ontological Overlap Analysis*. Business School, University of Queensland, Brisbane (2004)
26. Andrews, T., Curbera, F., Dholakia, H., Golland, 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>
27. Arkin, A.: Business Process Modeling Language. BPMI.org, (2002), available at: <http://www.bpmi.org/>
28. Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacs-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/>
29. Opdahl, A.L., Henderson-Sellers, B., Barbier, F.: An Ontological Evaluation of the OML Metamodel. In: Falkenberg, E.D., Lyytinen, K., Verrijn-Stuart, A.A. (ed.): Proceedings of the FIP TC8/WG8.1 International Conference on Information System Concepts: An Integrated Discipline Emerging. IFIP Conference Proceedings, Vol. 164. Kluwer, Leiden (1999) 217-232
30. 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
31. Soffer, P., Golany, B., Dori, D., Wand, Y.: Modelling Off-the-Shelf Information System Requirements. An Ontological Approach. Requirements Engineering 6 (2001) 183-199
32. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77 (1989) 541-580
33. BPMI.org: Business Process Modeling Notation (BPMN). Version 1.0 - May 3, 20. BPMI.org, (2004), available at: <http://www.bpmi.org/>
34. Rosemann, M., Green, P., Indulska, M.: Towards an Enhanced Methodology for Ontological Analyses. In: Grabis, J., Persson, A., Stirna, J. (ed.): Proceedings of the 16th International Conference on Advanced Information Systems Engineering. Faculty of Computer Science and Information Technology, Riga (2004) 122-131
35. 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. (ed.): Proceedings of the 23rd International Conference on Conceptual Modeling. Lecture Notes in Computer Science, Vol. 3288. Springer, Shanghai (2004) 110-121
36. Rosemann, M., Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. Information Systems 27 (2002) 75-91
37. Davies, I., Green, P., Milton, S., Rosemann, M.: Using Meta Models for the Comparison of Ontologies. In: Krogstie, J., Halpin, T., Siau, K. (ed.): Proceedings of the 8th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design. Velden (2003)

## Appendix

**Appendix 1.** Ontological constructs in the BWW representation model, assigned to cluster groups, adapted from [9, 14] with minor modifications.

Ontological Construct	Cluster Group	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 <b>associated</b> into a <b>composite</b> thing.
PROPERTY: IN GENERAL	Things including properties and types	Things possess properties. A property is modeled via a function that maps the thing into

<p>IN PARTICULAR HEREDITARY EMERGENT INTRINSIC NON-BINDING MUTUAL BINDING MUTUAL ATTRIBUTES</p>	<p>of things</p>	<p>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 <b>in general</b>. If we focus on the weight of a specific individual, we would be concerned with a property <b>in particular</b>. A property of a composite thing that belongs to a component thing is called a <b>hereditary</b> property. Otherwise it is called an <b>emergent</b> property. Some properties are inherent properties of individual things. Such properties are called <b>intrinsic</b>. Other properties are properties of pairs or many things. Such properties are called <b>mutual</b>. <b>Non-binding mutual</b> 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, <b>binding mutual</b> properties are those properties shared by two or more things that do “make a difference” to the things involved. <b>Attributes</b> are the names that we use to represent properties of things.</p>
<p>CLASS</p>	<p>Things including properties and types of things</p>	<p>A class is a set of things that can be defined via their possessing a single property.</p>
<p>KIND</p>	<p>Things including properties and types of things</p>	<p>A kind is a set of things that can be defined only via their possessing two or more common properties.</p>
<p>STATE</p>	<p>States assumed by things</p>	<p>The vector of values for all property functions of a thing is the state of the thing.</p>
<p>CONCEIVABLE STATE SPACE</p>	<p>States assumed by things</p>	<p>The set of all states that the thing might ever assume is the conceivable state space of the thing.</p>
<p>STATE LAW</p>	<p>States assumed by things</p>	<p>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.</p>
<p>LAWFUL STATE SPACE</p>	<p>States assumed by things</p>	<p>The lawful state space is the set of states of a thing that comply with the state laws of the thing.</p>
<p>EVENT</p>	<p>Events and transformations occurring on things</p>	<p>A change in state of a thing is an event.</p>
<p>CONCEIVABLE EVENT SPACE</p>	<p>Events and transformations occurring on things</p>	<p>The event space of a thing is the set of all possible events that can occur in the thing.</p>
<p>LAWFUL EVENT SPACE</p>	<p>Events and transformations occurring on things</p>	<p>The lawful event space is the set of all events in a thing that are lawful.</p>
<p>TRANSFORMATION</p>	<p>Events and transformations occurring on things</p>	<p>A transformation is a mapping from one state to another state.</p>
<p>LAWFUL TRANSFORMATION STABILITY CONDITION CORRECTIVE ACTION</p>	<p>Events and transformations occurring on things</p>	<p>A lawful transformation defines which events in a thing are lawful. The <b>stability condition</b> specifies the states that are allowable under the transformation law. The <b>corrective action</b> specifies how the values of the property functions must change to provide a state acceptable under the transformation law.</p>

HISTORY	States assumed by things	The chronologically-ordered states that a thing traverses in time are the history of the thing.
ACTS ON	Events and transformations occurring on things	A thing acts on another thing if its existence affects the history of the other thing.
COUPLING BINDING MUTUAL PROPERTY	Events and transformations occurring on things	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 <b>binding mutual property</b> (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	Systems structured around things	The things in the system are its composition.
SYSTEM ENVIRONMENT	Systems structured around things	Things that are not in the system but interact with things in the system are called the environment of the system.
SYSTEM STRUCTURE	Systems structured around things	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	Systems structured around things	A subsystem is a system whose composition and structure are subsets of the composition and structure of another system.
SYSTEM DECOMPOSITION	Systems structured around things	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	Systems structured around things	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.
STABLE STATE	States assumed by things	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	States assumed by things	An unstable state is a state that will be changed into another state by virtue of the action of transformations in the system.
EXTERNAL EVENT	Events and transformations occurring on things	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	Events and transformations occurring on things	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	Events and transformations occurring on things	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	Events and transformations occurring on things	A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known.

---