Business Process Modeling:
A Maturing Discipline?

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

[1] Faculty of Information Technology
Queensland University of Technology
{j.recker, m.rosemann}@qut.edu.au

[2] UQ Business School
University of Queensland
{m.indulska, p.green}@business.uq.edu.au

ABSTRACT

A plethora of business process modeling techniques has been proposed over the last decades, creating a demand for theory to assist in the comparison and evaluation of these techniques. A widely established way of determining the effectiveness and efficiency of modeling techniques is by way of representational analysis. The purpose of this paper is to comparatively assess the outcomes of representational analyses of twelve popular process modeling techniques in order to provide insights into the extent to which the representational capabilities of process modeling techniques differ between each other and evolve over time, measured by the extent to which the techniques are able to facilitate complete and clear descriptions of real-world domains.

Our research shows that, over time, process modeling technique have indeed increased their scope of coverage; however, this increased effectiveness of the techniques comes at the extent of increased complexity and ambiguity in the use of these techniques.

Keywords: Business Process Management, Process Modeling, Representation Theory, BWW Model

I. INTRODUCTION

Business Process Management (BPM) has been identified as a top business priority, and building Business Process Capability is seen as a major challenge for senior executives in the coming years [Gartner Group, 2005]. The interest in BPM has, *inter alia*, triggered substantial academic and commercial work aiming towards advanced business process management solutions. One prominent example in this context is the increased popularity of business process modeling [Davies *et al.*, 2006]. Due to a strengthened interest in a more disciplined approach for business process management, many organizations have made significant investments in process modeling initiatives, which in turn has triggered substantial related research. Many studies have shown the relevance of process modeling to BPM initiatives, *e.g.*, [Davenport, 1993]. Process modeling denotes a requirement for a number of ISO 9000 quality programs [Ould, 1995] and is the basis of process-related IT implementations, such as Enterprise Resource Planning systems [Robinson and Dilts, 1999] and workflow management systems [Dumas *et al.*, 2005]. The recent introduction of legislative frameworks such as the Sarbanes-Oxley Act [Nielsen and Main, 2004] further contributed to the increasing interest in business process modeling as a way of capturing and graphically documenting the processes of an organization.

The ongoing and strengthened interest in modeling for business process management has given rise to a wide range of modeling techniques, spanning simple flowcharting techniques [American National Standards Institute, 1970], techniques initially used as part of requirements engineering such as UML [Fowler, 2004], dedicated business-oriented modeling approaches such as Event-driven Process Chains [Keller *et al.*, 1992], and also formalized and academically studied techniques such as Petri nets [Petri, 1962] and their dialects. Consequently, a competitive market is providing a large selection of techniques and tools for process modeling [Sinur, 2004] and significant demand has been created for means to evaluate and compare the available set of techniques [Moody, 2005].

In addition to a practical need for evaluation, from a scholarly perspective the continuing emergence of “yet another” process modeling technique leads to the question if there are actual signs of increasing maturity within the capabilities of process modeling techniques. We understand maturity of a process modeling technique as its capability to facilitate complete descriptions of relevant real-world domains while at the same time being clear in the usage of the language constructs provided. Increasing maturity across process modeling techniques would then be the improvement in the scope of domain coverage (the completeness of a technique) and the improvement of the clarity of the technique specification. Following this understanding we are able to answer the question whether a cumulative tradition of the process modeling discipline has been established, in particular whether more recent approaches to process modeling actually learn from previous experiences. Such a move would be a pre-requisite for an evolving research discipline that builds on the existing body of knowledge, has an awareness for the remaining open challenges, and is guided by a methodological procedure in its future research efforts [Keen, 1980, Weber, 1997]. This is particularly the case in Information Systems analysis and design where the analysis of the strengths and weaknesses of existing approaches can be used as the basis for developing new and improved techniques [Bubenko, 1986] and where thereby the ultimate goal of using applied research, to improve practice [Benbasat and Zmud, 1999], can be assisted.

The aim of this paper then is to study the differences in the representational capabilities across leading process modeling techniques and to gauge the development of the representational capabilities of process modeling techniques over time. As measurements for the study we selected the notions of ontological completeness and ontological clarity [Weber, 1997]. From these overall objectives we derived the following more detailed research questions:

1) How do process modeling techniques perform in light of a representational analysis based on the Bunge-Wand-Weber representation model?
2) What are common concepts and key differentiators of leading process modeling techniques, measured by their levels of ontological completeness and clarity as based on the Bunge-Wand-Weber representation model?
3) Are there signs of increasing maturity in the development of process modeling techniques over time, as measured by ontological completeness and ontological clarity across the techniques?

We proceed as follows. The next section provides an overview of the Bunge-Wand-Weber representation model and its previous applications in the evaluation of process modeling techniques. We complement the existing work by conducting additional representational analyses of Petri nets and BPMN as two prominent examples for process modeling techniques. Section III reports on, and discusses, the findings of the comparative assessment of process modeling techniques from the viewpoint of their ontological completeness and ontological clarity. In section IV we discuss the evolution of the representational capabilities of the considered process modeling techniques over time. The paper concludes in section V with a review of contributions and limitations of our study.

II. BACKGROUND & RELATED WORK

REPRESENTATIONAL ANALYSIS IN INFORMATION SYSTEMS

The ongoing proliferation of modeling techniques stands in sharp contrast to the paucity of rigorous research frameworks that can be used for evaluation. Yet, while in general the lack of established quality frameworks for conceptual modeling has repeatedly been commented as critical [Moody, 2005], reasonably mature research has emerged over the last years introducing the research method of *representational analysis*.

Representational analysis uses models of representation, such as the Bunge-Wand-Weber (BWW) representation model [Wand and Weber, 1990, 1993, 1995], as a benchmark for the evaluation of the representational capabilities of a modeling technique. In this paper we use the principles of representational analysis to comparatively assess the most popular process modeling techniques from the viewpoint of the BWW representation model.

The BWW representation model originates from the adaptation of a well-established ontology proposed by Bunge [1977]. Wand and Weber [1990, 1993, 1995] adapted the ontology into a theory of representation that is closer to the demands and terminology of the Information Systems community. While a number of existing models of representation can be used as part of representational analysis, the deployment of the BWW representation model in our study can be justified on at least three premises. First, unlike other conceptual modeling theories based on ontology, *e.g.*, [Chisholm, 1996, Cocchiarella, 1995], the BWW model has specifically been derived with the Information Systems discipline in mind [Weber, 1997]. Second, the

BWW model officiates as an upper ontology for the modeling of Information Systems [Evermann, 2005], and its foundational character and comprehensive scope allows for wide applicability. Third, there is an established track record of individual studies and a demonstrated usefulness of representational analyses of modeling techniques using the representation model [Green and Rosemann, 2004], which allows comparison of the results with other studies.

Building on the observation that, in their essence, Information Systems are representations of real world systems [Wand and Weber, 1995] and drawing on an ontological model, the BWW model specifies a number of constructs that are deemed necessary to provide faithful representations of Information Systems, and which therefore should be included in any conceptual modeling technique. These constructs can be represented in a meta model [Rosemann and Green, 2002] that shows several clusters of BWW constructs: 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). Rosemann and Green's proposed clustering will in our study serve as an analysis framework through which we assess the outcomes of the representational analyses.

The process of using the BWW model as a reference benchmark for the evaluation of the representational capabilities of a modeling technique forms the core of the research method of representational analysis, which can be used to make predictions on the modeling strengths and weaknesses of the technique, *viz.*, its capabilities to provide *complete* and *clear* descriptions of the domain being modeled. In this process, the constructs of the BWW representation model (*e.g.*, thing, event, transformation) are compared with the language constructs of the modeling technique (*e.g.*, event, activity, actor) in a bi-directional mapping. The basic assumption is that any deviation from a 1-1 relationship between the corresponding constructs in the representation model and the modeling technique leads to representational deficiency in the use of the technique, which potentially causes confusion to its users. These undesirable situations can be further categorized into four sub-types (see Figure 1), resulting in two main evaluation criteria that may be studied according to the BWW model [Weber, 1997]: *ontological completeness* and *ontological clarity*. *Ontological completeness* is indicated by the inverse degree of construct deficit (1:0), *i.e.*, the extent to which a process modeling technique covers completely the constructs proposed in the BWW representation model. On the other hand, *ontological clarity* is constituted by the degrees of construct overload (m:1), being the extent to which single language constructs cover several BWW constructs, construct redundancy (1:m), *i.e.*, the extent to which a single BWW construct maps to several language constructs, and construct excess (0:1), being the extent of language constructs that do not map to any BWW construct.

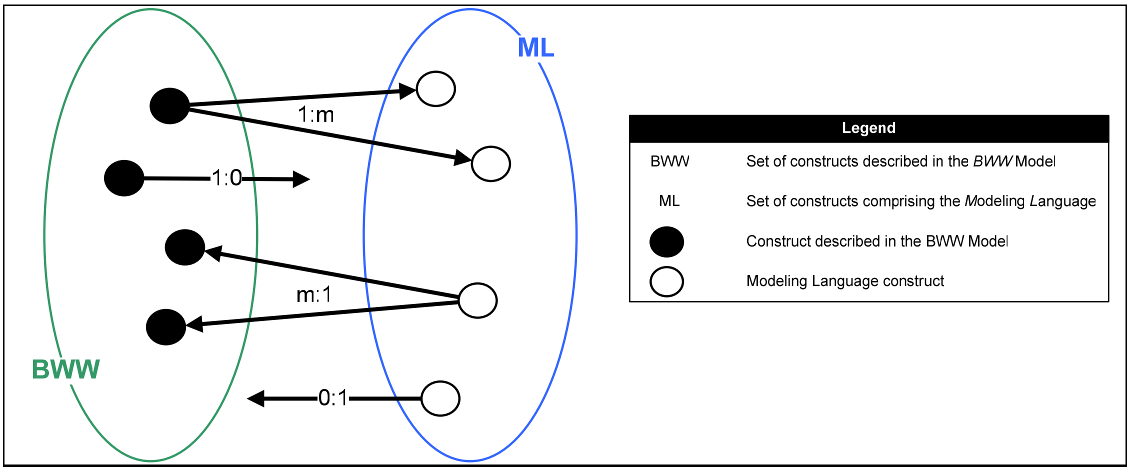

Figure 1. Potential representational deficiencies of a modeling technique. Adapted from [Weber, 1997]

The BWW model has over the years reached a significant level of maturity, adoption and dissemination, and has been used in over thirty research projects [Green and Rosemann, 2004] to evaluate a wide range of different techniques that are, for instance, used for data modeling [Wand and Weber, 1993], schema modeling [Weber and Zhang, 1996], object-oriented modeling [Opdahl and Henderson-Sellers, 2002], use case modeling [Irwin and Turk, 2005] and reference modeling [Fettke and Loos, 2003]. It also has a strong track record in the area of process modeling with contributions coming from various researchers. We will in the subsequent section briefly summarize those BWW related studies that focus specifically on process modeling techniques.

Regarding alternative ontologies that may form the basis for representational analysis of conceptual modeling in Information Systems, the approaches of Milton and Kazmierczak [2004], who rely on an ontology developed by Chisholm [1996], and Guizzardi [2005] are closest to the ideas of Wand and Weber. These upper-level ontologies have been built for similar purposes and seem to be equally expressive

[Davies *et al.*, 2005, Evermann, 2005]. However, these ontologies have not yet achieved the level of dissemination of the BWW model, which reasons our selection for this model as a benchmark for a comparative study.

PREVIOUS REPRESENTATIONAL ANALYSES OF PROCESS MODELING TECHNIQUES

A process model is typically a graphical depiction of at least the activities, events/states, and control flow logic that constitute a business process [Curtis *et al.*, 1992]. Additionally, many process models also include information regarding the involved data, organizational/IT resources and potentially other artifacts such as external stakeholders, performance metrics, etc. Process models in general 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 a business modeling technique used for such purposes is the Event-driven Process Chain (EPC). Second, business process models are used for process automation, which requires their conversion into executable specifications. Techniques used for depicting process models for this purpose have higher requirements in terms of expressive power. Examples include Petri nets or the Business Process Execution Language for Web Services (BPEL4WS).

Keen and Lakos [1996] determined essential features for a process modeling technique by using the BWW representation model to evaluate six process modeling techniques. Among the modeling techniques evaluated were: ANSI flowcharts [American National Standards Institute, 1970], Data Flow Diagrams (DFD) [Gane and Sarson, 1979], the IDEF Method 3 Process Description Capture Method [Mayer *et al.*, 1995] and their own Language for Object-Oriented Petri nets (LOOPN++). The evaluation was restricted to the assessment of the ontological completeness of each technique. 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 [2000] used the BWW model to analyze the Event-driven Process Chain (EPC) notation [Keller *et al.*, 1992, Scheer, 2000], assessing both ontological completeness and clarity. Empirically confirmed shortcomings were found in the EPC notation with regard to the representation of real world objects and business rules, and in the thorough demarcation of the analyzed process [Green and Rosemann, 2001].

Green *et al*. [2005] examined the Electronic Business using eXtensible Markup Language Business Process Specification Schema (ebXML BPSS) v1.01 [OASIS, 2001] in terms of ontological completeness and clarity. While the empirical validation of results has not yet been performed, the analysis indicates a relatively high degree of ontological completeness of ebXML.

Green *et al*. [2004] also compared different modeling standards for enterprise system interoperability, including Business Process Execution Language for Web Services v1.1 (BPEL4WS) [Andrews *et al.*, 2003], Business Process Modeling Language v1.0 (BPML) [Arkin, 2002], Web Service Choreography Interface v1.0 (WSCI) [Arkin *et al.*, 2002], and ebXML BPSS v1.01. These four standards, which proclaim to allow for specification of intra- and inter-organizational business processes, have been analyzed in terms of their ontological completeness and clarity. The study found that ebXML provides a wider range of language constructs for specification requirements than other techniques, indicated through its comparatively high degree of ontological completeness.

Furthermore, in preparation for this study, we conducted two more representational analyses (from the viewpoint of both ontological completeness and clarity) of process modeling techniques, namely Petri nets [Petri, 1962] and BPMN [BPMI.org and OMG, 2006b]. The importance of including an analysis of Petri nets in our study stems from the influence of this 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 backed by strong practitioner interest. A number of shortcomings, related to ontological completeness and clarity, in light of the BWW model were identified in terms of the use of these two techniques. For instance, Petri nets lack support for the modeling of systems structured around things and BPMN lacks capabilities to represent states assumed by things. The results have been empirically validated in the case of BPMN. We have summarized these analyses in form of a mapping table in Appendix 2. For details of the analyses of Petri nets and BPMN, as well as details of the empirical validation of the identified BPMN shortcomings, please refer to [Recker *et al.*, 2006, Rosemann *et al.*, 2006].

While there has been further work that uses the principles of representational analysis for studies on dynamic modeling techniques, see for instance [Irwin and Turk, 2005, Opdahl and Henderson-Sellers, 2002], these particular techniques are not included in our research. We have not considered those modeling techniques that have different or extended requirements regarding their expressiveness due to different design principles. 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, which are applied in software engineering rather than process management contexts, 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 limit the comparability of the results to 'regular' process modeling techniques.

III. COMPARISON OF REPRESENTATIONAL ANALYSES

RESEARCH DESIGN

While representational analysis of a process modeling technique provides means for exploring strengths and weaknesses of that technique, it can also be used for the *comparison* of various techniques, thereby allowing for a comparative assessment to highlight representational differences between the considered techniques. In order to extract common shortcomings and highlight main differentiating features between various process modeling techniques, we consolidated and compared analyses of twelve techniques with a focus on their ontological completeness and clarity. For each form of the representational deficiencies we constructed a table into which we mapped the results of the respective studies outlined in the previous section.

In performing the review and comparison, we were concerned with minimizing potential mapping errors and general subjective bias. The comparison was therefore accomplished as follows. Two researchers individually reviewed and compared the analyses of the selected techniques. The results were later consolidated and reviewed by two other researchers. By reaching a consensus over the review and comparison we are confident that we have significantly increased the objectivity and rigor in this type of research.

Many of the available process modeling techniques have been designed for distinct purposes. 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 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).

Because the prior analyses were independently conducted by different research groups, and because representational analyses may refer to varied research purposes [Rosemann and Green, 2000], effort was put into making the individual analyses comparable. We argue that the reduction of the BWW model constructs to the largest common set of used constructs enables the comparison of mapping results. We did neither question nor review the mapping results as proposed by the different research groups. Hence, our study consolidates previous analyses instead of revising or extending them. Nevertheless, to enable the comparison of previous studies, we had to generalize some constructs of the BWW model.

In particular, as some analyses did not entirely differentiate between the *property* sub-types as defined in [Wand and Weber, 1993, 1995, Weber, 1997], all sub-types were generalized to the super-type *property*. Therefore, if a mapping was found for a sub-type of *property*, *e.g.*, *emergent* or *mutual binding 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* (which form parts of the *lawful transformation construct*), we generalized mappings of these to a mapping of the *lawful transformation construct*. As a last item of consideration, the construct *process* [Green and Rosemann, 2000] was not specified in the BWW representation model as defined in [Wand and Weber, 1993, 1995, Weber, 1997]; therefore we did not consider it in our study.

An additional point of concern in the consolidation and comparison was related to the shortcoming of analyses focusing on both ontological completeness and clarity. As for the investigation of the evolution of ontological clarity of process modeling techniques, in particular construct excess, redundancy and overload, we had to reduce the size of our sample. This situation is due to a lack of consideration of aspects of ontological clarity in the study of ANSI Flowcharts, ISO/TC87, MERISE, DFD and IDEF3, as the evaluation performed by Keen and Lakos [1996] was restricted to ontological completeness only.

In the following section we will structure our line of investigation in accordance with the four types of representational deficiencies of modeling techniques, *viz*., construct deficit, redundancy, excess, and overload.

CONSTRUCT DEFICIT IN PROCESS MODELING TECHNIQUES

Construct deficit of a particular process modeling technique occurs in situations in which no language construct can be identified that maps to a particular BWW construct. This situation can be interpreted as the lack of means for users to capture and describe certain real-world phenomena. The focus of this aspect is to identify the degree of deficit (DoD), being the extent to which process modeling techniques are unable to provide complete descriptions of a real-world domain; hence, DoD is an inverse measurement of the degree of completeness of a modeling technique. DoD can be measured relatively as the number of BWW constructs found not to have a mapping to language constructs (#C) divided by the total number of constructs defined in the BWW representation model (#M).[1]

The results of our comparison are illustrated in Table 1. Each tick indicates that the specified BWW construct can be represented by the analyzed technique.

[1] This metric is based on the assumption that each construct in the BWW model is equally relevant, *viz*., each construct has the same weight. It has been argued that this may in fact not always hold true in modelling practice [Rosemann *et al.*, 2004]; however, our metric in principle also allows for the derivation of weighted measurements.

Table 1. Comparison of construct deficit of process modeling techniques												
BWW Construct / Language / Version / Year	Petri net	ANSI Flow-charts	DFD	ISO TC87	Merise	EPC	IDEF3	ebXML	BPML	WSCI	BPEL4WS	BPMN
Version								1.01	1.0	1.0	1.1	1.0
Year	1962	1970	1979	1982	1992	1992	1995	2001	2002	2002	2003	2004
Things including Properties and Types of Things												
THING	✓			✓	✓		✓					✓
CLASS	✓							✓	✓	✓	✓	✓
KIND												✓
PROPERTY			✓			✓	✓	✓	✓	✓	✓	✓
States assumed by Things												
STATE	✓					✓	✓	✓	✓	✓	✓	
CONCEIVABLE STATE SPACE								✓				
STATE LAW	✓			✓	✓	✓		✓				
LAWFUL STATE SPACE	✓							✓				
STABLE STATE						✓		✓				
UNSTABLE STATE	✓							✓				
HISTORY								✓				
Events and Transformations occurring on Things												
EVENT	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
CONCEIVABLE EVENT SPACE								✓				
LAWFUL EVENT SPACE								✓				
EXTERNAL EVENT				✓	✓	✓		✓	✓	✓	✓	✓
INTERNAL EVENT	✓			✓	✓	✓		✓	✓	✓	✓	✓
WELL-DEFINED EVENT	✓					✓		✓	✓	✓	✓	✓
POORLY DEFINED EVENT								✓	✓	✓	✓	✓
TRANSFORMATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LAWFUL TRANSFORMATION	✓			✓	✓	✓		✓	✓	✓	✓	✓
ACTS ON	✓									✓	✓	✓
COUPLING		✓	✓		✓		✓	✓		✓	✓	✓
Systems structured around Things												
SYSTEM			✓		✓		✓	✓		✓	✓	✓
SYSTEM COMPOSITION			✓		✓		✓			✓	✓	✓
SYSTEM ENVIRONMENT			✓									✓
SYSTEM STRUCTURE					✓		✓			✓	✓	
SUBSYSTEM								✓				✓
SYSTEM DECOMPOSITION			✓				✓					✓
LEVEL STRUCTURE			✓			✓	✓					✓
Degree of Deficit	58.6 %	93.1 %	72.4 %	75.9 %	62.1 %	62.1 %	62.1 %	27.6 %	65.5 %	48.3 %	48.3 %	34.5 %
Degree of Completeness	41.4 %	06.9 %	27.6 %	24.1 %	37.9 %	37.9 %	37.9 %	72.4 %	34.5 %	51.7 %	51.7 %	65.5 %

Drawing on the clusters identified by Rosemann and Green [2002], Table 1 presents interesting patterns in the representation capabilities of the process modeling techniques under observation.

In terms of the cluster *things including properties and types of things*, Table 1 reveals that only BPMN is able to cover all aspects of *things*. In this aspect, BPMN appears to denote a considerable improvement compared to other techniques. For example, the popular and widely used EPC performs poorly in terms of this cluster, indicated by a relatively high degree of deficit (75%). Also, the poor performances of Flowcharts (100%) and DFD (75%) are notable given their relatively high level of adoption in modeling practice [Davies *et al.*, 2006]. Closer inspection of Table 1 shows that while earlier process modeling techniques provided a construct for representing a specific thing, more recent standards have representation capabilities for classes of things rather than for an individual 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. These findings support earlier studies that reported that, for instance, DFD diagrams are often complemented with Entity-Relationship Diagrams [Chen, 1976] that specify the nature and relationships between the modeled real-world things [Wand and Weber, 1993].

From the perspective of the cluster *states assumed by things*, throughout the BPM domain, a lack of support for business rule definitions can be observed. For empirical support for this proposition refer, for example, to [Davies *et al.*, 2004, Green and Rosemann, 2001, Recker *et al.*, 2006]. In particular, the lack of support for the representation of conceivable and lawful state spaces indicates that state and transformation modeling will be unclear to the modeler when trying to determine which set of states can potentially occur in a system and which states are possible but should not be allowed. A closer look at Table 1 also reveals that most techniques have a very high degree of deficit in the cluster of states assumed by things (see, for instance, Flowcharts, IDEF3, and BPMN), except for ebXML (0% in this cluster) and Petri nets (48% in this cluster). This situation suggests that the modeling of business rules is heavily dependent on rigorous state and state law specification. The rigorous mathematical specification of Petri nets and the semi-formal specification of ebXML BPSS, by means of UML diagrams, appear to be advantageous in this aspect.

As would have been expected in the process modeling domain, Table 1 indicates that most of the investigated techniques perform reasonably well in the cluster *events and transformations occurring on things*. This finding supports the argumentation that things, events and transformations are core concepts in process modeling [Soffer and Wand, 2005]. An interesting observation can be made with respect to the degrees of deficit of Flowcharts (82%), DFD (82%) and IDEF3 (73%). We speculate that the relatively high degrees of deficit can be explained by the fact that these grammars were originally developed with the intention of modeling information flows rather than process or communication flows (see [Danesh and Kock, 2005]) and hence did not put emphasis on the consequences that events may have on the transformation of the modeled things. Also, note again that ebXML BPSS performs best from the viewpoint of construct deficit (9%). Moreover, it denotes the single technique capable of depicting both conceivable and lawful event spaces. This situation may partly be caused by its prevalent focus on direct process execution. Its rigorous specification allows for the well-defined differentiation between potential and/or inevitable events and also states.

In the cluster *systems structured around things*, in general, there appears to be inconsistent support. From the list of seven BWW constructs in this cluster, five have been found to be represented in fewer than 34% of the considered modeling techniques. Thus, appropriate structuring and differentiation of modeled things or entities, such as business partners, is not well supported. We find this fact 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. These three techniques have in common dedicated language constructs for decomposing process models into interlinked hierarchical subsets (for example, the sub-process construct in BPMN).

CONSTRUCT REDUNDANCY IN PROCESS MODELING TECHNIQUES

Construct redundancy occurs in situations in which a process modeling technique has more than one language construct mapping to the same BWW construct. This situation potentially causes confusion in the usage of the respective modeling technique. In light of the underlying theory semantically equal language constructs that seem to be indistinguishable in their real-world meaning and thus denote an unnecessary duplication, lead to potential confusion in the interpretation of the resulting model. The focus of this aspect is to identify the degree of redundancy (DoR) of a process modeling technique, which in turn serves as an indication of a technique’s capabilities to provide clear descriptions of the modeled domain [Weber, 1997]. DoR can be measured relatively as the number of language constructs found to have a mapping to the same BWW construct (#R) divided by the total number of constructs in the modeling technique (#T). For example, Table 2 reveals that ebXML

BPSS contains three language constructs for representing the BWW construct *event*. Hence, ebXML contains two potentially redundant constructs out of a total of 51 language constructs.

In order to comparatively assess the occurrences of construct redundancy in leading process modeling techniques, it is necessary to elaborate on the following situations.

Due to the generalization of all property-related sub-types to the super-type *property*, we cannot make predictions as to construct redundancy in terms of properties. Hence, in Table 2, an "x" indicates that the respective process modeling technique provides a differentiated set of constructs to depict certain properties. For instance, EPC allows for the definition of attribute types that group sets of free attributes in accordance to any given purpose.

Also note that *events* and *states* have further sub-types in the BWW model, namely *unstable/stable state*, *internal/external* and *well-defined/poorly-defined event*. If a technique contains two language constructs that provide representations for *state* (or *event*), each of which disjointly represents one of its BWW sub-types (for example, one representation for *stable state*, one for *unstable state*), these constructs are not deemed redundant.

The results of our comparison are illustrated in Table 2. For each BWW construct, we indicate the number of process modeling technique constructs that have been found to represent the BWW construct. Note again the reduced set of process modeling techniques that we were able to consider.

Table 2. Comparison of construct redundancy of process modeling techniques							
Language	Petri net	EPC	ebXML	BPML	WSCI	BPEL4WS	BPMN
Version			1.01	1.0	1.0	1.1	1.0
Year	1962	1992	2001	2002	2002	2003	2004
BWW Construct							
Things including Properties and Types of Things							
THING	**1**						**2**
CLASS	**1**		**3**	**1**	**1**	**1**	**2**
KIND							**1**
PROPERTY		**x**	**x**	**x**	**x**	**x**	**x**
States assumed by Things							
STATE	**3**	**1**	**5**	**1**	**1**	**1**	
CONCEIVABLE STATE SPACE			**1**				
STATE LAW	**1**	**1**	**1**				
LAWFUL STATE SPACE	**1**		**1**				
STABLE STATE		**1**	**4**				
UNSTABLE STATE	**3**		**1**				
HISTORY			**1**				
Events and Transformations occurring on Things							
EVENT	**1**	**1**	**3**	**3**	**6**	**4**	**9**
CONCEIVABLE EVENT SPACE			**1**				
LAWFUL EVENT SPACE			**3**				
EXTERNAL EVENT		**1**	**3**	**2**	**3**	**1**	**8**
INTERNAL EVENT	**1**	**1**	**4**	**1**	**3**	**3**	**8**
WELL-DEFINED EVENT	**1**	**1**	**1**	**2**	**2**	**1**	**2**
POORLY DEFINED EVENT			**2**	**1**	**1**	**2**	**7**
TRANSFORMATION	**1**	**1**	**1**	**10**	**8**	**11**	**6**
LAWFUL TRANSFORMATION	**1**	**1**	**1**	**4**	**4**	**3**	**7**
ACTS ON	**1**					**1**	**1**
COUPLING			**2**		**1**	**1**	**1**
Systems structured around Things							
SYSTEM			**2**		**1**	**1**	**2**
SYSTEM COMPOSITION					**1**	**1**	**2**
SYSTEM ENVIRONMENT							**2**
SYSTEM STRUCTURE					**1**	**1**	
SUBSYSTEM			**1**				**2**
SYSTEM DECOMPOSITION							**2**
LEVEL STRUCTURE		**1**					**2**
Degree of Redundancy	28.6 %	0.0 %	15.7 %	30.4 %	30.6 %	31.9 %	51.3 %

In terms of *things*, their types and properties, in general it appears that the relatively high degree of deficit in this cluster comes with a relatively low degree of redundancy. However, we can comment on two points. First, although BPMN provides full coverage for this cluster, this coverage comes at the cost of a high degree of redundancy. In particular, confusion arises as to the differentiation of the Lane construct from other representations for things and classes of things, namely Pool and Data Object [Recker *et al.*, 2006]. Second, ebXML BPSS provides several constructs for representing classes of things, which may cause confusion when some instances of a class participate in a relationship and other instances do not. For example, it may be unclear under what circumstances an instance of a DocumentEnvelope is used by RequestingBusinessActivity [Green *et al.*, 2005].

In terms of states assumed by things, the coverage by process modeling techniques is limited, which in turn is associated with a relatively low degree of redundancy. Two points can be made. First, Petri nets appear to have redundant constructs for modeling the states of *things* in light of the BWW representation model, in particular, unstable states. Specifically, our own analysis of construct redundancy in Petri nets revealed that Petri nets have three different concepts for representing the (unstable) state of a thing, being Place, Initial Marking and Token. From a representational perspective this situation induces ambiguity in the use of the technique. However, we note that this proposition should be subject to further discussion (which is outside the scope of this paper but a noted future research direction), as the necessity of the mentioned constructs for the formal verification and analysis of workflow specification languages cannot be neglected [Kiepuszewski *et al.*, 2003]. Second, ebXML BPSS appears to be subject to frequent redundancy with respect to the representation of stable states. Its constructs Start, Fork, Join and Success all appear to be redundant in their representational capability and thus potentially cause confusion in the use of this technique. It may hence be worthwhile to consider reducing the range of constructs available to a more limited set that avoids this redundancy.

Constructs for representing events and transformations occurring on things are found to have a higher level of redundancy. In fact, 71% of the techniques under investigation provide more than one construct for representing an event or internal event (83% in terms of external events). Similarly, the *lawful transformation* construct is found to be mapped to more than one language construct for 57% of the considered techniques, sometimes even to ten or more constructs, as it is the case of, for instance, BPML and BPEL4WS. This may have two underlying causes. Perhaps the underlying theory, the BWW model, lacks specificity in this cluster – see also the discussions in [Rosemann *et al.*, 2004, Rosemann *et al.*, 2006]. In particular, it appears that the BWW model may have too high a level of granularity. This situation may imply that, just as properties in the BWW representation model are specialized, perhaps events and transformations should also be further differentiated. A second interpretation is that process modeling techniques tend to provide a surplus of constructs for the representation of these domain phenomena without any representational need for such differentiation, as advocated by the theory. Our own recent empirical findings, however, indicate that the former interpretation is more likely, as only a limited number of interviewed process modelers classified the amount of language constructs for event and transformation representation as potentially confusing [Recker *et al.*, 2006]. In a related proposition, a closer inspection of Table 2 reveals the particularly high degree of redundancy of BPMN in this cluster (71%) as compared to alternative techniques, for example, EPCs (0% in this cluster).

BPMN also appears to be the single technique subject to frequent redundancy in the cluster of systems structured around things. Both the Lane and Pool constructs allow the depiction of various aspects of systems. This result implies that the differentiation of these constructs in the specification needs to be improved to allow for a better understanding in which context each of the specialized constructs is more appropriate.

CONSTRUCT EXCESS IN PROCESS MODELING TECHNIQUES

Construct excess occurs in situations in which a process modeling technique provides language constructs that do not map to any BWW construct. This situation can be interpreted as the provision of constructs that appear to have no real-world meaning as per the BWW representation model. Accordingly, users will get confused when using these constructs and, thus, will need mechanisms for further clarification. The focus of this aspect is to identify the degree of excess (DoE) of a process modeling technique, which in turn serves as another indication for its capabilities to provide clear descriptions of the modeled domain [Weber, 1997]. DoE can be measured relatively as the number of language constructs found not to have a mapping to any BWW construct (#E) divided by the total number of constructs in the modeling technique (#T). For example, BPMN contains a language construct named 'text annotation', which can be used to attach to a process diagram textual descriptions for which no graphical symbol is provided. Such a situation would indicate that BPMN users have to employ textual means for capturing real-world phenomena in the problem domain due to a lack of graphical means for doing so. The textual annotation is, as per BWW model, proposed as excess since its meaning is not prescriptively specified and thereby potentially subject to misuse and misinterpretation.

The results of our comparison of the occurrences of construct excess in leading process modeling techniques are illustrated in Table 3. It shows each process modeling technique construct that has been found not to have a mapping to any BWW construct.

Table 3. Comparison of construct excess of process modeling techniques

Language / Version / Year	Petri net	EPC	ebXML	BPML	WSCI	BPEL4WS	BPMN
			1.01	1.0	1.0	1.1	1.0
	1962	1992	2001	2002	2002	2003	2004
Proposed excess constructs		AND connector	Performs	Empty	Correlate	Empty	Link
		OR connector	Business Activity	Delay	Selector	Message Properties	Off-page connector
		XOR connector	Business State	Context	All	Message Definitions	Association Flow
			Business Action	Sequence	Sequence	Sequence	Text Annotation
			Document Security	All	Join	Flow	Group
			Completion State	Choice	For Each	Scope	Activity Looping
			Enumeration Status	For Each	Empty		Multiple Instances
				Synch	Spawn		Normal Flow
				Activity Instance State	Context		Event
				Activity Type			Gateway
				Spawn			Data-based XOR
				Parameters			Event-based XOR
				Identity			Inclusive OR
							Complex
							Parallel
Degree of Excess	0.0 %	42.9 %	13.7 %	28.3 %	18.4 %	12.8 %	38.5 %

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 repeatedly proposed as construct excess since they do not map to any construct of the BWW model. Again, this poses the question whether the underlying theory is of appropriate specificity to the domain of process modeling or whether such mechanisms for the description of control flow convergence and divergence really contribute to the description of a real-world domain. Based on the understanding that control flow mechanisms essentially support the notion of being “in between” states or activities [Kiepuszewski *et al.*, 2003] one may argue that this does not denote a representation facet of a real-world domain but rather the depiction of the decisions made “in-between” within such domains.

It further appears that some modeling techniques, such as BPMN, provide language constructs that, in their essence, may be useful for the act of modeling but not for capturing domain semantics or real-world phenomena. Candidates for these scenarios include for instance Off-page Connector, Group, and Text Annotation, which define means to link models, group model elements, or attach additional descriptions to models. Our research findings suggest the externalization of such modeling means from the respective technique into modeling tools. Thereby, the act of modeling can be supported through constructs such as text annotation, grouping elements or others in a technique-independent fashion, while the technique itself merely contains domain representation constructs. This would lead to reduced levels of complexity in the usage of the technique. Again, we see an interesting research challenge stemming from this observation.

Other candidates that are proposed as excess, such as DocumentSecurity and EnumerationStatus (ebXML BPSS), Parameters and Activity Instance State (BPML), Message Properties and Message Definitions (BPEL4WS), Spawn (WSCI), or Multiple Instances (BPMN), all have in common that they capture certain aspects of process implementation and execution but not domain phenomena. Again, taking the viewpoint of the BWW model, for the purpose of describing semantics of the modeled domain, these constructs may be considered unnecessary. This situation poses major implications to process modeling practice as our findings can be used to devise training courses or modeling methodologies for the techniques with respect to various roles (*e.g.*, business analyst versus technical analyst) or purposes (*e.g.*, documenting business requirements versus specifying system requirements).

CONSTRUCT OVERLOAD IN PROCESS MODELING TECHNIQUES

Construct overload occurs in situations in which a process modeling technique provides language constructs that map to more than one BWW construct. This situation can be interpreted as causing confusion in the usage of the respective modeling technique as it provides language constructs that appear to have multiple real-world meanings and thus can be used to describe various real-world phenomena. These cases are undesirable as they require users to bring to bear knowledge external to the model in order to understand the capacity in which such a construct is used in a particular scenario. The focus of this aspect is to identify the degree of overload

(DoO) of a process modeling technique, which in turn serves as a further indication for its capabilities to provide clear descriptions of the modeled domain [Weber, 1997]. DoO can be measured relatively as the number of language constructs found to have a mapping to more than one BWW construct (#O) divided by the total number of constructs in the modeling technique (#T). For example, the Petri nets technique has a place construct that can be used to represent a thing, class, or state. Hence, with respect to the BWW representation model, Petri nets contain at least one theoretically overloaded construct out of a total of seven language constructs.

Again, as with the discussion relating to redundancy of constructs, we consider here the same situations of *events* and *states* being able to also be represented as mutually exclusive sub-types of *events* and *states* without being considered as overloaded.

The results of our comparison of the occurrences of construct overload in leading process modeling techniques are illustrated in Table 4. The table shows each process modeling technique language construct that has been found to have a mapping to more than one BWW construct.

Table 4. Comparison of construct overload of process modeling techniques

Language	Petri net	EPC	ebXML	BPML	WSCI	BPEL4WS	BPMN
Version			1.01	1.0	1.0	1.1	1.0
Year	1962	1992	2001	2002	2002	2003	2004
Proposed overloaded constructs	Place	Function Type	Binary Collaboration		Connect	Partners	Lane
	Place Capacity	Event Type			Model		Pool
	Transition						Message Flow
							Start Event
							Intermediate Event
							End Event
							Message
							Error
							Cancel
							Compensation
Degree of Overload	42.9 %	28.6 %	2.0 %	0.0 %	4.1 %	2.1 %	25.6 %

It appears that process modeling techniques are quite diverse in their levels of construct overload. In an earlier study [Rosemann *et al.*, 2006] we mentioned that the same deliberately flexible specification that affords Petri nets a higher ontological completeness, also results in extensive overload of constructs such as Place, Place Capacity and Transition. We also mentioned earlier the design for flexibility in terms of the Lane and, to a lesser extent, Pool constructs in BPMN. Hence, the trade-off between flexible usage (and, therefore, multiple meanings versus ease of understanding of the model) and specification rigor (and, therefore, limited usability versus intuitiveness) of language constructs appears to be a recurring pattern in the development of modeling techniques, which designers have to face. The BWW model facilitates the generation of related propositions in that it advocates the clarity of a specification. The perceived impact of clear versus flexible specifications on modeling practice, however, may be subject to individual preferences and purposes, as shown in [Recker *et al.*, 2006].

Two other observations can be made from Table 4. First, both Petri nets and EPCs have a relatively high degree of overload (43% and 29%, respectively), which may be explained by the restricted number of language constructs overall (seven). From the viewpoint of the BWW model, such flexibility is only seemingly an advantage and can result in a model that is not easily interpreted by a user. Empirical findings from other related analyses support this view, for example, empirical findings in the case of BPMN [Recker *et al.*, 2006]. Second, BPML appears to be the single technique under investigation not exhibiting construct overload. Therefore, we can assume that modelers using this notation are not required to bring in extra-model knowledge to the modeling task and we further assume that the understandability of the resulting BPML models is relatively high. These propositions are subject to future empirical validation.

CONSOLIDATION OF RESULTS

Having extracted the similarities and differences in terms of the four representational deficiencies between the process modeling techniques under observation, we seek to obtain a consolidated picture of the overall representational capability of these techniques. In particular we are interested in identifying the relationship between the ontological completeness of the techniques (measured by the degree of deficit) and their ontological clarity. This allows us to identify the "costs" (in terms of the clarity of the technique specification) of obtaining a certain scope of coverage in a technique.

Representation theory [Weber, 1997, p. 85] advocates that process modeling techniques should be complete in their representation of real-world phenomena, *viz.*, they should have as low a degree of deficit as possible. The theory also states that process modeling techniques should be clear in their capabilities to facilitate representations of real-world domains, *viz.*, they should have as low degrees of redundancy, excess and overload as possible. We were interested in finding out to what extent the considered process modeling techniques adhere to the overall theory principles of providing complete as well as clear representations of real-world domains. Figure 2 presents a radar chart that gives a consolidated overview of the four degrees of deficiencies across the considered process modeling techniques. Note again the reduced set of process modeling techniques we were able to consider.

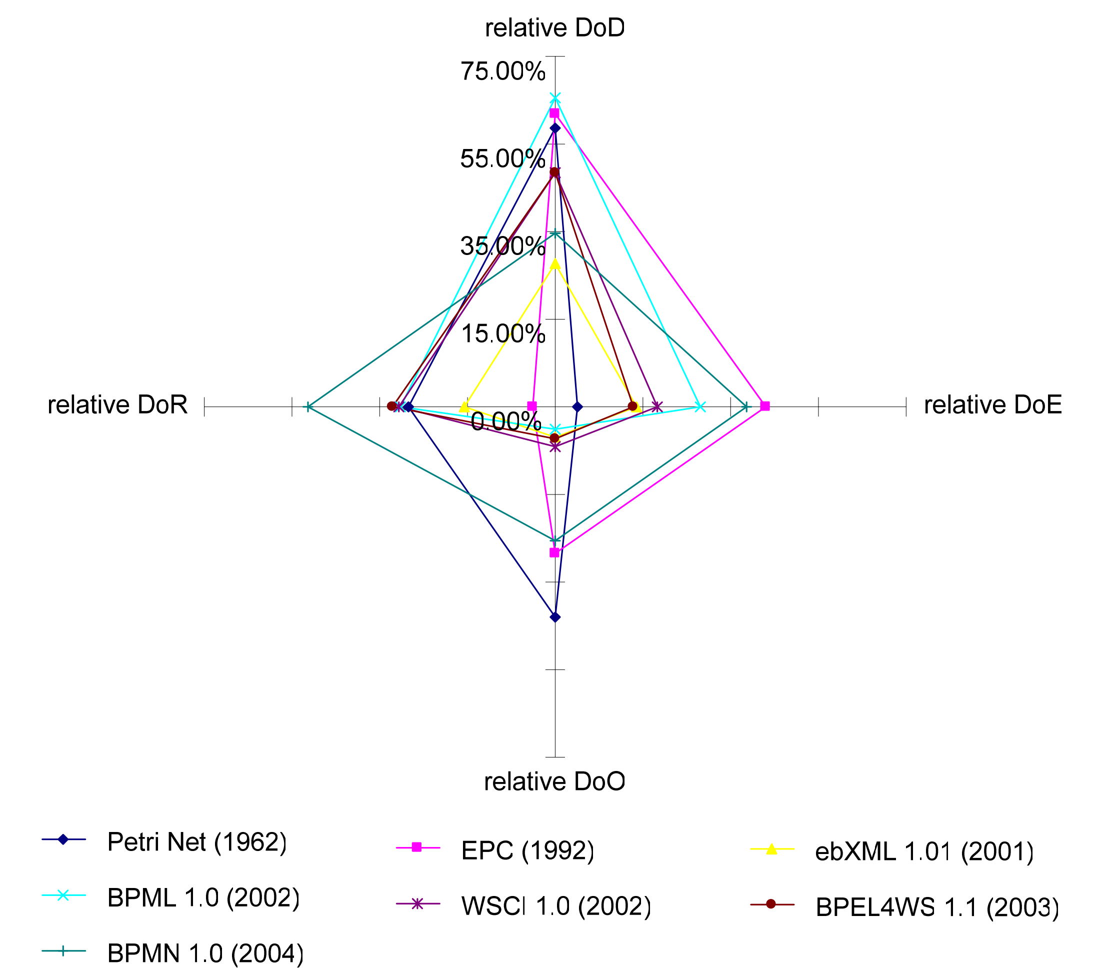

Figure 2. Overview of the degrees of completeness, redundancy, excess and overload of process modeling techniques

From Figure 2 some interesting conclusions can be drawn in regards to the representational capabilities of process modeling techniques. Clearly, the capability of ebXML is closest to the general principles of representation theory, as its relatively low degree of deficit (28%) is complemented by low degrees of redundancy (16%), excess (14%) and overload (2%). It can thus be assumed that the use of ebXML not only enables modelers to create reasonably complete descriptions of real-world domains but also relative clear descriptions that bear little complexity and can unambiguously be interpreted. BPMN, on the other hand, while obtaining a considerably low DoD (34%), achieves high degrees of deficiency across all clarity aspects (DoR: 51%; DoE: 38%; DoO: 26%). The use of BPMN can thus be expected to lead to quite complete but also unclear and potentially ambiguous representations of real-world domains.

Two interesting patterns can be observed from Figure 2. First, some techniques, such as Petri nets, achieve low degrees of redundancy and excess with high degrees of overload. In terms of Figure 2 the corresponding graph thus looks like a straight vertical line between the dimensions of deficit and overload. The scope of coverage of these techniques is thus obtained through a rather restricted set of language constructs, which in turn are subject to overload. From this observation a technique design principle emerges that advocates a process modeling technique specification with a minimal set of language constructs that is at the same time very flexible in meaning and purpose. The use of such a technique would thus not bear complexity due to a surplus of equivalent or excessive language constructs. However, the resulting models may still be prone to understandability concerns as the used language constructs have, *prima facie*, multiple meanings in the model. As opposed to this, a second set of techniques, such as BPML or WSCI, achieve a low degree of overload with high degrees of redundancy and excess. Their graphs in Figure 2 thus correspond more to a triangle between the dimensions of deficit, redundancy and excess. The observable underlying technique design principle is coined by a technique specification that offers an extensive set of language constructs for modeling that, while being clear in specification (indicated by a low degree of overload), are potentially redundant and/or excessive. In consequence, such techniques achieve a certain scope of coverage through a multitude of constructs, which in turn, *prima facie*, offer too many choices for representing the real-world phenomena the user seeks to describe. Such design principle seems to be based on technique extension rather than revision and clarification.

In conclusion, the consolidated overview of the representational capabilities of process modeling techniques in Figure 2 can be used to guide relevant stakeholders in the selection of an appropriate process modeling technique. Based on preferences that stem from factors such as modeling role or modeling purpose a technique that is potentially redundant in its use may or may not be favorable in contrast to a technique that is neither excessive nor redundant but overloaded. While the overall objective of providing complete representations of real-world domains can be regarded as given, certain trade-offs can be made with respect to the "costs of clarity" with which the desired scope of coverage can be achieved. The investigation of such preferences and trade-offs, however, is outside the scope of this paper and is designated as future work.

IV. GAUGING THE MATURITY OF PROCESS MODELING TECHNIQUES

So far we have put the individual analyses of process modeling techniques into a comparative context in order to extract similarities and differences in their representational capabilities. Next, we seek to assess the evolution of process modeling technique development over time. As the process modeling discipline evolved only recently as a dedicated research field, we were curious whether this emerging research field would follow the overall guideline of establishing, and building on, a cumulative tradition [Keen, 1980, Weber, 1997]. Our motivation then was to study the development of the capabilities of process modeling techniques over time, using the above defined measurements of ontological completeness (DoD) and clarity (DoR, DoE, DoO) to gauge the level of maturity increase over time. Following the propositions of representation theory [Weber, 1997] we understand a *mature* process modeling technique as one that facilitates a complete description of a real-world domain while being clear in its usage in the sense that the use of the technique does not cause confusion to the modeler due to redundant, excessive or overloaded language constructs. Increased maturity then is an increase of the degree of completeness of a technique combined with a decrease of the degrees of redundancy, excess and overload.

In the following two sections we will investigate the evolution of the degrees of completeness and clarity of process modeling techniques over time to then be able to draw conclusions about the state of maturity of these techniques.

DEVELOPMENT OF ONTOLOGICAL COMPLETENESS OF PROCESS MODELING TECHNIQUES

A longitudinal study of the degree of completeness of the analyzed techniques indicates an increase in the coverage of the BWW constructs. Figure 2 visualizes this trend over time, as measured by the inverse extent of construct deficit of each analyzed technique, listed in chronological order (the line connecting the data points in Figure 3 is used for illustration purposes only).

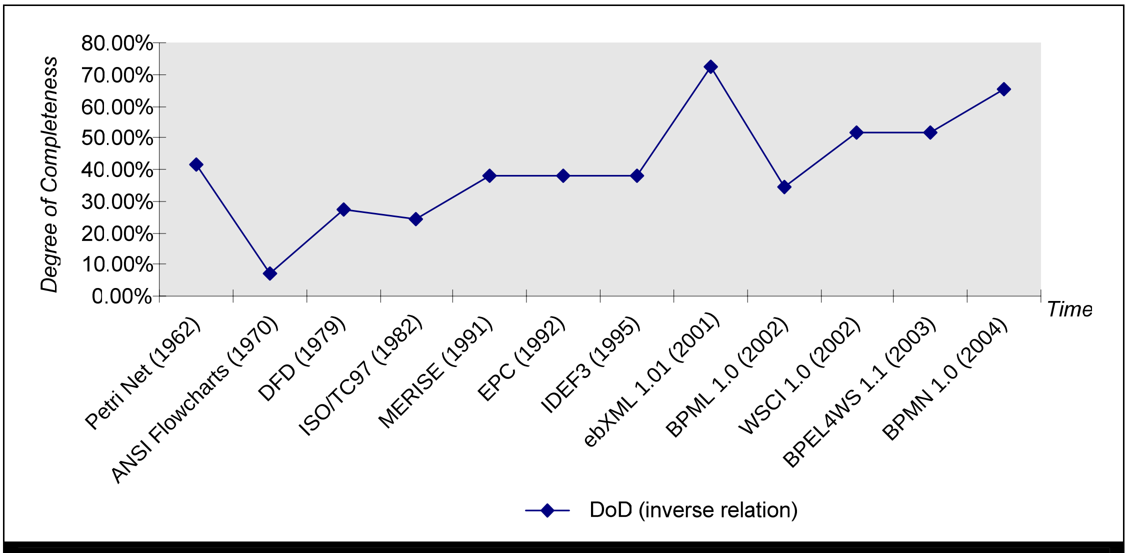

Figure 3. Development of the degree of completeness of process modeling techniques over time

From this visualization we can observe that while the original Petri nets specification did not achieve a relative high degree of completeness (41%) with respect to the BWW representation model, it still outperformed more recent techniques such as DFD or IDEF3 diagrams in this respect. In general, over time, the scope of coverage (measured by an inverse DoD) continuously increased. A noticeable spike in Figure 3 depicts the high level of maturity (in terms of ontological completeness) of the ebXML standard (76%), which appears to be the most complete process specification to date. It is interesting to note that ebXML is specified in UML [OASIS, 2001], with a semi-formal construct definition and description. This situation contrasts to the latest techniques. BPEL4WS, WSCI, and BPMN have textual specifications supplemented by diagrams of examples. As such, the ebXML specification is less subjective in its possible interpretations [Davies *et al.*, 2005, Rosemann *et al.*, 2004]. It is also worthwhile to note that the most recent standard, BPMN, performs very well from the viewpoint of ontological completeness (66%). This higher degree of completeness can perhaps partly be explained by the fact that previous approaches, including EPC and Petri nets, influenced the development of the BPMN specification [BPMI.org and OMG, 2006b].

It appears in general that techniques that focus on describing process flow from a business perspective (for instance DFD and IDEF3) are less complete than those that have to cater for more syntactical rigor due to their focus on executability (such as BPEL4WS or ebXML BPSS for example). Overall, Figure 3 suggests that there is an upward trend in the representational ability of the analyzed techniques in terms of their capabilities to provide complete domain descriptions. This finding further suggests that new techniques are in fact building on the capabilities of the previous techniques. BPMN specifically has been designed by its authors based on the analysis of previous techniques and their advantages [BPMI.org and OMG, 2006b], in particular, the developers sought to incorporate into the development of BPMN some of the successful design aspects of techniques such as IDEF3, EPC and others.

DEVELOPMENT OF ONTOLOGICAL CLARITY OF PROCESS MODELING TECHNIQUES

A longitudinal study of aspects of ontological clarity of the analyzed techniques also leads to several interesting findings. Figure 4 visualizes the trends over time as measured by the degrees of redundancy, excess and overload of each analyzed technique.

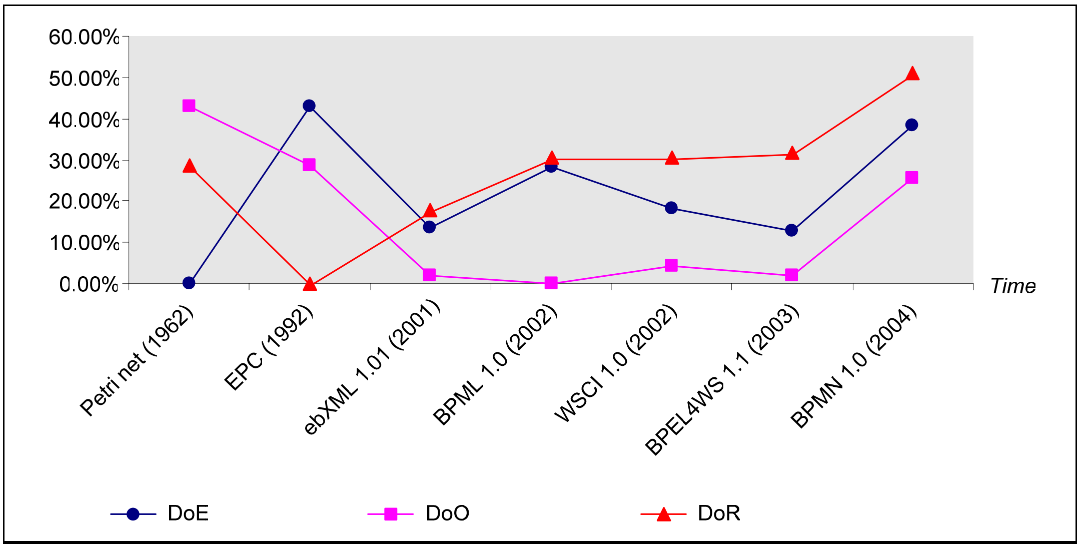

Figure 4. Development of the degrees of excess, overload and redundancy of process modeling techniques over time

In terms of DoE, the longitudinal study implies a slight upward trend. We observe that process modeling techniques that focus more on a business analyst perspective (such as EPCs, BPML or BPMN), *i.e.*, that are predominantly used for capturing business requirements, have more excess constructs (average DoE 37%) than techniques that have a focus on a technical analyst perspective and can be used for process automation and execution (such as Petri nets, ebXML, WSCI and BPEL4WS (average DoE 11%)). We see a reason for this in the more formal and rigorous specification of the expressive power of techniques such as Petri nets, ebXML, WSCI and BPEL4WS.

Figure 4 suggests an incline in the DoR of leading process modeling techniques since Petri nets. Starting with EPCs (0%), the redundancy of techniques continuously increased, reaching its peak in the most recent notation, BPMN (51%). For example, since the development of ebXML in 2001, process modeling techniques have started to develop differentiated sets of constructs for representing events and transformations (averages of 5.0 and 7.2 language constructs, respectively). Representation theory suggests in this regard that, over time, techniques have been developed that provide more and more constructs without having a representational need for doing so. In effect, the complexity of these techniques has continuously increased and the usage of these techniques is becoming more and more confusing. A resulting proposition would be to streamline process modeling techniques rather than extending them with even more constructs that essentially capture the same aspects.

However, we are also aware that requirements for process modeling have changed over time, with current generations, for instance, focusing on various aspects of advanced process orchestration and choreography [Si *et al.*, 2005]. The related question is whether or not the BWW representation model sufficiently reflects such specialized requirements.

Another interesting observation emerges from the trend of DoO. Not taking BPMN into account, Figure 4 illustrates a downward and then stable trend in terms of overloaded constructs in process modeling techniques, not exceeding 5% since 2001. We discussed before that a potential reason for the overload of Petri nets and EPCs lies within the relatively low extent of language constructs overall. As for the rest, we see one potential explanation for the low DoO to be the utilization of semi-formal or formal specification methods for technique development, for instance by means of meta-models, *e.g.*, ebXML [OASIS, 2001], or XML schemas, *e.g.*, BPML [Arkin, 2002]. The most recent technique, BPMN, however, contrasts with this overall trend. Not only does it provide by far the highest number of overloaded constructs (ten – the sum of overloaded constructs from the other considered techniques), some of the constructs were found to be excessively overloaded, for instance Lane and Pool, which in turn have been causing confusion in the use of this technique, see [Recker *et al.*, 2006]. We see two causes for this situation in particular. First, as opposed to the other techniques, BPMN is currently specified in a textual manner and thereby potentially subject to misinterpretation and misuse. However, plans to develop a semi-formal specification of BPMN using UML are underway [BPMI.org and OMG, 2006a], which in turn might clarify the specification of some constructs. Another reason is the deliberately flexible manner in which some constructs, such as Lane and Pool specifically, are specified. Again we raise the question whether process modeling obtains higher benefits from a flexible specification and usage of the language constructs provided than from a distinct specification and usage, as advocated by representation theory.

V. CONCLUSIONS

CONTRIBUTIONS & IMPLICATIONS

This paper presents a comprehensive comparative and longitudinal study of previous representational analyses of process modeling techniques, also including the outcomes of our representational analyses of Petri nets and BPMN. The findings show 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*). The findings also allow for conclusions to be drawn as to the signs of maturity, as measured by the degrees of completeness, excess, overload and redundancy of process modeling techniques over time. Furthermore, our findings serve as input to the question of the applicability of the BWW representation model as a benchmark for analyses of process modeling techniques in that we were able to find areas of the theory where further work is needed, *e.g.*, in the area of event and transformation specializations. We have not considered the specialization of these BWW model constructs in this paper, however, we perceive the findings discussed here as highly relevant to such a discussion [Rosemann *et al.*, 2006].

The outcomes of this study are of particular interest to both developers and users of process modeling techniques. Developers should be motivated to examine previous representational analyses of existing process modeling techniques in order to build upon these grammars and mitigate any weaknesses in newly developed or extended techniques. The results will also motivate users to consider ontological completeness and ontological clarity as potential evaluation criteria for the selection of an appropriate modeling technique.

Our findings suggest that process modeling technique development in fact builds upon earlier versions in terms of widening the scope of coverage, measured by an increasing degree of completeness of process modeling techniques over time. As such, it can be expected that the effectiveness of process modeling techniques, and ultimately process modeling, has been increasing over time and will hopefully continue to do so in future generations of modeling techniques. Regarding the level of efficiency of process modeling, however, it appears that the discipline is heading in a direction where the widened scope of coverage also induces increased modeling complexity, measured by the degrees of overload, excess and redundancy of a technique. For example, the shown upward trend of construct redundancy is an indication of a design trend that is based on technique *extension* rather than *revision* or deletion of language constructs. A recent interview with the design team of the BPMN technique supports this proposition - the BPMN developers stated specifically that it is far more common to add constructs in technique revisions rather than to delete or replace them. Our findings can be used to guide modeling technique developers in their design efforts as they provide a theoretical base from which relevant design principles can be drawn that potentially counteract the indicated trend towards technique complexity whilst still enabling sufficient domain coverage.

LIMITATIONS

We identify four limitations in our research. Most notably, we based our study on previous representational analyses that have been conducted by different researchers. We are aware that the actual process of conducting a representational analysis is exposed to the impact of the subjective interpretations of the researcher [Rosemann *et al.*, 2004]. Therefore, we spent considerable effort on making the individual mapping results comparable. Second, we limited the considered representational analyses to studies based on the BWW representation model, which in turn constrains the generalization of the results and also the number of techniques we were able to consider. The BWW model provides a filtering lens that gives insights into some potential representational issues with a modeling technique. Yet, we are very much aware that ontological completeness and clarity are not the only relevant criteria for the evaluation of the capabilities of a modeling technique, and they need to be put into an overall context of other measures of quality of a modeling language. Third, we limited our research to ten previously analyzed process modeling techniques, adding to this the analysis of Petri nets and BPMN to have a more complete picture. While the selected sample can by no means claim to be complete, we believe it is representative of the most popular techniques. This finding can be supported by earlier surveys [Davies *et al.*, 2006]. The smaller scope also enables us to focus our work and to avoid the necessity to translate findings from different theoretical bases. Fourth, our research denotes a form of analytical study, which in turn can only result in theoretical propositions. The findings from our work call for appropriate empirical research strategies in order to confirm or falsify the implications drawn from our analysis. In this paper we have indicated some interesting propositions that require further operationalisation and testing. We would also like to invite other researchers to contribute in this field of study.

REFERENCES

American National Standards Institute (1970) *ANSI Standard flowchart symbols and their use in information processing (X3.5)*. New York, New York: American National Standards Institute.

Andrews, T., F. Curbera, H. Dholakia, Y. Goland et al. (2003) "Business Process Execution Language for Web Services. Version 1.1," BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG and Siebel Systems, http://xml.coverpages.org/BPELv11-May052003Final.pdf (February 10, 2006).

Arkin, A. (2002) "Business Process Modeling Language," BPMI.org, http://www.bpmi.org/ (January 16, 2006).

Arkin, A., S. Askary, S. Fordin, W. Jekeli et al. (2002) "Web Service Choreography Interface (WSCI) 1.0," BEA Systems, Intalio, SAP, Sun Microsystems, http://www.w3.org/TR/wsci/ (January 16, 2006).

Benbasat, I. and R. W. Zmud (1999) "Empirical Research in Information Systems. The Practice of Relevance," *MIS Quarterly* (23) 1, pp. 3-16.

BPMI.org and OMG (2006a) "BPMNModel UML Documentation," OMG, http://www.bpmn.org/Documents/BPMNMetaModel.zip (March 10, 2006).

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

Bubenko, J. A. (1986) "Information Systems Methodologies - A Research View," in T. W. Olle, H. G. Sol, and A. A. Verrijn-Stuart (Eds.) *Information Systems Design Methodologies: Improving the Practice*, Amsterdam, The Netherlands: North-Holland, pp. 289-318.

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

Chen, P. P.-S. (1976) "The Entity Relationship Model - Toward a Unified View of Data," *ACM Transactions on Database Systems* (1) 1, pp. 9-36.

Chisholm, R. M. (1996) *A Realistic Theory of Categories: An Essay on Ontology*. Cambridge, Massachusetts: Cambridge University Press.

Cocchiarella, N. B. (1995) "Knowledge Representation in Conceptual Realism," *International Journal of Human-Computer Studies* (43) 5-6, pp. 697-721.

Curtis, B., M. I. Kellner, and J. Over (1992) "Process Modeling," *Communications of the ACM* (35) 9, pp. 75-90.

Danesh, A. and N. Kock (2005) "An experimental study of process representation approaches and their impact on perceived modeling quality and redesign success," *Business Process Management Journal* (11) 6, pp. 724-735.

Davenport, T. H. (1993) *Process Innovation: Reengineering Work Through Information Technology*. Boston, MA: Harvard Business School Press.

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

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

Davies, I., M. Rosemann, and P. Green. (2004) “Exploring Proposed Ontological Issues of ARIS with Different Categories of Modellers.” *15th Australasian Conference on Information Systems, Hobart, Australia, 2004*.

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

Evermann, J. (2005) "Towards a Cognitive Foundation for Knowledge Representation," *Information Systems Journal* (15) 2, pp. 147-178.

Fettke, P. and P. Loos. (2003) “Ontological Evaluation of Reference Models using the Bunge-Wand-Weber Model.” *9th Americas Conference on Information Systems, Tampa, Florida, 2003*, pp. 2944-2955.

Fowler, M. (2004) *UML Distilled. A Brief Guide To The Standard Object Modelling Language*, 3rd edition. Boston, Massachusetts: Addison-Wesley Longman.

Gane, C. and T. Sarson (1979) *Structured Systems Analysis: Tools and Techniques*. Englewood Cliffs, California: Prentice-Hall.

Gartner Group (2005) *Delivering IT’s Contribution: The 2005 CIO Agenda*. Stamford, Connecticut: Gartner, Inc.

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

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

Green, P. and M. Rosemann (2004) "Applying Ontologies to Business and Systems Modeling Techniques and Perspectives: Lessons Learned," *Journal of Database Management* (15) 2, pp. 105-117.

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

Green, P., M. Rosemann, M. Indulska, and C. Manning (2004) "Candidate Interoperability Standards: An Ontological Overlap Analysis", *Technical Report*, University of Queensland, Brisbane, Australia

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

Irwin, G. and D. Turk (2005) "An Ontological Analysis of Use Case Modeling Grammar," *Journal of the Association for Information Systems* (6) 1, pp. 1-36.

Keen, C. D. and C. Lakos. (1996) “Analysis of the Design Constructs Required in Process Modelling.” *International Conference on Software Engineering: Education and Practice, Dunedin, Ireland, 1996*, pp. 434-441.

Keen, P. G. W. (1980) “MIS Research: Reference Disciplines and a Cumulative Tradition.” *1st International Conference on Information Systems, Philadelphia, Pennsylvania, 1980*, pp. 9-18.

Keller, G., M. Nüttgens, and A.-W. Scheer (1992) "Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)"", *Working Paper 89*, Institut für Wirtschaftsinformatik, Universität Saarbrücken (in German), Saarbrücken, Germany

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

Mayer, R. J., C. P. Menzel, M. K. Painter, P. S. de Witte et al. (1995) "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

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

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

Nielsen, P. and C. Main (2004) "Company Liability After the Sarbanes-Oxley Act," *Insights* (18) 10, pp. 2-12.

OASIS (2001) "ebXML Business Process Specification Schema Version 1.01," UN/CEFACT and OASIS, http://www.ebxml.org/specs/ebBPSS.pdf (March 12, 2005).

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

Ould, M. A. (1995) *Business Processes: Modelling and Analysis for Re-Engineering and Improvement*. Chichester, UK: John Wiley.

Petri, C. A. (1962) “Fundamentals of a Theory of Asynchronous Information Flow,” in C. M. Popplewell (Ed.) *IFIP Congress 62: Information Processing*, Munich, Germany: North-Holland, pp. 386-390.

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

Robinson, A. G. and D. M. Dilts (1999) "OR & ERP: a Match for the new Millenium?," *OR/MS Today* (26) 3, pp. 30-35.

Rosemann, M. and P. Green. (2000) “Integrating Multi-Perspective Views Into Ontological Analysis.” *21st International Conference on Information systems, Brisbane, Australia, 2000*, pp. 618-627.

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

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

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

Scheer, A.-W. (2000) *ARIS - Business Process Modeling*, 3rd edition. Berlin, Germany et al.: Springer.

Si, Y.-W., D. Edmond, A. H. M. ter Hofstede, and M. Dumas (2005) "Orchestrating Interrelated Trading Activities," *International Journal of Business Process Integration and Management* (1) 1, pp. 12-25.

Sinur, J. (2004) "Magic Quadrant for Business Process Analysis", *Gartner Research Note M-22-0651 March*, Gartner, Inc, Stamford, Connecticut

Soffer, P. and Y. Wand (2005) "On the Notion of Soft-Goals in Business Process Modeling," *Business Process Management Journal* (11) 6, pp. 663-679.

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

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

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

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

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

APPENDIX

Appendix 1. Constructs in the BWW Representation Model, grouped by cluster. Adapted from (Weber, 1997) 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 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.

Appendix 2. Mapping results from the representational analyses of Petri nets and BPMN

BWW Construct	Cluster	Petri nets Construct	BPMN Construct
THING	Things including properties and types of things	Place	Lane, Pool
PROPERTY		*N/A*	*N/A*
In General			Attributes of Pools, Attributes of Lanes
In Particular			
Hereditary			
Emergent			
Intrinsic			
Mutual: Non-binding			
Mutual: Binding			
Attributes			
CLASS		Place	Lane, Data Object
KIND			Lane
STATE	States assumed by things	Place, Initial Marking, Token	
CONCEIVABLE STATE SPACE			
STATE LAW		Place Capacity	
LAWFUL STATE SPACE		Place Capacity	
STABLE STATE			
UNSTABLE STATE		Place, Initial Marking, Token	
HISTORY			
EVENT	Events and transformations occurring on things	Transition	Start Event, Intermediate Event, End Event, Message, Timer, Error, Cancel, Compensation, Terminate
CONCEIVABLE EVENT SPACE			
LAWFUL EVENT SPACE			
EXTERNAL EVENT			Start Event, Intermediate Event, End Event, Message, Timer, Error, Cancel, Compensation
INTERNAL EVENT		Transition	Start Event, Intermediate Event, End Event, Message, Error, Cancel, Compensation, Terminate
WELL-DEFINED EVENT		Transition	Compensation, End Event
POORLY-DEFINED EVENT			Message, Timer, Error, Cancel, Terminate, Start Event, Intermediate Event
TRANSFORMATION		Transition	Activity, Task, Collapsed Sub-Process, Expanded Sub-Process, Nested Sub-Process, Transaction
LAWFUL TRANSFORMATION			Default Flow, Uncontrolled Flow, Exception Flow
Stability Condition		Arc weight	Rule, Conditional Flow
Corrective Action			'Exception Task', Compensation Activity
ACTS ON		Arc	Message Flow
COUPLING			Message Flow
SYSTEM	Systems structured around things		Pool, Lane
SYSTEM COMPOSITION			Pool, Lane
SYSTEM ENVIRONMENT			Pool, Lane
SYSTEM STRUCTURE			
SUBSYSTEM			Pool, Lane
SYSTEM DECOMPOSITION			Pool, Lane
LEVEL STRUCTURE			Pool, Lane
Construct excess			Link, Off-Page Connector, Gateway Types, Association Flow, Text Annotation, Group, Activity, Looping, Multiple Instances, Normal Flow, Event (super type), Gateway (super type)