Business Process Modeling- A Comparative Analysis

Article *in* Journal of the Association for Information Systems · April 2009

DOI: 10.17705/1jais.00193 · Source: OAI

4 authors:

Jan Recker
University of Hamburg

Michael Rosemann
Queensland University of Technology

Marta Indulska
The University of Queensland

Peter Green
Queensland University of Technology

Journal of the Association for Information Systems

JAIS

Research Article

Business Process Modeling- A Comparative Analysis*

Jan Recker
Faculty of Science and Technology
Queensland University of Technology
j.recker@qut.edu.au

Marta Indulska
UQ Business School
The University of Queensland
m.indulska@business.uq.edu.au

Michael Rosemann
Faculty of Science and Technology
Queensland University of Technology
m.rosemann@qut.edu.au

Peter Green
UQ Business School
The University of Queensland
p.green@business.uq.edu.au

Abstract

Many business process modeling techniques have 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. This paper comparatively assesses representational analyses of 12 popular process modeling techniques in order to provide insights into the extent to which they differ from each other. We discuss several implications of our findings. Our analysis uncovers and explores representational root causes for a number of shortcomings that remain in process modeling practice, such as lack of process decomposition and integration of business rule specification. Our findings also serve as motivation and input to future research in areas such as context-aware business process design and conventions management.

Keywords: *Business process management, Process modeling, Representation theory, BWW model*

* Yair Wand was the accepting senior editor. Joerg Evermann, Andreas Opdahl, and Pnina Soffer were the reviewers. This article was submitted on January 9, 2006 and went through two revisions.

Volume 10, Issue 4, pp. 333-363, April 2009

1. Introduction

Business process management (BPM) continues to be a top business priority, and building business process capability is still a major challenge for senior executives(Gartner Group, 2009). The interest in BPM has, *inter alia*, triggered substantial academic and commercial work aiming toward advanced business process management solutions. One prominent example in this context is increasingly popular business process modeling (Davies *et al.*, 2006). Due to a strengthened interest in a more disciplined approach to business process management, many organizations have made significant investments in process modeling initiatives, which in turn have triggered substantial related research. The recent introduction of legislation such as the Sarbanes-Oxley Act (Nielsen and Main, 2004) for example, further contributed to the increasing interest in business process modeling as a way to document 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, from simple flowcharting techniques (American National Standards Institute, 1970), to techniques initially used as part of software design such as UML (Fowler, 2004), to dedicated business-oriented modeling approaches such as Event-driven Process Chains (Scheer, 2000), to 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 (Ami and Sommer, 2007), and significant demand has been created for means to evaluate and compare the available techniques (Moody, 2005). Indeed, many available “standards” for process modeling lack rigorous evaluation (van der Aalst, 2003).

Given the keen interest in process modeling as a way of capturing the operations of organizations in real-world domains, and given the multitude of available techniques for such a task, our interest is to understand the capabilities of different process modeling techniques to facilitate the modeling of real-world business domains.

While in earlier work we examined the *evolution* of representational capabilities of process modeling techniques (Rosemann *et al.*, 2006), the aim of this paper is to study the *differences* in the representational capabilities across leading process modeling techniques. We use a theory of representation and the associated notions of ontological completeness and ontological clarity (Weber, 1997) as measurements for the study. From these overall objectives, we derive the following research questions:

1) How do process modeling techniques perform in light of representation theory?
2) What are the common concepts and key differentiators of leading process modeling techniques, measured by their levels of ontological completeness and clarity as based on the representation theory?
3) What are the key implications and what lessons can be learned from the representational analysis of leading process modeling techniques for the modeling of business processes?

We proceed as follows. The next section provides an introduction to process modeling and an overview of Wand and Weber’s representation theory, including 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 of process modeling techniques. Section 3 reports on and discusses the findings of the comparative assessment of process modeling techniques from the viewpoint of their ontological completeness and ontological clarity. The paper concludes in Section 4 with a review of contributions and a discussion of the implications and limitations of our study.

2. Background & related Work

2.1. Process Modeling

Significant attention has been paid to the role conceptual models and conceptual modeling play in the process of Information Systems development (Karimi, 1988; Wand and Weber, 2002; Garda *et al.*, 2004). Traditional forms of conceptual modeling, i.e., building a representation of selected phenomena in a problem domain for the purpose of understanding and communication among stakeholders (Mylopoulos, 1992; Siau, 2004), accounted only for the organization's data and, if at all, that portion of its processes that interacted with the data. Newer uses of information systems, however, extend deployment beyond transaction processing and into communication and coordination. This extension is known as a process-aware perspective on information systems (Dumas *et al.*, 2005), and it is this perspective that gave rise to the conceptual modeling of business processes, *viz*., process modeling.

Process modeling is widely used within organizations as a method to increase awareness and knowledge of business processes, and to deconstruct organizational complexity (Bandara *et al.*, 2005). It is an approach for describing how businesses conduct their operations and typically includes graphical depictions of at least the activities, events/states, and control flow logic that constitute a business process (Curtis *et al.*, 1992; Davenport, 2005). Additionally, process models may also include information regarding the involved data, organizational/IT resources, and potentially other artifacts such as external stakeholders and performance metrics, to name just a few (Scheer, 2000).

Existing business process modeling techniques fall into two categories (Phalp, 1998). Intuitive graphical modeling techniques such as the Event-driven Process Chain (EPC) (Scheer, 2000) are mostly concerned with capturing and understanding processes for project scoping tasks and for discussing business requirements and process improvement initiatives with subject matter experts. Conversely, other process modeling techniques such as Petri nets (Petri, 1962) are founded on mathematical, rigorous paradigms. These techniques are typically used for process analysis (Verbeek *et al.*, 2007) or process execution (van der Aalst and ter Hofstede, 2005), and can also facilitate simulation or experimentation with process scenarios (Hansen, 1996).

In considering how to model business processes, the decision of the type of notation (or technique) to be used for process modeling is an important consideration (Rosemann, 2006). This decision can be seen as essentially the same problem that software engineers encounter when carrying out analysis or design tasks. One might choose to use either structured analysis notations or object-oriented approaches. One important aspect in the consideration of a particular technique then is that different techniques have different capabilities for articulating real-world process domains. Different modeling techniques tend to emphasize diverse aspects of processes, such as activity sequencing, resource allocation, communications, or organizational responsibilities (Soffer and Wand, 2007). As an example, the Petri net model of a business domain looks considerably different from a data flow diagram or BPMN model of the same domain.

While this observation seems obvious, there is a need to understand *why* these differences exist and what implications they introduce. Furthermore, being mostly practice driven, available process modeling techniques often lack a formal theoretical foundation on which differences between the techniques can be examined (Soffer and Wand, 2007). Hence, there is a need for a theoretical framework to facilitate these explanations (Phalp, 1998; Moody, 2005). While, in general, the lack of established quality frameworks for conceptual modeling has repeatedly been noted as critical (Moody, 2005), a promising base has emerged over the last few years that builds on *representation theory* (e.g., Wand and Weber, 1990; 1993; 1995). Accordingly, to address this critical gap, we use representation theory as a means for establishing the differences among 12 leading process modeling techniques.

2.2. Representation Theory in Information Systems

Representation theory (e.g., Weber, 1997) was developed by Wand and Weber as an adaptation of an ontology proposed by Bunge (1977). The theory suggests a model of representation, known 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 the Information Systems domain. In this paper we employ this model and the associated principles of representational analysis to comparatively assess 12 popular process modeling techniques.

While a number of existing ontological models of representation (e.g., Cocchiarella, 1995; Chisholm, 1996; Guizzardi, 2005) 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., Cocchiarella, 1995; Chisholm, 1996), the BWW representation model has specifically been derived with the Information Systems discipline in mind (Weber, 1997). Second, the BWW representation model serves as an upper ontology for the modeling of Information Systems, and its foundational character and comprehensive scope allow 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; Wand and Weber, 2006), 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 Bunge's ontological model, the BWW model specifies a number of constructs that are deemed necessary to provide faithful representations of any domain to be represented by Information Systems. Therefore, these constructs should be included in any conceptual modeling technique. These constructs can be represented in a meta model. The meta model by Evermann (2009), for instance, describes the nature, type, and relationships of the ontological representation constructs using the UML and OWL formats. The comprehensiveness and detail of this meta model would suggest that this specification could provide a potential starting point for future representational analyses of modeling techniques on the basis of a meta model comparison of technique constructs to representation constructs. However, this suggestion remains to be verified empirically. The meta model by Rosemann and Green (2002) highlights 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). We deem this suggested clustering a valuable analysis framework for our work, through which the outcomes of the individual representational analyses of process modeling techniques can be assessed, which is why we selected this meta model for our forthcoming discussion.

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 (e.g., Rosemann *et al.*, 2009). Representational analysis can be used to make predictions of 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 measured by the 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 1) construct overload (m:1), or the extent to which single language constructs cover several BWW constructs, 2) construct redundancy (1:m), or the extent to which a single BWW construct maps to several language constructs, and 3) construct excess (0:1), or the extent of language constructs that do not map to any BWW construct.

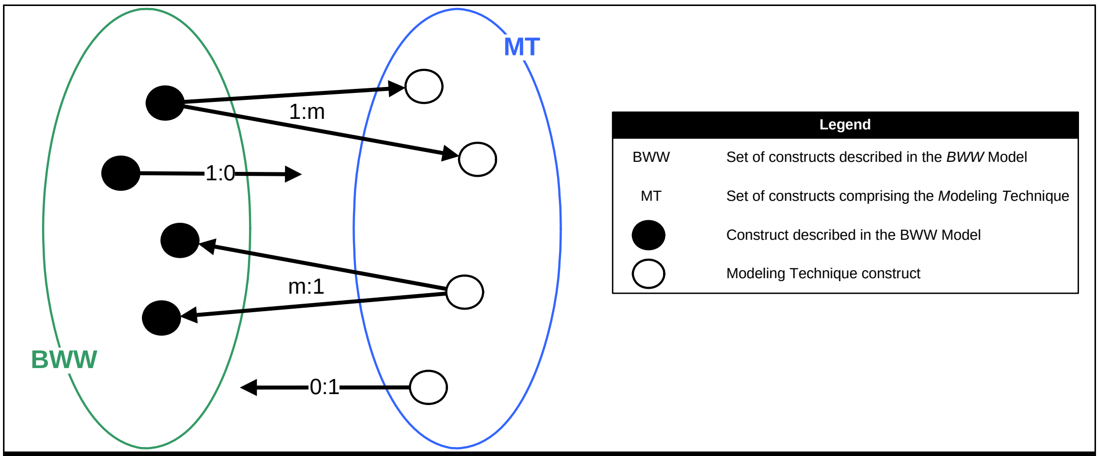

Figure 1. Potential representational deficiencies of a modeling technique. Adapted from (Weber, 1997)

Over the last 15 years, the BWW model has reached a significant level of maturity, adoption, and dissemination, and has been used in a wide range of research projects (Green and Rosemann, 2004) to evaluate different modeling techniques. The evaluated techniques cover a wide spectrum of modeling, from data modeling (Wand and Weber, 1993), to schema modeling (Weber and Zhang, 1996), to object-oriented modeling (Opdahl and Henderson-Sellers, 2002), to use case modeling (Irwin and Turk, 2005), to business modeling (Zhang *et al.*, 2007), to reference modeling (Fettke and Loos, 2007). The model also has a growing track record in the area of process modeling, with contributions coming from various researchers. We review such BWW-related studies that focus specifically on process modeling techniques in the next section.

Some criticisms have been leveled over the years at the use of representation theory, *viz.*, limited empirical testing (Wyssusek, 2006), a lack of coverage caused by the representation model focusing just on the representational algebra ("notation") of a technique, and a lack of understandability of the BWW constructs (Rosemann *et al.*, 2004). Certainly, the work to date has attempted to mitigate each of these criticisms. For instance, many authors have empirically tested the validity of the predictions stemming from representation theory (e.g., Bodart *et al.*, 2001; Green and Rosemann, 2001; Parsons and Cole, 2004; Gemino and Wand, 2005; Bowen *et al.*, 2006; Burton-Jones and Meso, 2006; Recker *et al.*, 2006; Shanks *et al.*, 2008). These studies found that the premises offered by representation theory, indeed, inform researchers about conceptual modeling activities, outcomes and success, and, moreover, leverage "better" conceptual modeling. Other researchers have undertaken efforts to provide procedural guidelines for the application of the theory (Green *et al.*, 2006; Rosemann *et al.*, 2009).

In the absence of compelling evidence in favor of a specific ontology to be used as part of a representational analysis, the final verdict about the validity of any ontology-based conclusions should be based on empirical methods and outcomes (Wand and Weber, 2006). In light of the empirical insights gained on the basis of the BWW representation model, it would appear that there is support for the usefulness, appropriateness, and validity of representation theory, which, in turn, serves as justification for the selection of this theory in the present study.

2.3. Previous Representational Analyses of Process Modeling Techniques

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 they evaluated were: ANSI flowcharts (American National Standards Institute, 1970), Data Flow Diagrams (DFD) (Gane and Sarson, 1979), IDEF Method 3 Process Description Capture Method (Mayer *et al.*, 1995), ISO/TC/97 standard for conceptual schema specification (van

Griethuysen, 1982), Merise (Tardieu, 1992), and the researchers' own Language for Object-Oriented Petri nets (LOOPN++) (Keen and Lakos, 1994). 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 (Scheer, 2000), assessing both ontological completeness and clarity. They found empirically confirmed shortcomings in the EPC notation with regard to users' ability to represent real world objects and business rules, and users' ability to clearly delineate the scope and boundary of the domain being modeled (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 that ebXML has a relatively high degree of ontological completeness.

Green *et al*. (2007) also compared different modeling standards for enterprise system interoperability, including Business Process Execution Language for Web Services v1.1 (WS-BPEL) (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, were 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, for the present study, we conducted two additional representational analyses (from the viewpoint of both ontological completeness and clarity) of process modeling techniques, namely analyses of Petri nets (Petri, 1962) and BPMN v1.0 (BPMI.org and OMG, 2006). The importance of including an analysis of Petri nets stems from the influence of this technique on a number of other modeling techniques. On the other hand, we chose to analyze BPMN because it denotes the most recently proposed notation for process modeling, one that has now been ratified by the OMG as a process modeling standard, and is backed by strong practitioner interest. A number of shortcomings related to ontological completeness and clarity were identified in terms of the use of these two techniques. For instance, in BWW terms, Petri nets lack support for the modeling of systems structured around things, and BPMN lacks capabilities to represent states assumed by things. The analyses are summarized in the form of a mapping table in Appendix 2 and have been empirically validated in the case of BPMN.[1]

Further work has used the principles of representational analysis to explore other conceptual modeling techniques (e.g., Rohde, 1995; Opdahl and Henderson-Sellers, 2001; Opdahl and Henderson-Sellers, 2002; Irwin and Turk, 2005; Fettke and Loos, 2007; Zhang *et al.*, 2007). The techniques under consideration in these analyses, however, are based on concepts different from the notion of a "process" or "activity" that is central for the decomposition and partition of a real-world system with a process modeling technique. For example, object-centric modeling techniques use the concept of an "object" as the unit for partitioning and decomposition of a real-world system (Vessey and Conger, 1994). Therefore, we have limited our analyses to activity-centric process modeling techniques. We believe that the inclusion of techniques not generally accepted as pure process modeling techniques (e.g., state-transition diagrams, OML diagrams, and Use cases) could potentially confound the results.

---

[1] 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 and Indulska (2007) and Recker *et al.* (2006).

3. COMPARISON OF REPRESENTATIONAL ANALYSES

3.1. 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 representational analyses of ten techniques, performed representational analysis of two additional techniques, and then performed a comparison of the twelve analyses, with a focus on both completeness and clarity. The analyses considered were those of Keen and Lakos (1996) (viz., ANSI flowcharts, DFD, IDEF Method 3 Process Description Capture Method, ISO/TC97, Merise), Green and Rosemann (2000) (viz., EPC), Green *et al*. (2007) (viz., BPML, WSCI, ebXML, WS-BPEL), and our own analysis of Petri nets and BPMN. For each representational deficiency situation — deficit, overload, redundancy and excess — we constructed a table into which we mapped the results of the respective analyses. Our analysis covered a wide selection of process modeling techniques, ranging from illustration methods (*e.g.*, Flowcharts) to integrated techniques (*e.g.*, EPC) and covering more recent techniques capable of both process description and execution (*e.g.*, ebXML and WS-BPEL).

In performing the representational analysis of BPMN and Petri nets, we followed an extended representational methodology that allows for maximizing the objectivity and internal validity of such work (Green *et al.*, 2006; Rosemann *et al.*, 2009). In both analyses, we also measured inter-coder reliability between the researchers, creating representation mapping results using Cohen's Kappa (Cohen, 1960). In the case of BPMN, over the mapping rounds conducted, we calculated Cohen's Kappa to be .62 in the first round and .83 in the second round (Recker *et al.*, 2007b). In the case of Petri nets, Cohen's Kappa was.69 in the first round and .92 in the second round (Recker and Indulska, 2007). These statistics exceed generally recommended Kappa levels of .6 (Moore and Benbasat, 1991).

In the comparative analysis that followed, we were concerned with minimizing potential mapping errors and general subjective bias. Therefore, we accomplished the comparison as follows. Two researchers individually extracted the mapping analyses of the selected techniques from the respective studies into four tables, one each for construct deficit, redundancy, overload and excess. The two researchers then met to compare the two versions for each mapping table and cross-checked for mapping inconsistencies. For instance, the two researchers identified an inconsistency in their consolidation of the representation mapping of Data Flow Diagrams from (Keen and Lakos, 1996). The inconsistency encountered was caused due to the use of the term "event space" in (Keen and Lakos, 1996), which potentially could refer to the two ontological constructs "lawful event space" and "conceivable event space" in the original work by Wand and Weber (1990; 1993; 1995). In the meeting, the researchers revisited both the original works and the mapping performed in (Keen and Lakos, 1996) and agreed that Keen and Lakos (1996) referred to 'conceivable event spaces'. After this stage, we again consolidated all four tables. By reaching a consensus over the consistency of the four final mapping tables, we are confident that we have demonstrated objectivity and rigor in this type of research.

Because the analyses were independently conducted by different research groups, and because representational analyses may refer to varied research purposes (Rosemann and Green, 2000), we put effort into making the individual analyses comparable. We neither questioned nor reviewed the mapping results as proposed by the different research groups. Hence, our study consolidates previous analyses instead of revises or extends them.

One point of note is the fact that analyses did not entirely differentiate between the *property* sub-types as defined in (Wand and Weber, 1993; 1995; Weber, 1997) and as defined in Appendix 1 (*viz., in general, in particular, hereditary, emergent*, *intrinsic, non-binding mutual, binding mutual,* and *attributes).* In order to enable the comparison of the studies, we had to generalize all these *property* sub-types to the super-type *property*. Therefore, if a mapping was found for a sub-type of *property*, *e.g.*, *emergent* or *binding mutual property*, then we recorded the mapping 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 constructs to the *lawful transformation construct*. We realize that this generalization for comparison purposes brings with it the limitation that some of the specialized sub-types of the *property* and *transformation* constructs that may be important for specific purposes are not incorporated in the analysis. For example, mutual properties specify attributes that two things share due to their participation in a binding relationship, and which characterize the type of relationship. These properties could be of particular interest when analyzing modeling scripts that relate to the domain of interoperability, as they could be used to specify the role or behavior of two mutually inter-dependent process entities participating in a collaborative business scenario. For instance, Green et al. (2005) describe an example of how mutual properties affect transactions in a collaboration scenario using ebXML.

A second point of note stems from the fact that we were restricted in our comparative analysis to 1:1 mappings between constructs in the modeling technique and constructs in the BWW representation model. While, in general, representation theory allows for the comparison of BWW model constructs to a combination of several technique constructs (1:n mappings) (Wand and Weber, 1993), or even vice versa, representational analyses typically are restricted to 1:1 comparisons. All of the studies that we examine in this paper were restricted to 1:1 mappings. This situation, in turn, posits a limitation of our study. It would, indeed, be interesting and challenging to examine how different process modeling techniques employ production rules to form ontologically meaningful clusters of technique constructs.[2] Yet, we cannot consider the potentially unlimited variety of construct compositions across all techniques in our study.

A final point of note in the consolidation and comparison is related to the shortcoming of analyses focusing on both ontological completeness and clarity. As for the investigation of the ontological clarity of process modeling techniques — in particular construct excess, redundancy, and overload — we reduced the number of techniques considered in the analysis. This reduction is due to a lack of consideration of aspects of ontological clarity in the study of ANSI Flowcharts, ISO/TC97, MERISE, DFD, and IDEF3, as the evaluation performed by Keen and Lakos (1996) was restricted to ontological completeness only.

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

3.2. 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 completeness (DoC), or the extent to which process modeling techniques are able to provide complete descriptions of a real-world domain. DoC can be measured relatively as one minus the degree of deficit, with the degree of deficit being 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). 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 assumption may not always hold true in modeling practice (Rosemann *et al.*, 2004); however, the selected metric also allows for the derivation of weighted measurements.

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.

---

[2] An example of how such a task can be approached is illustrated in (Soffer *et al.*, 2007).

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 1.01	BPML 1.0	WSCI 1.0	WS-BPEL 1.1	BPMN 1.0
	1962	1970	1979	1982	1992	1992	1995	2001	2002	2002	2003	2004
Things including Properties and Types of Things												
THING	✓			✓	✓		✓					✓
CLASS	✓							✓	✓	✓	✓	✓
KIND												✓
PROPERTY			✓			✓	✓	✓	✓	✓	✓	✓
Cluster Degree of Deficit	*50.0 %*	*100 %*	*75.0 %*	*75.0 %*	*75.0 %*	*75.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*0.0 %*
Cluster Degree of Completeness	*50.0 %*	*0.0 %*	*25.0 %*	*25.0 %*	*25.0 %*	*25.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*50.0 %*	*100 %*
States assumed by Things												
STATE	✓					✓	✓	✓	✓	✓	✓	
CONCEIVABLE STATE SPACE								✓				
STATE LAW	✓			✓	✓	✓		✓				
LAWFUL STATE SPACE	✓							✓				
STABLE STATE						✓		✓				
UNSTABLE STATE	✓							✓				
HISTORY								✓				
Cluster Degree of Deficit	*42.9 %*	*100 %*	*100 %*	*85.7 %*	*85.7 %*	*57.1 %*	*85.7 %*	*0.0 %*	*85.7 %*	*85.7 %*	*85.7 %*	*100 %*
Cluster Degree of Completeness	*57.1 %*	*0.0 %*	*0.0 %*	*14.3 %*	*14.3 %*	*42.9 %*	*14.3 %*	*100 %*	*14.3 %*	*14.3 %*	*14.3 %*	*0.0 %*
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		✓	✓		✓		✓	✓		✓	✓	✓
Cluster Degree of Deficit	*45.5 %*	*81.8 %*	*81.8 %*	*54.5 %*	*45.5 %*	*45.5 %*	*72.7 %*	*9.1 %*	*36.4 %*	*18.2 %*	*18.2 %*	*18.2 %*
Cluster Degree of Completeness	*54.5 %*	*18.2 %*	*18.2 %*	*45.5 %*	*54.5 %*	*54.5 %*	*27.3 %*	*90.9 %*	*63.6 %*	*81.8 %*	*81.8 %*	*81.8 %*
Systems structured around Things												
SYSTEM			✓		✓		✓	✓		✓	✓	✓
SYSTEM COMPOSITION			✓		✓		✓			✓	✓	✓
SYSTEM ENVIRONMENT			✓									✓
SYSTEM STRUCTURE					✓		✓			✓	✓	
SUBSYSTEM								✓				✓
SYSTEM DECOMPOSITION			✓				✓					✓
LEVEL STRUCTURE			✓			✓	✓					✓
Cluster Degree of Deficit	*100 %*	*100 %*	*28.6 %*	*100 %*	*57.1 %*	*85.7 %*	*28.6 %*	*71.4 %*	*100 %*	*57.1 %*	*57.1 %*	*14.3 %*
Cluster Degree of Completeness	*0.0 %*	*0.0 %*	*71.4 %*	*0.0 %*	*42.9 %*	*14.3 %*	*71.4 %*	*28.6 %*	*0.0 %*	*42.9 %*	*42.9 %*	*85.7 %*
Total 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 %**
Total 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 consideration. 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 low degree of completeness (25%). Also, the poor performances of Flowcharts (0%) and DFD (25%) are notable given their historically 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).

The move from things to classes of things can be seen as a strong shift toward an understanding that processes are performed by a class of things with generally common properties, rather than specific things with unique properties. This situation can be seen, for instance, in the increased application of process analysis techniques (such as Activity-based costing, root-cause analysis, Pareto analysis) that consider classes of processes with common properties (*e.g.,* sets of processes for VIP customers, processes with different types of involved business objects) instead of dedicated process instances.

Nevertheless, the overall limited coverage of things and classes of things in business process models is still an issue. The limited coverage of things and classes of things inhibits users from decomposing processes according to the properties of things (*e.g.,* what are the differences in the processes for handling domestic vs. international invoices?) instead of functional decomposition (*e.g.,* what are the detailed steps of invoice verification?). Anticipating a move toward better support for things as well as classes of things in process modeling, a technique would allow users to craft process models that can be used as direct input for advanced analysis techniques such as root-cause and process cost analysis.

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 (see, for instance, 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 modeling will be unclear to the modeler when trying to determine which set of states can potentially occur in a process and which states are possible but should not be allowed. This shortcoming is one explanation for the often limited capabilities of process modeling techniques in supporting exception handling. Exception handling requires semantically richer process models in which certain states are classified as exceptions, i.e., deviations from the expected daily practices (Russell *et al.*, 2006). Lack of representational capability in the cluster *states assumed by things* is a root cause of these current limitations. For instance, to represent exceptions, we need, in particular, *lawful states* and *lawful state spaces.* Representational support for these concepts would allow for the definition of the particular set of state vectors of the domain in which a process operates. *A lawful transformation* should be enacted (i.e., an exception handling routine triggered) so that reaching an *unlawful state* can be prevented (i.e., a state in which the process system cannot faithfully terminate).

By having representations for states and transformations, clarity can also be given to the representations, and implications, of process-relevant events. Indeed, if a technique had representations for states and state changes, it would be possible to deduce relevant events from this information. For instance, in State Charts, it is possible to derive transitions based on the representation of states and state changes. Thus, representation for states and transformation could mitigate a potential deficit in representing events.

Moreover, having a representation of state-based concepts would allow process model users to answer questions pertaining to business rule specification (when should which process change be effected?), exception handling (what are the event-driven changes that regular process transformations do not cover?), or process recovery (what was the state of a process object before an event occurred that resulted in a process failure?). In other words, advanced representation support for the essential concepts of states underlying exception handling procedures would allow users to specify conceptually a number of process change strategies in the occurrence of exceptional events impacting day-to-day business operations. A closer look at Table 1 reveals, indeed, that most techniques have a very low degree of completeness in the cluster of *states assumed by things* (see, for instance, Flowcharts, IDEF3, and BPMN), except for ebXML (100% in this cluster) and Petri nets (52% in this cluster). One particular issue here is the limited support for the *history* construct. Available techniques appear to support the design of process models with only limited consideration of the traceability of the process objects that are the focus of the models. The specification of the history of states that a process object has traversed through its lifecycle could be leveraged for a range of areas of process-related decision-making scenarios, especially in the context of business rule management. Consider the case of credit history checks or customer relationship management processes, where key decisions are made, and special rules applied, on the basis of the history of the relevant process object (e.g., a credit card applicant or a frequent flier member). These and similar scenarios point to the area of business rule specification, which is dependent on accurate specification of not only the current process flow but also the nature and history of the objects that have traversed through the process previously. Anticipating better representational support for state-based concepts — in particular, the history of state changes — could, thus, be leveraged for a closer integration of business rule specification with process modeling practice. Both business rule modeling and business process modeling are used to document organizational policies and procedures. Yet, very little synergy and overlap has been identified. Indeed, the lack of support for state and history constructs in process modeling techniques can be seen as one root cause for this dilemma. Anticipating better representational support could lead to an advanced understanding of the relationship between the two modeling types and allow organizations to maximize synergies and reduce their modeling efforts. One alternative application area of the understanding of the representational differences would be a clear demarcation of business process and business rule modeling, their purposes and touch points, and to see both as complementary yet orthogonal views (Herbst *et al.*, 1994; Kovacic, 2004). Techniques could then be developed that, together, provide maximal representational coverage while sharing minimal representational overlap (Green *et al.*, 2007).

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 argument that things, events, and transformations are core concepts in process modeling (Soffer and Wand, 2005; 2007). An interesting observation can be made with respect to the degrees of completeness of Flowcharts (18%), DFD (18%), and IDEF3 (29%). We speculate that the relatively low degrees of completeness 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, they 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 (DoC: 91%). Moreover, it denotes the single technique capable of depicting both conceivable and lawful event spaces. In other words, there is a realization that it is important to give an indication clearly of allowable states that a thing can take on as a result of suffering an event.

Contrasting the representations of events and transformations in process modeling techniques with the representations for states assumed by things, the overall representational support for events and transformations may mitigate the lack of representation for states, and indeed, may even explain the lack of state-based constructs in most techniques considered. Indeed, given the complimentary nature of events, states, and transformations, it may be possible that state representations in process modeling techniques might not be required to achieve sufficient completeness.

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, *viz.*, *system environment, system structure, subsystem, system decomposition, and level structure*. 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. For example, these three techniques have in common dedicated language constructs to model system decompositions (e.g., the Pool or Lane construct in BPMN). Constructs to model system decompositions allow users to present a composition view that articulates the components of which a system is composed. While the ontological concept of system decomposition relates to the things within that system, it is reasonable that a sub-process will modify at most the domain in which the main process operates, and usually only these sub-parts of this domain. Thus, language constructs that support process decomposition point indirectly at supporting system decomposition in the ontological sense, and vice versa.

There are at least two critical implications of the limited support for depicting the organizational and wider setting of a modeled process (for instance, only two out of 12 techniques support the “system environment” construct). First, process models tend to be decoupled from their surrounding system, and thus, the design of context-aware process models becomes impossible. Context-aware process models have explicit relationships with external factors (e.g., time, location, weather, market conditions, etc.), and these relationships allow changes in the factors and the related process changes to be anticipated (Rosemann *et al.*, 2008). However, moving process modeling toward awareness of its organizational and contextual setting, and the potential change drivers within, would require a strong representation support in the modeling techniques such that the relevant environmental aspects of the systems in which a process is embedded and operates can be specified conceptually. Second, modeled processes without consideration of their wider organizational setting are more vulnerable to unexpected behaviors in the environment. Unexpected changes in process environments require the instantiation and execution of well-defined exception handling strategies to cope with the change. Currently, support for exception handling is not fully present in the conceptual specification of processes (Russell et al., 2006) or in the methods available for process verification. Current approaches for model verification (e.g., van der Aalst, 1999; Verbeek *et al.*, 2007) focus on internal structure properties such as soundness, liveness, or boundedness (Dehnert and Zimmermann, 2005), while disregarding the external stimuli of process behavior. This finding calls for further research in the areas of conceptual process specification and structural process verification.

3.3. 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 construct. In light of the underlying representation 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 is 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 the leading process modeling techniques we consider, 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 analyze construct redundancy for properties. Hence, in Table 2, an “x” indicates that the respective

Table 2. Comparison of construct redundancy of process modeling techniques

Language / Version / Year — BWW Construct	Petri net	EPC	ebXML 1.01	BPML 1.0	WSCI 1.0	WS-BPEL 1.1	BPMN 1.0
	1962	1992	2001	2002	2002	2003	2004
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 %

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 because we relied on earlier studies that did not differentiate the analysis of events and states into their respective sub-types.

In terms of *things*, their *types* and *properties*, it generally 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, specifically the Pool construct (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. Indeed, confusion in the relationship may exist even when there is only one class construct. For example, it may be unclear under what circumstances an instance of a DocumentEnvelope is used by RequestingBusinessActivity (Green *et al.*, 2005). Yet, because ebXML BPSS provides more than one construct to represent classes of things, the potential confusion is amplified.

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. We can make two points. First, Petri nets appear to have redundant constructs for modeling the states of *things* in light of the BWW representation model, particularly, unstable states. Specifically, our own analysis of construct redundancy in Petri nets revealed that they have three different concepts for representing the (unstable) state of a thing: 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, to potentially cause confusion in the use of this technique. It may be worthwhile to consider reducing the range of constructs available to a more limited set that avoids this redundancy. Clearly, the answer to this question requires empirical investigation.

We found that constructs for representing events and transformations occurring on things 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). Indeed, the use of external events may help to mitigate the lack of constructs in some of the techniques for system environment, in that the external events allow us to model the impact of the environment, if not the environment itself.

Similarly, we found the *lawful transformation* construct to be mapped to more than one language construct for 57% of the considered techniques, sometimes even to 10 or more constructs, as in BPML and WS-BPEL. A possible 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 representation theory. In other words, these additional constructs may provide further information on “how” to undertake the transformation rather than enhancing the meaning of the transformation. 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 such as EPC (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.

The increasing amount of construct redundancy in more recent techniques such as WS-BPEL, WSCI, or BPMN can be seen as a reaction to the increasing number of purposes of business process modeling. Traditionally, process modeling has been used to assist stakeholder communication and to specify business requirements for process improvements. In recent years, process modeling has also become popular in advanced application areas such as compliance management (Sadiq *et al.*, 2007), Enterprise Systems configuration (Dreiling *et al.*, 2008), simulation (Gregoriades and Sutcliffe, 2008), and software design (Ouyang *et al.*, 2009). In light of this trend, it will become practically impossible for vendors of process modeling support to sell the idea of a “one size fits all” approach. Rather, it can be assumed that the practical adoption of process modeling techniques now follows a typical two-stage approach. First, techniques are selected based on their completeness (see Table 1). Second, company-specific conventions are defined that customize (and often reduce) the increasingly rich modeling techniques for the very specific demands of an organization (Rosemann, 1998). As Table 2 shows, the market has reacted by providing modeling techniques with embedded redundancies that often can be seen as detailed variants of the same construct for different purposes (consider, for instance, the differentiation of BPMN into a core and an extended set). While the move toward richer sets of process modeling constructs may be seen as an advancement in this area, it also clearly highlights the need for more research on convention management, viz., how to manage the increasing redundancies in modeling techniques and how to establish a common core of process modeling concepts to be taught and used by business analysts. In fact, recent empirical studies show that users avoid the available multiplicity of constructs and instead tend to use a very restricted set of constructs (zur Muehlen and Recker, 2008). Clearly, more research is needed into the nature, management, and use of modeling conventions that are more and more frequently employed to manage the representation complexity afforded by advanced process modeling solutions.

3.4. 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 in the BWW representation model. Accordingly, users will be unclear as to the real-world situations for which they use these constructs as representations and, thus, they 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 serves as another indication of its capability 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 as per the BWW model is 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	WS-BPEL	BPMN
Language	Petri net	EPC	ebXML	BPML	WSCI	WS-BPEL	BPMN
Version			1.01	1.0	1.0	1.1	1.0
Year	1962	1992	2001	2002	2002	2003	2004
Proposed excess constructs		AND connector	Performs	All	All	Scope	Parallel
		OR connector	Business Activity	For Each	For Each	Message Properties	Inclusive OR
		XOR connector	Business State	Choice	Selector	Message Definitions	Event-based XOR
			Business Action	Sequence	Sequence	Sequence	Data-based XOR
			Document Security	Delay	Join	Flow	Group
			Completion State	Empty	Empty	Empty	Activity Looping
			Enumeration Status	Identity	Correlate		Multiple Instances
				Spawn	Spawn		Normal Flow
				Context	Context		Event
				Activity Type			Gateway
				Synch			Text Annotation
				Parameters			Association Flow
				Activity Instance State			Off-page Connector
							Complex
							Link
Degree of Excess	0.0 %	42.9 %	13.7 %	28.3 %	18.4 %	12.8 %	38.5 %

Perusal of Table 3 suggests that there is a lack of a process modeling "common definition" as the different techniques use different terms to specify constructs (e.g., OR connector, Selector, and Choice). Furthermore, unlike in the data modeling community, the process modeling community has no generally accepted differentiation into conceptual, logical, and physical layers of modeling. As Table 3 indicates, conceptual constructs such as business activity (ebXML) are compared with constructs on the physical layer (e.g. activity instance state). This situation clearly demonstrates the need for the development of a common definition for processes that provides a meta-standard as well as well-specified layers of abstractions for the development and comparison of process modeling techniques. The representation theory that underlies our study might be a first potential candidate that could inform the development of such a common definition. A first attempt is provided in (Soffer and Wand, 2007).

In more detail, 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. Indeed, from a low-level perspective, these constructs bear no real-world meaning at all. The real-world meaning of these constructs is only revealed when examined in the wider setting of a control flow pattern (such as mergers, joins and splits) (Soffer *et al.*, 2007). 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 per se. Candidates for these scenarios include Off-page Connector, Group, and Text Annotation, which define means to link models or group model elements or attach additional descriptions to models. Our research findings suggest that these elements should be removed from the respective technique and that they should be provided by the supporting modeling tool. Thereby, the act of modeling can be supported through constructs such as text annotation, grouping elements or others in a techniqueindependent fashion, while the technique itself merely contains domain representation constructs. This situation would lead to reduced levels of complexity in the usage of the technique, and it would allow a user to choose for himself/herself whether or not such elements are required in his/her process modeling work.

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 (WS-BPEL), Spawn (WSCI), or Multiple Instances (BPMN) — all have in common that they capture certain aspects of process implementation and execution (which are required by technical analysts seeking to develop and deploy process execution engines) but they are not the conceptual “deep structure” of domain phenomena. Again, for the purpose of describing semantics of the modeled domain (which would be the task of business or process analysts in earlier stages of an IS development project), these constructs may be considered unnecessary. This situation has major implications for 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 vs. technical analyst) or purposes (*e.g.*, documenting business requirements vs. specifying system requirements). Indeed, in the early stages of process modeling projects concerned with scoping, documentation, and communication, our findings suggest that the proposed excess constructs should be excluded from process modeling, perhaps via a related policy established in conventions management. Again, our findings motivate further research in this area.

3.5. 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 interpretation 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 serves as a further indication of its capability 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 be represented as mutually exclusive sub-types of *events* (internal/external, well-defined/poorly-defined) and *states* (stable/unstable) without being considered 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.

It appears that process modeling techniques are quite diverse in their levels of construct overload. For instance, the same deliberately flexible specification that affords Petri nets a considerably high level of 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) and specification precision (and, therefore, intuitiveness due to precise semantics) of modeling constructs appears to be a recurring pattern that must be managed by designers in the development of modeling techniques. The BWW model facilitates the generation of related propositions in that it advocates the clarity of a specification. In other words, our findings

Table 4. Comparison of construct overload of process modeling techniques

Language / Version / Year	Petri net	EPC	ebXML 1.01	BPML 1.0	WSCI 1.0	WS-BPEL 1.1	BPMN 1.0
	1962	1992	2001	2002	2002	2003	2004
Proposed overloaded constructs	Place Place Capacity Transition	Function Type Event Type	Binary Collaboration		Connect Model	Partners	Lane Pool 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 %

indicate that the extra effort required for specifying the representational capacity in which overloaded constructs are used diminishes the ease with which these models can be built. Moreover, better support for differentiating the multiple purposes for which these constructs can be used (*e.g.,* by adding additional graphical markers) would appear advisable. Another option is to clearly specify the specific semantic capacity in which a construct in question is to be used. Both options can be expected to lead to improved ease of process modeling.

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). Such flexibility is only seemingly an advantage and can result in a model that users cannot easily interpret. Empirical findings from other related analyses support this view such as 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, it would appear that modelers using this technique are not required to bring in extra model knowledge to the modeling task, and we can further assume that the understandability of the resulting BPML models is relatively high. Again, this question is an empirical one.

Similar to the identified implications related to construct redundancy, we see these deficits as expressing demand for more research on convention management. Again, the identified deficiencies require a choice on the part of the organizations adopting a process modeling technique. For example, this situation is part of convention management, which has been largely neglected as a focus of academic analysis. A related research stream could investigate how the complexity of process modeling techniques can be reduced by leveraging features available in state-of-the-art process visualization engines.

3.6. Consolidation and Synopsis

We seek now to provide a consolidated picture of the overall representational capability of the analyzed techniques. In particular, we are interested in identifying the relationship between the ontological completeness of the techniques (measured by the degree of completeness (DoC), viz., one minus the degree of deficit) and their ontological clarity. This relationship allows us to identify the costs (in terms of the clarity – or lack thereof – of the technique specification) of obtaining a certain scope of coverage (measured by the degree of completeness) in a technique. Standardizing the scope of coverage (*i.e.,* the degree of completeness) across all techniques considered also allows us to comparatively assess the associated representational costs of these techniques (*i.e.,* their levels of construct redundancy, excess, and overload). Accordingly, for each technique considered, we divided DoR, DoE and DoO by DoC, and then calculated the average total factor score of these three measures to obtain a relative "lack-of-clarity-to-coverage" measure. Table 5 presents the results in decreasing order of the average lack-of-clarity-to-coverage ratio.

Table 5. Relative lack-of-clarity-to-coverage for process modeling techniques

Technique (year)	DoC	DoR (DoR / DoC)	DoE (DoE / DoC)	DoO (DoO / DoC)	Average 'lack-of-clarity-to-coverage'
ebXML 1.01	72.41%	15.69% (21.67%)	13.73% (18.96%)	1.96% (2.71%)	14.45%
WS-BPEL 1.1	51.72%	31.91% (61.70%)	12.77% (24.69%)	2.13% (4.12%)	30.17%
WSCI 1.0	51.72%	30.61% (59.18%)	18.37% (35.52%)	4.08% (7.89%)	34.20%
BPML 1.0	34.48%	30.43% (88.25%)	28.26% (81.96%)	0.00% (0.00%)	56.74%
Petri nets	41.38%	28.57% (69.04%)	0.00% (0.00%)	42.86% (103.58%)	57.54%
BPMN 1.0	65.52%	51.28% (78.27%)	38.46% (58.70%)	25.64% (39.13%)	58.70%
EPC	37.93%	0.00% (0.00%)	42.86% (113.00%)	28.57% (75.32%)	62.77%

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 high a degree of completeness as possible. Representation theory also states that process modeling techniques should be clear in their capabilities to facilitate representations of real-world domains, *viz.*, they should have relatively low degrees of redundancy, excess, and overload. As Table 5 indicates, however, this is not the case for all techniques.

A number of interesting insights can be derived in terms of the representational costs of process modeling techniques (as measured by the clarity-of-coverage ratio). Clearly, the capability of ebXML is closest to the general principles of representation theory, as its comparatively high degree of completeness (72.41%) is complemented by low relative degrees of redundancy (DoR/DoC: 21.67%), excess (DoE/DoC: 18.96%), and overload (DoO/DoC: 2.71%). Correspondingly, the average lack-of-clarity-to-coverage is roughly 14.45%. This suggests that the use of ebXML not only enables modelers to create reasonably complete descriptions of real-world domains but also relatively clear descriptions with little complexity and ambiguity. The second most complete technique (DoC of 65.52%), BPMN, on the other hand, achieves relatively poor measures across all clarity aspects when normalized (DoR/DoC: 78.27%; DoE/DoC: 58.70%; DoO/DoC: 39.13%). In sum, BPMN has an average lack-of-clarity-to-coverage ratio of 58.70%, ranking second to last in the set of techniques considered. Thus, the use of BPMN can be expected to lead to quite complete but potentially unclear and ambiguous representations of real-world domains. Users of BPMN can expect to be required to make extra efforts and bring knowledge external to the model when creating and interpreting BPMN diagrams (the study in Recker *et al*., 2006 provides empirical support for this observation).

Overall, three clusters of techniques can be identified through perusal of Table 5. One set of techniques, including ebXML, WS-BPEL, and WSCI, achieves good average costs-of-clarity because their degrees of redundancy, excess, and overload are reasonably compensated by comparatively high degrees of completeness. A second set of techniques, including Petri nets, EPC, and BPML, are afforded relatively high costs-of-clarity. This situation is because their degrees of redundancy, excess, and overload, when normalized by DoC, are not compensated by a high degree of completeness. Instead, the relative clarity decreases due to limited scope of coverage. A third set of techniques, viz., BPMN, would have been expected to achieve a reasonably high lack-of-clarity-to-coverage ratio due to its high degree of completeness. The comparatively high degrees of lack-of-clarity (measured by DoR, DoE, and DoO), however, overcompensate for the scope of coverage and lead to a comparatively low lack-of-clarity-to-coverage ratio overall.

Two more interesting patterns can be observed from Table 5. First, some techniques, such as Petri nets, exhibit low degrees of redundancy (28.57) and excess (0.00%) with high degrees of overload (42.86%). 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 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 situation, a second set of techniques, such as BPML or WSCI, achieve a relatively low degree of overload (e.g., BPML: 0.00%) and higher degrees of redundancy (e.g., BPML: 30.43%) and excess (e.g., BPML: 28.26%). The observable underlying technique design principle is 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. Consequently, such techniques achieve a certain scope of coverage through a multitude of constructs, which in turn, *prima facie*, offer a great many choices for representing the real-world phenomena the user seeks to describe. Such a design principle seems to be based on technique extension rather than revision and clarification. Based on these observations, it would appear that earlier developed techniques (such as Petri Nets or EPC) were frequently overloaded yet not excessive or redundant, which would indicate that they were intended for a restricted set of modeling purposes. More recent techniques (such as BPML, BPMN, or WSCI) appear to have been designed to fit a wider variety of process modeling purposes beyond typical communication and requirements specification purposes. Also, more recent process modeling techniques face the challenge of having to persuade existing user communities to become adopters of the new technique (Recker and Dreiling, 2007). One means of enabling the switch to a new technique could be to provide backward compatibility. In terms of process modeling techniques, backward compatibility could entice designers to add to the stack of constructs already in use, or to provide subtypes for existing constructs, rather than to revisit, amend, or potentially eliminate constructs already in use. Both rationales — extended application purposes and backward compatibility — are potential explanations for the notable rise of degrees of redundancy and excess in recent process modeling techniques.

In conclusion, the consolidated overview of the representational capabilities of process modeling techniques in Table 5 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 the modeling role occupied by a modeling stakeholder (e.g., process modeler, model user, process modeling coach) or the modeling purpose of the modeling initiative (e.g., to analyze a process, to document a process, to improve a process), a technique that is potentially redundant in its use may or may not be favored over 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 through 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.

4. Contributions

This paper presents a comprehensive comparative study of previous representational analyses of process modeling techniques, and includes 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 about the signs of representational "goodness," as measured by the degrees of completeness, excess, overload and redundancy of process modeling techniques.
Our examination delivers a comprehensive picture of the capabilities of process modeling techniques. Our findings can be used for a root cause analysis of some prevalent hurdles in current process modeling practices, such as lack of support for process decomposition, integration with business rule specification, and the development and management of organizational modeling conventions.

4.1. Implications for Practice

The outcomes of this study can be of interest to both developers and users of process modeling techniques. Developers should be motivated to examine representational analyses of existing process modeling techniques in order to build upon these techniques 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 the most recent process modeling techniques provide a rather wide scope of coverage, indicated by their high degrees of completeness. This finding suggests that the effectiveness and application of process modeling techniques, and process modeling, overall, 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 toward a widened scope of coverage that induces increased modeling complexity, as indicated by the high degrees of overload, excess, and redundancy of more recent techniques such as BPML or BPMN (see Table 5). For example, the upward trend of construct redundancy from EPCs to ebXML, WS-BPEL, WSCI and BPMN (see Table 5) points to 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 than it is to delete or replace them (Recker *et al.*, 2007a). 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. Developers can potentially counteract the indicated trend toward technique complexity while still enabling sufficient domain coverage.

Across the four types of deficits that we analyzed in this paper, the identified representational shortcomings appear to underlie many of the current issues in the practical application of process modeling. While additional evidence is required to further examine the root causes of current process modeling issues, our analysis suggests that a number of potential root causes for such issues may be related to the deficiencies identified in current process modeling techniques. For example, while we acknowledge that further proof is required to define system decomposition and process decomposition shortcomings, our analysis indicates that overcoming deficits related to modeling classes of things could potentially lead to better representational support for decomposition principles. For instance, representational support many articulate things with different properties, such as domestic vs. international invoices, A- vs. B-types of procurement goods, regular vs. first-class costumers etc. Better support, in turn, may enable users to articulate faithfully composed models of the domains in which processes operate. Such a solution could ultimately assist process modelers in the design of multi-level process architectures, or of process variants for different involved things. The identified deficits in depicting states appear to explain the limited exception handling capabilities as well as the lack of support for modeling the history of process objects. This is an obvious obstacle to achieving a better integration with business rule specification to comprehensively cover organizational processes and related policies. A further limitation in the actual application of process modeling is the challenge of adequate convention management. As our analysis showed, the increased construct redundancy, overload and excess more recent process modeling techniques (see Table 5) made the management of company-specific modeling conventions even more important. Our own anecdotal examination of process modeling practice, however, indicates that this important aspect of process modeling initiatives appears to be typically overlooked. Organizations should be motivated to address this critical challenge that has gained in significance because process modeling is utilized for a wide variety of purposes today.

Last, the findings related to the availability of excess constructs in process modeling techniques can inform process modeling practitioners about the type of constructs that should be provided at various stages of a modeling project. For instance, managers in charge of modeling conventions can use the findings to define a restricted set of process modeling technique constructs to be used in project stages concerned with domain representation (e.g., scoping, documentation, and communication).

In addition to addressing these shortcomings, we further argue that our representational analysis has the power to trigger innovative extensions of process models. An example is the design of context-aware process models (Rosemann *et al.*, 2008). Context-awareness, however, requires overcoming current limitations with respect to modeling environmental system elements that are related to processes. Envisaging better representational coverage, this move could lead to a more comprehensive conceptual specification of a process including relevant contextual change drivers, which ultimately will assist in process flexibility and change implementation.

4.2. Implications for Research

In addition to its practical merits, our work serves both as motivation and input to the extension of process modeling-related research. In particular, the uncovered representational issues can trigger a number of related design science efforts (Hevner *et al.*, 2004) dedicated to improving and extending current process modeling approaches such that some of the existing limitations can be counteracted. The field of context-awareness in process management (Rosemann *et al.*, 2008), the challenge of process decomposition in large-scale initiatives (Raduescu *et al.*, 2006), and the integration of process with business rule specification (zur Muehlen and Indulska, 2009) are only three examples of emerging IS research topics related to process modeling that can leverage the findings from our analysis. In Table 6 we suggest some specific programs of research based on the findings reported in this paper.

At present, little is known about process modeling practice and process modeling technique usage overall. We believe that we have laid the groundwork for extensive empirical research into process modeling practice. Some of the conjectures we derived from our conceptual analysis (e.g., the question of redundant event and transformation articulations, the move toward conceptual specification of process classes rather than instances, etc.) call for appropriate empirical research strategies that further operationalize and test our propositions. In particular, future empirical research could address the potential consequences of the deficiencies that have been discovered through our analysis of the various techniques.

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. Our analysis showed that insights into the nature problems of current process modeling techniques can be generated based on the premises of representation theory. The possible consequences for process modeling we derived based on representation theory allow fellow colleagues to generate ideas and to proceed and test empirically whether the consequences manifest in process modeling practice.

However, our analysis also indicates that there may be areas of the theory where further work is needed, *e.g.*, in the area of event and transformation specializations or in the handling of process orchestration concepts such as parallel splits and exclusive routing. 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. And, indeed, a number of researchers have in the recent past started to address some of the challenges we identified (e.g., Soffer *et al.*, 2007; zur Muehlen and Indulska, 2009).

We believe that further work is required on a theory of process and system decomposition. Similar to the work on faithful decomposition of object-oriented systems (Burton-Jones and Meso, 2006), more research is needed to understand how processes and process-oriented systems can be decomposed on the basis of the principles of representation theory.

Table 6. Advocated programs of research

Area	*Description*
Process decomposition	Process decomposition is a vital element in large-scale initiatives. Current techniques provide little support for breaking down complex scenarios into smaller, manageable models. Representation theory stipulates principles of good decomposition (Wand and Weber, 1990; Burton-Jones and Meso, 2006) that can be used to provide better modeling support for large-scale process initiatives.
Process and Business rule specification	Little work has been done to understand the synergies and overlap between using business rule specification vs. process modeling for the documentation of organizational policies and procedures. Based on the representational capabilities of process modeling techniques and business rule specification techniques, the relationship between the two modeling types can be researched to allow organizations to maximize synergies and reduce their modeling efforts.
Differentiating the act of modeling from domain representation	Further research could be carried out to provide a differentiation of process modeling into the conceptual, logical, and physical stages of modeling, similar to the data modeling discipline. Currently, techniques provide a variety of constructs that are relevant only to certain stages of a modeling project (e.g., only at process documentation stages, or only at process implementation stages). Following the premise of construct excess, research could examine which domain representation constructs should be provided at which stage of modeling, and which constructs are required to support the act of modeling at a later, more implementation-oriented stage of modeling.
Context-awareness	Little research has focused on the extrinsic drivers in the environmental setting of a process that, in light of changes and perturbations, require processes to adapt to the new situation. Current process modeling techniques only capture the reactive, intrinsic part of process flexibility, but lack contextualization, i.e., the stimulus for change. Representational support for conceptualizing the system and environment in which a process is embedded can be a starting point for the specification of context-aware and truly agile processes.
Conventions management	The increased number of application areas for process modeling, and the increasing complexity of process modeling techniques, induce a strong need for organizational policies and guidelines for managing this complexity in process initiatives. Virtually no research yet exists that taps into procedural guidelines (the process of process modeling), the requirements of process modeling for different application areas and stakeholder groups, or the impact of layout and naming conventions on process model understandability (let alone project success).

4.3. 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 subject to the interpretations of the researcher (Rosemann *et al.*, 2004). Therefore, we spent considerable effort to make the individual mapping results comparable. Second, we limited the considered representational analyses to studies based on the BWW representation model, which, in turn, constrained 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 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 modeling technique quality. For instance, cognitive aspects need to be taken into consideration when seeking to examine the effects of lack of ontological completeness or clarity on a user working with a modeling technique (e.g., Gemino and Wand, 2005). To that end, empirical work needs to be conducted to test predictions resulting from the evaluation of modeling techniques to determine whether deficiencies actually have consequences or not.

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 we cannot claim that the selected sample is 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 having to translate findings from different theoretical bases. Fourth, our research denotes a form of analytical study, which can only result in theoretical propositions. The findings call for appropriate empirical research strategies in order to confirm or falsify the implications drawn from our analysis. However, as one of our contributions, we have developed some conjectures based on our findings that require further operationalization and testing. We invite other researchers to also contribute in this field of study.

Acknowledgements

We would like to thank the editors and reviewers of CAiSE 2006 who provided stimulating feedback on an earlier version of this paper. We would also like to thank Yair Wand and the JAIS review team for their comments that assisted significantly in improving this paper.

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

Ami, T. and R. Sommer (2007) "Comparison and Evaluation of Business Process Modelling and Management Tools", *International Journal of Services and Standards* (3)2, pp. 249-261

Andrews, T. 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 (current February 10, 2006)

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

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

Bandara, W., G. G. Gable, and M. Rosemann (2005) "Factors and Measures of Business Process Modelling: Model Building Through a Multiple Case Study", *European Journal of Information Systems* (14)4, pp. 347-360

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

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

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

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

Burton-Jones, A. and P. Meso (2006) "Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis", *Information Systems Research* (17)1, pp. 38-60

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
Cohen, J. (1960) "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement* (20)1, pp. 37-46
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 Two 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. (2005) "The Coming Commoditization of Processes", *Harvard Business Review* (83)6, pp. 100-108
Davies, I. et al. (2006) "How do Practitioners Use Conceptual Modeling in Practice?", *Data & Knowledge Engineering* (58)3, pp. 358-380
Dehnert, J. and A. Zimmermann (2005) "On the Suitability of Correctness Criteria for Business Process Models", *3rd International Conference on Business Process Management,* Nancy, France: Springer, pp. 386-391
Dreiling, A. et al. (2008) "From Conceptual Process Models to Running Systems: A Holistic Approach for the Configuration of Enterprise System Processes", *Decision Support Systems* (45)2, pp. 189-207
Dumas, M., W. M. P. van der Aalst, and A. H. M. ter Hofstede (2005) *Process Aware Information Systems: Bridging People and Software Through Process Technology*, Hoboken, New Jersey: John Wiley & Sons
Evermann, J. (2009) "A UML and OWL Description of Bunge's Upper-level Ontology Model", *Software and Systems Modeling* (8)2, pp. 235-249
Fettke, P. and P. Loos (2007) "Ontological Evaluation of Scheer's Reference Model for Production Planning and Control Systems", *Journal of Interoperability in Business Information Systems* (2)1, pp. 9-28
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
Garda, J. A. et al. (2004) "A Methodological Framework for Generic Conceptualisation: Problem-Sensitivity in Software Engineering", *Information and Software Technology* (46)10, pp. 635-649
Gartner Group (2009) "Meeting the Challenge: The 2009 CIO Agenda", *EXP Premier Report January2009*, Gartner, Inc, Stamford, Connecticut
Gemino, A. and Y. Wand (2005) "Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties", *Data & Knowledge Engineering* (55)3, pp. 301-326
Green, P. and M. Rosemann (2000) "Integrated Process Modeling. An Ontological Evaluation", *Information Systems* (25)2, pp. 73-87
Green, P. and M. Rosemann (2001) "Ontological Analysis of Integrated Process Models: Testing Hypotheses", *Australasian Journal of Information Systems* (9)1, pp. 30-38
Green, P. 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. et al. (2007) "Candidate Interoperability Standards: An Ontological Overlap Analysis", *Data & Knowledge Engineering* (62)2, pp. 274-291
Green, P. et al. (2006) "Improving Representational Analysis: An Example from the Enterprise Interoperability Domain", *17th Australasian Conference on Information Systems,* Adelaide, Australia: Australasian Association for Information Systems
Gregoriades, A. and A. G. Sutcliffe (2008) "A Socio-technical Approach to Business Process

Simulation", *Decision Support Systems* (45)4, pp. 1017-1030
Guizzardi, G. (2005) *Ontological Foundations for Structural Conceptual Models*, Enschede, The Netherlands: Telematica Instituut
Hansen, G. A. (1996) "Simulating Software Development Processes", *Computer* (29)1, pp. 73-77
Herbst, H. et al. (1994) “The Specification of Business Rules: A Comparison of Selected Methodologies”, in Verrijn-Stuart, A. A. and T. W. Olle (eds.) *Methods and Associated Tools for the Information Systems Life Cycle*, Amsterdam, The Netherlands: Elsevier, pp. 29-46
Hevner, A. R. et al. (2004) "Design Science in Information Systems Research", *MIS Quarterly* (28)1, pp. 75-105
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
Karimi, J. (1988) "Strategic Planning for Information Systems: Requirements and Information Engineering Methods", *Journal of Management Information Systems* (4)4, pp. 5-24
Keen, C. D. and C. Lakos (1994) “Information Systems Modelling using LOOPN++, an Object Petri Net Scheme”, *4th International Working Conference on Dynamic Modelling and Information Systems,* Noordwijkerhout, The Netherlands: Delft University Press, pp. 31-52
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: IEEE Computer Society, pp. 434-441
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
Kovacic, A. (2004) "Business Renovation: Business Rules (Still) the Missing Link", *Business Process Management Journal* (10)2, pp. 158-170
Mayer, R. J. 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
Moody, D. L. (2005) "Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions", *Data & Knowledge Engineering* (15)3, pp. 243-276
Moore, G. C. and I. Benbasat (1991) "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation", *Information Systems Research* (2)3, pp. 192-222
Mylopoulos, J. (1992) “Conceptual Modelling and Telos”, in Loucopoulos, P. and R. Zicari (eds.) *Conceptual Modelling, Databases, and CASE: an Integrated View of Information System Development*, New York, New York: John Wiley & Sons, pp. 49-68
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 (current March 12, 2005)
Opdahl, A. L. and B. Henderson-Sellers (2001) "Grounding the OML Metamodel in Ontology", *Journal of Systems and Software* (57)2, pp. 119-143
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
Ouyang, C. et al. (2009) "From Business Process Models to Process-Oriented Software Systems", *ACM Transactions on Software Engineering Methodology* (19), In Press
Parsons, J. and L. Cole (2004) “An Experimental Examination of Property Precedence in Conceptual Modelling”, *1st Asian-Pacific Conference on Conceptual Modelling,* Dunedin, New Zealand: Australian Computer Society, pp. 101-110
Petri, C. A. (1962) “Fundamentals of a Theory of Asynchronous Information Flow”, in Popplewell, C. M. (ed.) *IFIP Congress 62: Information Processing*, Munich, Germany: North-Holland, pp. 386-390
Phalp, K. T. (1998) "The CAP Framework for Business Process Modelling", *Information and Software Technology* (40)13, pp. 731-744
Raduescu, C. et al. (2006) “A Framework of Issues in Large Process Modeling Projects”, *14th European Conference on Information Systems,* Goeteborg, Sweden: Association for Information Systems, pp. 1594-1605
Recker, J. and A. Dreiling (2007) “Does It Matter Which Process Modelling Language We Teach or

Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education", *18th Australasian Conference on Information Systems,* Toowoomba, Australia: The University of Southern Queensland, pp. 356-366

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

Recker, J., M. Indulska, and P. Green (2007a) "Extending Representational Analysis: BPMN User and Developer Perspectives", in Alonso, G., P. Dadam, and M. Rosemann (eds.) *Business Process Management - BPM 2007*, Brisbane, Australia: Springer, pp. 384-399

Recker, J. et al. (2006) "How Good is BPMN Really? Insights from Theory and Practice", *14th European Conference on Information Systems,* Goeteborg, Sweden: Association for Information Systems, pp. 1582-1593

Recker, J., M. Rosemann, and J. Krogstie (2007b) "Ontology- versus Pattern-based Evaluation of Process Modeling Languages: A Comparison", *Communications of the Association for Information Systems* (20)48, pp. 774-799

Rohde, F. (1995) "An Ontological Evaluation of Jackson's System Development Model", *Australasian Journal of Information Systems* (2)2, pp. 77-87

Rosemann, M. (1998) "Managing the Complexity of Multiperspective Information Models using the Guidelines of Modelling", *3rd Australian Conference on Requirements Engineering,* Geelong: School of Management Information Systems, pp. 101-118

Rosemann, M. (2006) "Potential Pitfalls of Process Modeling: Part A", *Business Process Management Journal* (12)2, pp. 249-254

Rosemann, M. and P. Green (2000) "Integrating Multi-Perspective Views Into Ontological Analysis", *21st International Conference on Information Systems,* Brisbane, Australia: Association for Information Systems, 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 Lu, H. et al. (eds.) *Conceptual Modeling – ER 2004*, Shanghai, China: Springer, pp. 110-121

Rosemann, M. et al. (2009) "Using Ontology for the Representational Analysis of Process Modeling Techniques", *International Journal of Business Process Integration and Management* (4)1

Rosemann, M., J. Recker, and C. Flender (2008) "Contextualization of Business Processes", *International Journal of Business Process Integration and Management* (3)1, pp. 47-60

Rosemann, M. et al. (2006) "A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars", in Dubois, E. and K. Pohl (eds.) *Advanced Information Systems Engineering - CAiSE 2006*, Luxembourg, Grand-Duchy of Luxembourg: Springer, pp. 447-461

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

Sadiq, S., G. Governatori, and K. Niamiri (2007) "Modeling Control Objectives for Business Process Compliance", in Alonso, G., P. Dadam, and M. Rosemann (eds.) *Business Process Management - BPM 2007*, Brisbane, Australia: Springer, pp. 149-164

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

Shanks, G. et al. (2008) "Representing Part–Whole Relations in Conceptual Modeling: An Empirical Evaluation", *MIS Quarterly* (32)3, pp. 553-573

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

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

Soffer, P. and Y. Wand (2007) "Goal-Driven Multi-Process Analysis", *Journal of the Association for Information Systems* (8)3, pp. 175-202

Soffer, P., Y. Wand, and M. Kaner (2007) "Semantic Analysis of Flow Patterns in Business Process Modelling", in Alonso, G., P. Dadam, and M. Rosemann (eds.) *Business Process Management - BPM 2007*, Brisbane, Australia: Springer, pp. 400-407

Tardieu, H. (1992) "Issues for Dynamic Modelling through Recent Development in European Methods", in Sol, H. G. and R. L. Crosslin (eds.) *Dynamic Modelling of Information Systems*

II, Amsterdam, The Netherlands: North-Holland, pp. 3-23
van der Aalst, W. M. P. (1999) "Formalization and Verification of Event-driven Process Chains", *Information and Software Technology* (41)10, pp. 639-650
van der Aalst, W. M. P. (2003) "Don't Go with the Flow: Web Services Composition Standards Exposed", *IEEE Intelligent Systems* (18)1, pp. 72-76
van der Aalst, W. M. P. and A. H. M. ter Hofstede (2005) "YAWL: Yet Another Workflow Language", *Information Systems* (30)4, pp. 245-275
van Griethuysen, J. J. (1982) "Concepts and Terminology for the Conceptual Schema and the Information Base", *ISO/TC97/SC5 Report N695*, International Organization for Standardization, Geneva, Italy
Verbeek, H. M. V., W. M. P. van der Aalst, and A. H. M. ter Hofstede (2007) "Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Relaxed Soundness and Invariants", *The Computer Journal* (50)3, pp. 294-314
Vessey, I. and S. A. Conger (1994) "Requirements Specification: Learning Object, Process, and Data Methodologies", *Communications of the ACM* (37)5, pp. 102-113
Wand, Y. and R. Weber (1990) "An Ontological Model of an Information System", *IEEE Transactions on Software Engineering* (16)11, pp. 1282-1292
Wand, Y. and R. Weber (1993) "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars", *Journal of Information Systems* (3)4, pp. 217-237
Wand, Y. and R. Weber (1995) "On the Deep Structure of Information Systems", *Information Systems Journal* (5)3, pp. 203-223
Wand, Y. and R. Weber (2002) "Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda", *Information Systems Research* (13)4, pp. 363-376
Wand, Y. and R. Weber (2006) "On Ontological Foundations of Conceptual Modeling: A Response to Wyssusek", *Scandinavian Journal of Information Systems* (18)1, pp. 127-138
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
Wyssusek, B. (2006) "On Ontological Foundations of Conceptual Modelling", *Scandinavian Journal of Information Systems* (18)1, pp. 63-80
Zhang, H., R. Kishore, and R. Ramesh (2007) "Semantics of the MibML Conceptual Modeling Grammar: An Ontological Analysis Using the Bunge-Wand-Weber Framework", *Journal of Database Management* (18)1, pp. 1-19
zur Muehlen, M. and M. Indulska (2009) "Modeling Languages for Business Processes and Business Rules: A Representational Analysis", *Information Systems* (In Press)
zur Muehlen, M. and J. Recker (2008) “How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation”, in Léonard, M. and Z. Bellahsène (eds.) *Advanced Information Systems Engineering - CAiSE 2008*, Montpellier, France: Springer, pp. 465-479

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 뱡eight?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)

About the Authors

Dr. Jan Recker is Senior Lecturer at the Information Systems Program at Queensland University of Technology Brisbane, Australia. He received a BScIS and MScIS from the University of Muenster, Germany, and a PhD in Information Systems from Queensland University of Technology. His research interests include BPM Standards, User-centered Systems Analysis and Design, Process Flexibility and Post-Adoptive Usage. Findings from his research have been published in journals such as *Information Systems*, the *Communications of the Association for Information Systems*, the *Australasian Journal of Information Systems*, the *Business Process Management Journal*, and others.

Dr. Michael Rosemann is full Professor and co-leader of the Business Process Management Group at the Queensland University of Technology, Brisbane, Australia. Dr Rosemann's research interests are Business Process Management, Conceptual Modeling and Enterprise Systems. He is the Chief investigator of a number of research projects funded by the Australian Research Council and industry partners. Dr Rosemann's publications have appeared in journals such as *MIS Quarterly*, *European Journal of Information Systems*, *Decision Support Systems*, *IEEE Transactions on Knowledge and data Engineering*, and *Information System*. He is the author and editor of six books and has been the General Chair of the International Business Process Management Conference in 2007.

Dr. Marta Indulska is Senior Lecturer at the UQ Business School, The University of Queensland. She obtained her PhD in Computer Science at the University of Queensland in 2004. Marta's main research interests lie in the domains of conceptual modeling, Business Process Management, and Compliance Management. In particular, her current research interests focus on the representation of business rules and the analysis and improvement of process modeling techniques. Her work has appeared in journals such as *IEEE Transactions on Knowledge & Data Engineering*, I*nformation Systems*, *Decision Support Systems*, and *Data & Knowledge Engineering*.

Dr. Peter Green is Professor of Electronic Commerce and Business Information Systems cluster leader in the UQ Business School at the University of Queensland. He has qualifications in Computer Science, Accounting, and a PhD in Commerce (Information Systems) from the University of Queensland. He has worked during his career as the Systems Support Manager at the South-East Queensland Electricity Board (SEQEB), for a Chartered Accountancy firm, and a Queensland government department. He has researched, presented, and published widely on systems analysis and design, conceptual modelling, information systems auditing, and eCommerce. His publications have appeared in such internationally refereed journals as *Information Systems*, *IEEE Transactions on Knowledge & Data Engineering, Data & Knowledge Engineering, Journal of Database Management*, and the *Australian Journal of Information Systems*.

Journal of the Association for Information Systems

ISSN: 1536-9323