Information Systems 35 (2010) 379–390

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Modeling languages for business processes and business rules: A representational analysis

Michael zur Muehlen [a], Marta Indulska [b,*]

[a] Howe School of Technology Management, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, United States
[b] UQ Business School, The University of Queensland, St. Lucia, QLD 4072, Australia

ARTICLE INFO

Keywords:
Business process modeling
Business rule modeling
Business process management
Representation theory
BWW
BPMN
SBVR
PRR
SRML
SWRL

ABSTRACT

Process modeling and rule modeling languages are both used to document organizational policies and procedures. To date, their synergies and overlap are under-researched. Understanding the relationship between the two modeling types would allow organizations to maximize synergies, avoid content duplication, and thus reduce their overall modeling effort. In this paper, we use the Bunge–Wand–Weber (BWW) representation theory to compare the representation capabilities of process and rule modeling languages. We perform a representational analysis of four rule modeling specifications: The Simple Rule Markup Language (SRML), the Semantic Web Rules Language (SWRL), the Production Rule Representation (PRR), and the Semantics of Business Vocabulary and Business Rules (SBVR) specification. We compare their BWW representation capabilities with those of four popular conceptual process modeling languages. In our analysis, we focus on the aspects of maximum ontological completeness and minimum ontological overlap. The outcome of this study shows that no single language is internally complete with respect to the BWW representation model. We also show that a combination of two languages, in particular SRML and BPMN, appears to be better suited for combined process and rule modeling than any of these modeling languages used independently.

1. Introduction

The improvement of corporate processes has consistently been identified as a top priority of CIOs for the last few years [5]. Organizations are increasingly interested in understanding, managing and improving their process portfolio, and in identifying and quantifying processes with outsourcing potential. The collection of tools and methods to achieve these objectives is referred to as Business Process Management (BPM). The design of innovative processes is constrained by the rules and regulations an organization has to comply with. Business Rules Management (BRM) describes the identification, definition, and management of these rules using technology such as Business Rules Management Systems.

Both process modeling languages and rule modeling languages offer constructs to represent business operations and constraints, but they do so in different ways. While process modeling languages typically describe a procedural sequence of activities, including decisions and concurrency, rule modeling languages often rely on a declarative description of facts, conditions, and constraints. This situation presents a selection dilemma for organizations, and little guidance exists as to which modeling approach is preferable in a particular situation. Despite a significant focus on the evaluation of the representational capability of process modeling languages

* Corresponding author. Tel.: +61733811413.
E-mail addresses: mzurmuehlen@stevens.edu (M. zur Muehlen), m.indulska@business.uq.edu.au (M. Indulska).

0306-4379/$ - see front matter
doi:10.1016/j.is.2009.02.006

[29], the comparative evaluation of rule modeling languages has received considerably less attention. This situation is a concern because an increasing number of organizations are deploying both BPM and BRM solutions (see [30]). In addition, a number of – partially overlapping – efforts are underway to specify standard representations for business rules. Comparing and contrasting these approaches will help organizations select the most appropriate representation for their purposes.

Recent empirical research has identified representational weaknesses in process modeling languages [27]. This research has led to speculation that business rule modeling languages might be suitable to fill these weak spots. It is an open question whether the two language types should be used in combination, i.e. whether the integrated use of business rules and business process modeling languages allows organizations to better understand, represent, and improve their operations. Accordingly, there is a need for a rigorous analysis of the two types of languages in order to identify their potential synergies and conflicts.

The main *goal* of the work we present in this paper is to investigate the representation capability of four rule modeling specifications. The four specifications selected for analysis are the Simple Rule Markup Language (SRML) [34], the Semantic Web Rules Language (SWRL) [10], the Production Rule Representation (PRR) [23], and the Semantics of Business Vocabulary and Business Rules (SBVR) specification [22]. We place this evaluation in the context of previous evaluations of conceptual process modeling languages by using the same evaluation framework and measurement techniques. Our evaluation is based on the well-established Bunge–Wand–Weber (BWW) representation theory [38], which allows us to gauge the degree to which each specification is capable of representing fundamental elements of the real world. In line with these goals, our two *research questions* are as follows:

RQ1: What are the representational capabilities, with respect to the BWW representation theory, of SRML, SWRL, PRR, and SBVR?

RQ2: Are the representational capabilities of SRML, SWRL, PRR, and SBVR complementary or substitutive to those of process modeling languages?

To answer the first question we will map the elements of the four business rule specifications against the constructs of the BWW representation model. To answer the second question we will compare these mappings to the BWW mappings of process modeling languages. The degree to which the mappings overlap and/or diverge will allow us to infer representational capabilities of each specification independently, and in combination with other specifications.

The remainder of this paper is structured as follows. In the next section, we present a brief review of business rules and business processes. The section also provides a review of related work on the integration of the two approaches and discusses studies that identify representational deficiencies in process modeling languages. Section 3 presents the justification for the use of the BWW representation theory as a suitable benchmark for the analysis of representation capabilities of process and rule modeling languages. Section 4 describes the research methodology adopted in this work and provides a justification for the selection of languages under consideration. In Section 5, we present a summary of the results of the BWW-based representation analysis of SRML, SWRL, PRR, and SBVR and discuss the results of the analysis in light of combined representation capabilities of process and rule modeling languages. We conclude the paper in Section 6 with a discussion of limitations and future work directions in this area.

2. Background

While, to the best of our knowledge, no representational evaluation of rule modeling languages has been carried out, some attempts at the integration of rule- and process-based modeling approaches have been made, different approaches to specify business rules have been surveyed in the existing literature, and the strengths and weaknesses of process modeling languages have been explored in some studies. These works inform our research.

2.1. Business rules

A business rule is a statement that aims to influence or guide behavior and information in an organization [33]. According to their structure, different types of business rules can be distinguished [37]:

- *Integrity rules* express constraints. These rules typically define the acceptable relationship between data elements. For example, each project must have one and only one project manager.
- *Derivation rules* express conditions that result in conclusions. These rules define the validity of facts and can be used to infer new facts based on known facts. For example, platinum customers receive a 5% discount. John Doe is a platinum customer. As a conclusion, John Doe receives a 5% discount.
- *Reaction rules* (also known as Event-Condition-Action (ECA) rules, alternative-action rules, or post-conditions) specify a trigger that activates the evaluation of the rule, a condition that is evaluated, and a subsequent activity that will be carried out if the specified condition is met; for example, the evaluation of a reaction rule is triggered as soon as a new invoice is received. If the invoice amount is more than $1000 then a supervisor review is initiated.
- *Production rules* (also known as condition, action rules) are similar to reaction rules, but do not specify a particular circumstance in which the evaluation takes place; For example, if there are no defects in the last 10 widgets, the entire batch is quality approved.
- *Transformation rules* restrict the state changes of objects; for example, an employee's age can change from 30 to 31, but not from 31 to 30.

Business rule modeling languages are typically based on formal logic and have strong and precise expressive power [20]. In general, they belong to the declarative modeling category in that they focus on specifying *what* is required to take place, rather than *how* something is accomplished.

2.2. Business processes

Business processes are logically ordered sets of activities that produce a result of value to the customer. The modeling, execution (including automation), and evaluation of processes is known as Business Process Management. Process models are constructed using modeling grammars or languages (also referred to as techniques or notations). Most of these languages represent processes as procedural models, in that they focus on specifying the step-by-step activities that are required to take place in order to perform an action. Some authors further distinguish between process representations as scripts (e.g. tightly prescribed) and maps (e.g. loosely specified) [32]. Process modeling languages provide a set of primitives (modeling constructs) and a set of construction rules that governs the use and permissible combination of the primitives.

Process modeling languages or grammars can be classified according to their focal modeling construct:

- *Activity centered* process modeling languages represent processes as a network of tasks or activities that are linked through control- or data-flow connectors.
- *Process object centered* approaches specify processes as the permissible sequence of state changes of the process object.
- *Resource centered* process modeling languages represent processes as networks of processing stations that interact with each other.

Process languages appear as graph-based languages (e.g. BPMN, EPC), net-based languages (e.g. Petri nets, flow nets), and workflow programming languages (e.g. Business Process Execution Language (BPEL)).

2.3. Integration of business rules and business processes

Early work on the integration of business rules and business processes appeared shortly after the introduction of the rule modeling concept [13,14]. Krogstie et al. [16] were the first to suggest that business process and rule modeling approaches should be merged to improve the capture of temporal information for Information Systems (IS) development. They presented a top-down approach for model specification that involves the use of the External Rule Language for specification of process logic at the lowest level of decomposition. McBrien and Seltveit [20] further enhanced this concept by defining the structure of rules within the process model. Knolmayer et al. [14] refined process modeling and linked the resulting models to workflow execution through layers of Reaction Business Rules. Kappel et al. [13] use Reaction Business Rules to model the coordination in workflow systems. Kovacic [15] developed a meta-model that represents important business constructs (goal, process, activity and events) and technical constructs (data objects, software components, actions in Information Systems). He demonstrates how rules can link these two categories of constructs. Charfi and Mezini [1] argues that business rules are often hard-coded into web services and proposes a hybrid approach of separating business processes and business rules. Meng et al. [21] introduced a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes. The system uses an event- and rule-server to trigger business rules during the enactment of workflow processes in order to enforce business constraints and policies.

While the integration of rule and process modeling has been the subject of some investigation in the research community, anecdotal evidence shows that organizations struggle to effectively capture business processes and rules. In a recent study of the representational capabilities of the Business Process Modeling Notation (BPMN), we found that organizations frequently supplement their BPMN process models with textual annotations of business rules [27]. This practice introduces problems regarding the consistency, reuse, and enforcement of rules – problems that are acknowledged by some of the organizations using this technique.

The need to improve the representation of business rules within process model diagrams is apparent, yet little is known about which representation aspects, if any, are unique to each of the two types of modeling languages. Previous work by Recker et al. [26] has identified a general lack among process modeling languages to adequately represent business rules. Similarly, Green and Rosemann [6] found limitations with respect to modeling business rules in their BWW-based investigation of all five views of Architecture of Integrated Information Systems a popular enterprise architecture framework.

Rule modeling languages are likely candidates to fill such gaps. An earlier study by Herbst et al. [9] suggests that rule specification languages should be considered as a potential addition to graphical representation languages when modeling for Information Systems design. While their analysis is not based on any formal framework, they suggest that many of the popular IS modeling techniques lack the ability to adequately represent business rules. The work of Rosemann et al. [29] suggests that the same shortcomings exist in the process modeling domain; hence, an integration of business rule and business process modeling approaches may help overcome these perceived shortcomings.

In order to effectively integrate graphical business process modeling approaches with business rule modeling approaches, we need to understand their synergies and overlap. In our research, we were unable to locate any attempts to evaluate the expressiveness of rule modeling languages or their relationships to conceptual process modeling approaches. The only related work appears to be that of Lu and Sadiq [17] who compared graph-based and rule-based modeling approaches. Since their work was focused on workflow modeling in particular, rather than conceptual modeling in general, no specific rule modeling languages were considered. The authors used a set of workflow patterns [36] as a basis for the evaluation, and found, that rule- and graph-based modeling approaches had similar levels of expressiveness in terms of the control flows specified by the workflow patterns.

In summary, while some work has been performed to integrate process modeling and rule modeling approaches, no theory-based evaluation of the usefulness of these combinations has been conducted so far. There is a need to augment existing research on the representational capabilities of process modeling languages with matching evaluations on the rule modeling side. Furthermore, there is a need to provide practitioners with guidance as to which rule modeling language and which combination of rule and process modeling languages will allow them to capture the most real-world details using language primitives.

3. Representation theory

An ontology, or representation theory, can be used as a benchmark to make predictions about the capabilities of a grammar to provide complete and clear representations of a real-world domain [39]. The application of an ontology for such a purpose is known as *representational analysis*.

Representational analysis is performed by comparing the constructs of the chosen representation theory with the constructs of the modeling grammar and by identifying any representation equivalence between these. Any deviation from a one-to-one mapping relationship between these constructs indicates potential representational deficiencies in the grammar. Two principal evaluation criteria are *ontological completeness*, i.e. the extent to which the modeling grammar has a deficit of constructs that map to the set of representation theory constructs, and *ontological clarity*, i.e. the extent to which the modeling grammar constructs are deemed overloaded, redundant, or excessive [28]. These criteria provide a theoretical basis on which conceptual modeling languages can be compared with regard to their completeness and clarity of representation.

In this study, we use the Bunge–Wand–Weber ontology [39], specifically the representation model, since it is understood to contain all necessary constructs to describe things, and the interaction between things, in the real world. The BWW model is not without criticisms. [40,41] are notable examples relating to lack of empirical testing and to a realist view of the world, for instance. Other ontologies could also be applied as well, for example Chisholm's ontology [2] or the Enterprise Ontology [35]. However, our use of the BWW representation model is motivated in three ways: first, the model was developed specifically for the IS domain, has a formal specification (which helps ensure in-depth understanding of the model) and an established track record in the process modeling domain. Second, over the last two decades the model has achieved a good level of maturity, adoption and dissemination, allowing us to consider existing BWW analyses of process modeling languages. Third, results of many BWW-based studies were further tested empirically and the results of these tests indicate that the model is a good basis to study the representational capabilities of conceptual modeling languages (for example, see [6,26,27] among others). In the end, it is such empirical evidence that shows that the representation theory is useful as a representational analysis benchmark in the IS domain. In other words, if the representation theory is able to accurately predict issues with the modeling language in practice, then we consider it appropriate for the investigation of the representational capabilities of process and rule modeling language. Perhaps the strongest empirical evidence of BWW suitability to date comes from a recent study of the limitations of BPMN [25,27]. BWW was used in the study as the benchmark for analysis of BPMN representational capabilities. A series of interviews with BPMN modelers was conducted to test the propositions derived from the BWW BPMN analysis [27]. The study found support (in varying degrees) for 75% of the BWW-based propositions. Furthermore, as part of a larger study, Recker [25] hypothesized that ontological completeness impacts the modeler's perception of the usefulness of the grammar, which in turn impacts their intention to continue using the grammar. The hypotheses were tested via a survey, which attracted 590 full responses from BPMN modelers worldwide. The study found that BPMN's lack of representation for business rules had a strong and negative effect on its perceived usefulness ($p<0.01$). The study also found that the lack of representation for process structure and decomposition also had a significant negative effect of perceived usefulness on BPMN ($p<0.05$). It is based on results like these that we consider the BWW model to be a fruitful basis on which process and rule modeling languages can be compared.

The model consists of some 40 higher-level abstract constructs, which can be grouped into four categories: things and their properties, states of a thing, events and transformations, and systems and their composition. If a process or rule modeling language construct is found to have a representation for each of the BWW representation model constructs then that language fulfills the representation requirements criteria necessary to model all things and their interactions in the real world (with respect to the BWW representation model), without limiting the user's representation capabilities. While this may be the case, the language may still suffer from lack of clarity (e.g. an overload of constructs), which impacts its usability. Accordingly, a language that is complete and has the lowest levels of construct overload, redundancy and excess should be chosen. In this work, however, we focus only on the ontological completeness of a grammar. This focus is appropriate at this stage of the study since we are interested to find out the capacity of various rule modeling specifications to represent real-world concepts (i.e. their provision of representation of BWW constructs), rather than the clarity with which they are able to represent them.

When no one language provides the required representation capability in terms of completeness with respect to the BWW ontology, Green et al. [8] show that users will make use of combinations of languages that allow them to obtain maximum representation capability. Green et al. [7] discuss two theories for selecting two or more grammars for Information Systems modeling. The first, maximum ontological completeness (MOC), states that users will select combinations of languages that, together, afford them the maximum possible representation power for their domain, i.e. if more constructs from an underlying ontology are incorporated in the chosen grammar, the expressive power of the resulting language combination will be higher. The second, minimum ontological overlap (MOO), states that, when selecting languages to satisfy MOC, users will prefer languages with minimum overlap in the representation of ontological constructs, i.e. language combinations where no more than one grammatical construct maps to one construct within the chosen ontology (in this case, BWW). Higher levels of construct overlap will create confusion and conflict in the work of the users. Together, the application of the MOC and MOO theories is known as *overlap analysis* [7]. Overlap analysis [7], hence, specifically focuses on identifying representational overlaps between languages mapped to the same ontology by identifying ontological constructs that have mappings from more than one language. Overlap analysis is performed only when full ontological completeness cannot be achieved with a single language and is done with the intent to identify language combinations with the highest expressive power but low construct overlap.

We make use of the BWW representation model, together with the conduct of overlap analysis, in order to analyze the representational capabilities of SRML, SWRL, PRR, and SBVR, and identify combinations of business rule and business process modeling languages that are likely to be used in combination according to the BWW representation model.

4. Methodology

4.1. Selection and analysis of rule modeling languages

A variety of formats for the representation of business rules have been developed over the past decade, as shown in Fig. 1. These languages vary between research prototypes (e.g. N3), vendor-specific formats (e.g. Fair Isaac's SRL or ILOG's IRL), and proposals for the XML-based exchange of business rules (e.g. SRML, PRR, and SBVR). The variety of approaches is caused by the variety in rules philosophies. For instance, production rules require a different representation from rules used for reasoning applications. Another differentiating factor is the intended domain of use. For example, rules designed for governing business operations and human behavior may distinguish between permissible, desirable, and illegal states of an organization or process, while rules designed for the semantic web may focus on generating new facts from a set of known facts.

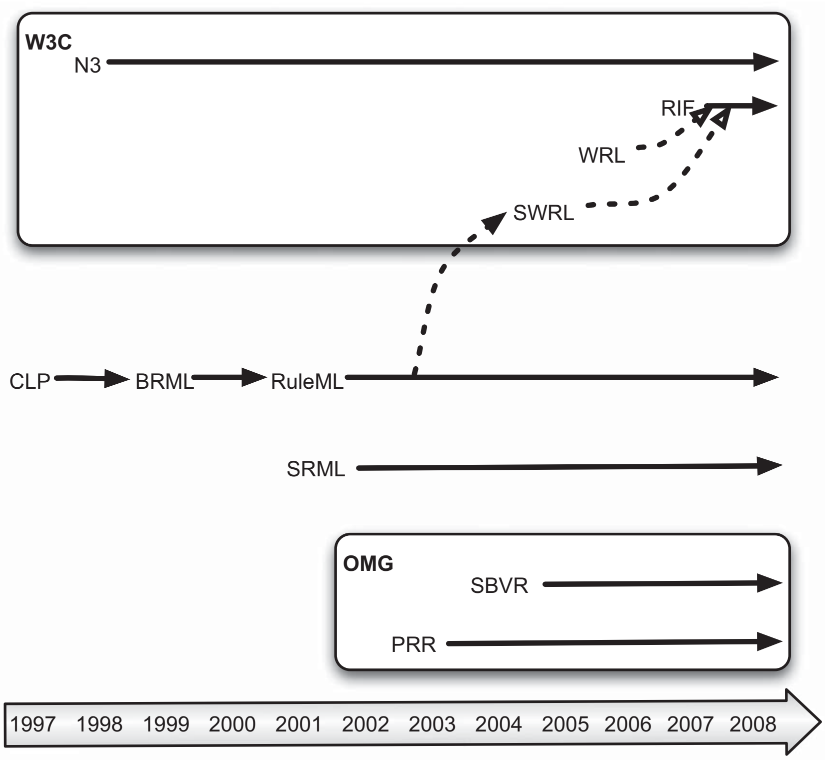

Fig. 1. Development of rule languages.

For our initial study, we selected four rule modeling specifications that provided a comprehensive explanation of their vocabularies. We chose the Simple Rule Markup Language as a representative example of a rule modeling language with a small vocabulary. A clear definition of its constructs is available and is not based on any other vocabulary [34]. We selected the Semantics of Business Vocabulary and Business Rules specification since it represents an attempt at the definition of a standardized rule modeling vocabulary [22]. SBVR presents a vocabulary that is intended to become a standard upon which many grammars can be based. For this reason, its inclusion in the ontological analysis is useful. We chose the Semantic Web Rule Language [10] as a representative grammar of rules in the specific application domain of the semantic web, and because it is one of the contributing specifications to the emerging Rule Interchange Format standard that a working group within the World Wide Web Consortium is developing. Finally, we selected the Production Rules Representation standard [23] because it was developed by the same organization as SBVR but with the intent to serve as an interchange format for commercial Production Rule Management systems, covering a spectrum of business rules that hierarchically lies below the intended use of SBVR.

In order to reduce the subjectivity and to improve the internal validity of our research, we employed the extended representational analysis methodology as suggested by Rosemann et al. [28]. We followed the reference methodology as closely as possible. In particular, to increase objective comparison, an Entity-Relationship Diagram representation of the BWW meta-model was obtained from the authors of [28] and an Entity-Relationship Diagram representation of the of the SRML meta-model was created to guide the mapping between SRML and BWW. The developed meta-model is shown in Fig. 2. The use of the meta-model was also supplemented with various examples of SRML use so as to aid in the understanding of the SRML constructs.

SBVR was not transformed into an Entity-Relationship Diagram because the SBVR specification contains the SBVR meta-model in form of UML class diagrams, which were sufficient to obtain a thorough understanding of the language. The analyses of SWRL and PRR were carried out based on their textual specifications and supplementary presentations and documents that were obtained from the web sites of the respective working groups.

Three researchers independently conducted the BWW representation analysis of the languages. The independent analyses were followed by coordination sessions, during which consensus was gained about the construct mappings. The analyses and respective consensus development meetings were conducted in turn for each language. First, SRML was analyzed and the analysis was finalized, then SBVR, followed by PRR and, finally, SWRL. Once the analyses were finalized, two researchers independently performed the overlap analysis in order to reduce the potential for error.

4.2. Selection of process modeling languages

The selection and analysis of conceptual process modeling languages was based on existing work. Many BWW-based representational analyses of process modeling languages have already been published, and some of them empirically tested (for an overview please refer to [26,27]). We specifically focus on conceptual modeling languages rather than executable languages such as Business Process Execution Language, since we concentrate on the documentation of policies and processes rather than their execution. Accordingly, we chose Petri nets in their colored Petri nets extension (CPN) [11], Event-driven Process Chains (EPC) [31], the Integrated

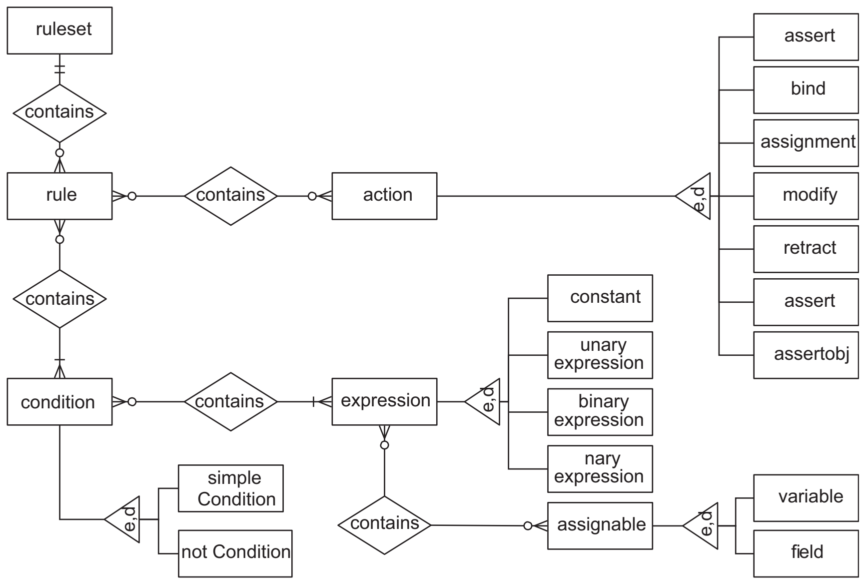

Fig. 2. SRML meta-model [e,d = exhaustive, disjoint specialization].

DEFinition methodology – Process Description Capture Method ($IDEF_3$) [19], and the Business Process Modeling Notation [24] as the basis of our comparison. We want to determine if business rule languages can indeed contribute to these popular graphical process modeling languages, as was speculated by Herbst et al. [9].

4.3. Overlap analysis procedure

The analysis of how business rule and process modeling languages complement each other was performed based on the results shown in Table 1. As a first step, we compare the representational capabilities of the eight languages mapped to the BWW ontology, and derive a set of BWW representation model constructs that *do not* have a corresponding construct across the chosen languages. This situation implies that full completeness of representation cannot be achieved with the selected languages with respect to the BWW representation model.

As a second step, we apply the earlier discussed process of overlap analysis [7] in order to determine a pair of languages that provides, with respect to the BWW representation model, the highest representation modeling power while having the lowest amount of construct overlap between the languages. Since the MOC theory takes precedence in such analysis, we focus on all language pairs that provide MOC and select the pair(s) that have the lowest construct overlap (MOO) with respect to the BWW representation model. While this is a simple measure of overlap of representational capability of languages, and has limitations in terms of lack of consideration of the granularity of representation, it is currently the only method applicable in a situation where representational capability of various modeling notations, with no consistent formal specification, is analyzed.

There is a significant body of research focusing on the overlap of ontologies (see, for example, [4,12,18]), in particular when they are specified in OWL and have formal concept hierarchies that can be used to measure

Table 1
BWW analysis results (including material from [8,26]).

Language Year/version/BWW constructs	SRML 2001	SBVR 2006	PRR 2007 v1.0	SWRL 2006 v0.6	CPN 1981	EPC 1992	IDEF3 1995	BPMN 2004 v1.0
BWW Constructs								
Things								
Thing	✓	✓		✓	✓		✓	✓
Property	✓	✓	✓	✓	✓	✓	✓	✓
Class		✓	✓	✓	✓			✓
Kind								✓
States								
State	✓				✓	✓	✓	
Conceivable state space	✓							
State law	✓	✓	✓	✓	✓	✓		
Lawful state space	✓		✓		✓			
Stable state						✓		
Unstable state					✓			
History								
Events								
Event	✓				✓	✓	✓	✓
Conceivable event space								
Lawful event space								
External event						✓		✓
Internal event					✓	✓		✓
Well-defined event					✓	✓		✓
Poorly defined event								✓
Transformation	✓			✓	✓	✓	✓	✓
Lawful transformation	✓		✓	✓	✓	✓		✓
Coupling							✓	✓
Acts on					✓			✓
Systems								
System	✓	✓	✓				✓	✓
System Composition		✓	✓				✓	✓
System environment								✓
System structure							✓	
System decomposition		✓					✓	✓
Level structure						✓	✓	✓
Sub-system								✓
# BWW constructs repres.	10	7	7	6	13	11	11	19

the distance between concepts. There is also related work on concept similarity (see, for example, [3]), which is generally used in analysis and merging of ontologies. However, many conceptual modeling languages such as Flowcharts, BPMN, or event-driven Process Chains are not designed on the basis of a formal meta-model (as opposed to, for example, UML). The interpretation of the natural language specifications of these languages requires a significant degree of subjective interpretation to classify each modeling construct. The application of mediational analysis and concept similarity would require a formal classification of each modeling language element in a modeling ontology. For instance, an activity in BPMN and the action component of the ECA rule both map to the BWW construct Transformation. Since Transformation is a high-level construct it may be specialized in practice. A mapping of activity and action to subtypes of Transformation might reveal that they map to different subtypes, and that these subtypes are separated by several layers of disjoint subtypes – rendering them relatively dissimilar [3]. Since the authors could not locate an agreed-upon subtyping of BWW constructs we continue with the application of overlap analysis as discussed in [7]. The results of our analysis should be taken as a starting point for further investigation into the practical integration of two modeling languages.

5. Results

The summary of the BWW representational analysis of SRML, SWRL, PRR, and SBVR is shown in Table 1. A tick indicates that the rule modeling language was found to have capability to represent the corresponding BWW representation model construct. The table also indicates the results of the analyses of the four considered process modeling languages, including our extension of the traditional Petri net analysis to the analysis of the colored extension. The full details of the mapping reasoning are omitted due to paper length limitations.

The analysis shows that the four chosen rule modeling languages are less expressive than their process modeling counterparts, with respect to the BWW representation model. This is perhaps not surprising given that their focus is narrower that that of graphical process modeling languages. For example, while we would expect business rules modeling languages to provide a corresponding representation for a BWW construct such as *state law*, the lack of graphical representation might lessen the need for a representation of *system decomposition* or *system structure*. However, this observation implies that business rule modeling languages like SRML, SWRL, PRR, or SBVR in isolation do not provide an equivalent or better means for modeling processes than some of the established and popular process modeling languages (most notably, BPMN).

5.1. BWW constructs lacking in process and rule modeling

Closer investigation of Table 1 shows that, even when used in combination, none of the popular process and rule modeling languages considered provides complete coverage for all BWW constructs. In particular, representations for the BWW representation theory constructs of *history*, *conceivable event space*, and *lawful event space* are missing across all languages under consideration. This implies that even combinations of conceptual process modeling languages and business rule modeling languages are neither able to represent audit trails of activities nor the sets of all the possible or allowed events that can occur in a given situation. More specifically, while the addition of business rule modeling languages appears to alleviate some of the empirically validated weaknesses of the popular BPMN language in particular [27], we still expect the following shortcomings to manifest in practice:

P1: The lack of corresponding representation for the BWW construct of *history* may have an impact where a log of an entity's state changes is required. The lack of such explicit representation can, for example, impact exception modeling, in particular recovery.

P2: The lack of corresponding representation for the BWW construct of *conceivable event space* implies an inability to model the set of all possible events that can occur within a process. While this representation capability may not always be required in the process modeling domain, the lack of it increases the complexity of identifying events of interest to the process being modeled and events that are not allowed to have a triggering impact.

P3: The lack of corresponding representation for the BWW construct of *lawful event space*, similarly to proposition P2, has a negative impact on the modeling of allowable events in a process model. Specifically, there is no modeling construct that would allow for the representation of all events that are legal in a given process context, thus impacting exception handling modeling.

5.2. Overlap analysis results

The overlap analysis of the eight chosen languages is summarized in Table 2. Each cell in the table is a quadrant indicating the symmetric difference, relative comple-

Table 2
Overlap analysis of the selected languages.

R \ P	CPN		EPC		IDEF$_3$		BPMN	
SRML	7	5	9	5	9	5	17	13
	8	2	6	4	6	4	6	4
SBVR	12	9	14	9	8	6	14	13
	4	3	2	5	5	2	6	1
PRR	10	8	12	8	12	8	16	14
	5	2	3	4	3	4	5	2
SWRL	7	7	9	7	11	8	15	14
	6	0	4	2	3	3	5	1

Legend for each quadrant.
1. $P \triangle R$
2. $P \cap R$
3. $P \setminus R$
4. $R \setminus P$

ments and intersection of the set of BWW construct representations provided in the process modeling specification (P) and the set of BWW construct representations provided in the rule modeling specification (R):

1. *Symmetric difference*: $P\Delta R=(P-R)\bigcup(R-P)$. That is, the number of BWW constructs represented distinctly by the given combination of languages, free of overlap.
2. *Intersection*: $P\bigcap R=[x\in P|x\in R]$. That is, the number of BWW concepts that can additionally be represented by both languages, with construct overlap.
3. *Relative complement of R in P*: $P\backslash R=[x\in P|x\notin R]$. That is, the number of non-overlapping BWW constructs contributed by the process modeling language shown in Table 2 column to the rule modeling language shown in Table 2 row.
4. *Relative complement of P in R*: $R\backslash P=[x\in R|x\notin P]$ That is, the number of non-overlapping BWW constructs contributed by the rule modeling language shown in Table 2 row to the process modeling language shown in Table 2 column.

While the overlap analysis shows synergies between rule modeling languages and process modeling languages, these do not appear to be as dramatic as we expected. It is clear that business process modeling languages are a better choice for modeling organizational procedures, and that BPMN is a construct-rich process modeling language that could be enriched by the addition of SRML for this domain.

Investigation of the overlap results summarized in Table 2 indicates that incorporating the use of SRML with any of the four popular conceptual process modeling languages allows users the ability to represent between fifteen and 23 representation theory constructs (i.e. MOC = 17+6 = 23). Minimal ontological overlap is equal to six, implying that whichever combination of languages is chosen, a minimum of six (and maximum of eight) constructs will be overlapping in the language pair. Considering both MOC and MOO theories, the analysis clearly shows that the combination of BPMN and SRML provides users with the highest BWW representation power, while having minimal ontological overlap. SRML complements BPMN by providing constructs that express state, state laws, conceivable and legal state space, while providing potential integration points through the shared concepts thing, event and property. From an operational process engineering perspective this allows for a tight integration of SRML with individual BPMN models through the use of the BWW event construct. In SRML, the event construct maps to the action part of a rule, i.e. the firing of a rule. In BPMN, the event construct maps to a variety of events that a process can either raise or react to. The latter variant allows a process to react to the firing of a rule. In practice, this means that the execution of a BPMN process can be governed through rules expressed in SRML, if the firing of these rules is represented via event listeners in the BPMN model.

In contrast, the analysis also shows that the combination of SRML and CPN is not a good option, given the higher level of overlap and representation of only fifteen representation theory constructs (as compared with 23 from the BPMN/SRML combination).

A closer examination of the overlap analysis of PRR shows that the best achievable result is a combination of PRR and BPMN. Such a combination of languages provides representation for 21 BWW constructs at the cost of five overlapping constructs (i.e. MOC = 21, MOO = 5). While PRR provides constructs that map to state law and lawful state space, its integration points with BPMN include system, system composition, and lawful transformation. In BPMN, lawful transformation maps to the sequence flow constructs used to define the control flow of a process. In PRR, this construct maps to the action part of a rule, i.e. PRR rules can be used to govern the control flow behavior of a process. However, from a representation theory perspective the combination of PRR and BPMN is still inferior to combining SRML and BPMN (MOC = 23). Furthermore, the combination of PRR with any of the remaining three process modeling languages under consideration results in a MOC of fifteen – hence results in a significant drop in representation capability but with a lower overlap of constructs.

A closer look at SBVR reveals that it provides representation of three additional representation theory constructs at the same level of construct overlap compared with SRML. The representational capability of SBVR in combination with any of the process modeling languages ranges from 13 to 20 constructs, while overlap ranges from 2 to 6. MOO indicates in this case that a combination of SBVR and EPC is good from a clarity perspective, although such a combination offers just 16 representation theory constructs. Table 1 also shows that such a combination would be lacking the representation of the *conceivable* and *lawful state space*, constructs that, intuitively, are important for modeling organizational policies and rules. The integration of SBVR with BPMN shows an overlap of six constructs with only one additional BWW construct covered by SBVR. The main integration points between SBVR and BPMN are thing, property, class, system, system composition, and system decomposition.

In similar fashion, the combination of SWRL and BPMN produces the highest representation power for any pair of languages that includes SWRL as the rule modeling counterpart. As in the case of SBVR and BPMN, SWRL and BPMN provide representation for twenty BWW constructs (i.e. MOC = 20). However, this representation capability in SWRL and BPMN comes at a smaller cost of construct overlap. Hence, MOC and MOO together would point to SWRL and BPMN being the preferred option over SBVR and BPMN. Of course, the combination of SRML and BPMN is still superior in terms of BWW representation capability. Accordingly, when a higher representation capability is required, the SWRL and BPMN combination should be chosen only if SRML is not an option.

The results discussed thus far do not consider that some rule modeling languages might also be designed to be used in combination with other rule specifications (e.g. business rule vocabulary specification with a production rule specification). In order to investigate this aspect, we

Table 3
Overlap analysis of SBVR/PRR and SBVR/SWRL language pairs.

R \ P	CPN		EPC		$IDEF_3$		BPMN	
SBVR+PRR	10	7	12	8	10	6	14	12
	6	3	3	4	5	4	7	2
SBVR+SWRL	10	7	12	7	8	5	12	11
	6	3	4	5	6	3	8	1

Legend for each quadrant.

1. $P \triangle R$
2. $P \cap R$
3. $P \setminus R$
4. $R \setminus P$

consider the use of the Semantics of Business Vocabulary and Business Rules together with two specifications for rule representation, viz. PRR and SWRL. This combined overlap analysis is shown in Table 3.

The consideration of pairs of the chosen rule modeling specifications shows that, while SBVR/PRR/BPMN is a stronger combination than SBVR/SWRL/BPMN (based on the theory of MOC), the combination of SRML and BPMN is still a preferred option due to its higher power of BWW representation capability. Furthermore, the law of parsimony dictates that the smallest possible number of modeling languages should be chosen to provide MOC since the addition of a modeling language bears with it added complexity and added training requirements for the modelers.

These initial findings are significant for four reasons. First, they provide guidance to developers of modeling languages by highlighting the areas that require improvement. Clearly, there are a number of constructs missing across the board and further investigation is necessary in terms of their criticality and potential addition in future revisions of these modeling languages. Second, our results can provide guidance to organizations in adoption of a specific set of modeling languages for their process documentation efforts. For example, an organization already using BPMN for process modeling has a theoretical basis to support the choice of SRML over SWRL, SBVR or PRR. An organization already committed to using colored Petri nets would be inclined to adopt SBVR (over the other considered rule modeling options) to obtain higher representational power with lower construct overlap, or may investigate a switch to BPMN and the associated costs of converting their models. Third, once a pair of languages is in use, organizations may use the results of the analysis (Table 1) as guidance for the development of consistent workaround policies to alleviate the weaknesses their modelers will encounter while using the language pair to model organizational procedures. Last, our work shows that SRML, despite the lack of recent activity relating to the specification, is a stronger specification than the newer offerings. Since many approaches to specify standard representations for rules have been initiated in different standards bodies since SRML was proposed, a further investigation into the factors that determine continuance of specification development would be a logical extension of this work.

6. Conclusions

This paper presents the first theory-based analysis of representational capabilities of four rule modeling languages, viz. SRML, SWRL, PRR, and SBVR. The consideration of our analyses, together with existing representational analyses of four popular conceptual process modeling languages, has allowed us to provide some initial direction on which combinations of languages provide users with the best representational capabilities. Our findings show that the combination of BPMN with SRML provides users with the highest representation power while suffering an amount of construct overlap that is no higher than that of other language pairs. However, the analysis also shows that even this combination of languages is still deficient in some constructs, viz. *history*, *conceivable event space*, and *lawful event space*.

While our initial findings encourage further investigation of the integration of process and rule modeling languages, there are some known limitations to our current approach. First, different authors performed the analysis of EPC and $IDEF_3$, and their interpretation of the language constructs may differ from ours. Second, the published representational analyses of modeling grammars generally do not include analysis of representational capability of *combinations* of the modeling grammar constructs, focusing instead on representation of each construct in isolation. Third, we assume that each BWW representation model construct is equally important for the rule modeling domain. In the future, we will conduct an expert study to investigate a more refined ranking of ontological constructs in order to determine the criticality of missing representations for the domain of business rule and process modeling. Fourth, intuitively, some amount of construct overlap is necessary for the purposes of integrating two types of modeling languages. In this paper, we adopt the MOO and MOC theories; however, we recognize that an overlap of at least one construct should exist. Exactly how much overlap is necessary remains an open research question. Finally, the indication of overlap needs to be empirically tested, as is the case with any representational analysis, and the results overall need to compared against a real-world example. Overlap analysis considers overlap at the ontological construct level, hence does not take into account the granularity of representation of the modeling language. For example, two languages might have a construct for representing events; however, they may represent different types of events. Moreover, the use of a language in practice might differ to its specified expected use (potentially due to representational shortcomings such as lack of completeness and clarity of representation), hence the need to identify actual users of the considered languages.

Work is currently underway to evaluate additional rule modeling specifications (e.g. RuleML) for inclusion, since more suitable combinations of languages may be identified if more rule languages are included. Further work is also required to develop a cost/benefit calculation that would indicate whether the additional representational capability provided by, for example, SRML over and above BPMN, is worth the complexity and cost of adding an additional language. Last, we see the need for further research that focuses on how to achieve a meaningful and seamless integration of business process and business rule modeling languages. In a related stream of research, work has commenced on the analysis of various extensions to the Petri nets notation in order to determine whether the notation extensions denote a representational improvement with regard to the BWW model.

Acknowledgements

We wish to thank the organizers, reviewers and audience of the 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2007) for their valuable and insightful comments on an earlier version of this work. We extend further gratitude to the reviewers involved in the *Information Systems* review process. We also would like to thank Gerrit Kamp for his contribution to the initial analysis of rule modeling languages.

References

[1] A. Charfi, M. Mezini, Hybrid web service composition: business processes meet business rules, in: Paper presented at the Proceedings of the Second International Conference on Service Oriented Computing, New York, NY, USA, 2004.
[2] R.M. Chisholm, The basic ontological categories, in: K. Mulligan (Ed.), Language, Truth, and Ontology, vol. 51, Kluwer Academic Publishers, Dordrecht, 1992, pp. 1–13.
[3] C. D'amato, S. Staab, N. Fanizzi, On the influence of description logics ontologies on conceptual similarity, in: Paper presented at the 16th International Conference on Knowledge Engineering: Practice and Patterns, Italy, 2008.
[4] M. Ehrig, S. Staab, QOM – quick ontology mapping, in: Paper presented at the Third International Semantic Web Conference (ISWC2004), Japan, 2004.
[5] Gartner Group, Making the Difference: The 2008 CIO Agenda (Gartner EXP Premier Reports ed.), Gartner Group, Inc., Egham, UK, 2008.
[6] P. Green, M. Rosemann, Perceived ontological weaknesses of process modelling techniques: further evidence, in: Paper presented at the 10th European Conference on Information Systems, Gdansk, 2002.
[7] P. Green, M. Rosemann, M. Indulska, C. Manning, Candidate interoperability standards: an ontological overlap analysis, Data and Knowledge Engineering 62 (2) (2007) 274–291.
[8] P.F. Green, M. Rosemann, M. Indulska, Ontological evaluation of enterprise systems interoperability using Ebxml, IEEE Transactions on Knowledge and Data Engineering 17 (5) (2005) 713–725.
[9] H. Herbst, G. Knolmayer, T. Myrach, M. Schlesinger, The specification of business rules: a comparison of selected methodologies, in: A.A. Verijn-Stuart, T.-W. Olle (Eds.), Methods and Associated Tools for the Information System Life Cycle, Elsevier, Amsterdam, 1994, pp. 29–46.
[10] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, Swrl: A Semantic Web Rule Language Combining Owl and Ruleml, W3C Member Submission, 2004.
[11] K. Jensen, Coloured petri nets and the invariant method, Theoretical Computer Science 14 (3) (1981) 317–336.
[12] D.B. Jos, E. Marc, F. Cristina, J.M.-R. Francisco, S. Francois, W. Moritz, Ontology mediation, merging and aligning, in: D. John, W. Paul, S. Rudi (Eds.), Wiley, New York, 2006.
[13] G. Kappel, S. Rausch-Schott, W. Retschitzegger, Coordination in workflow management systems – a rule-based approach, in: W. Conen, G. Neumann (Eds.), Coordination Technology for Collaborative Applications – Organizations, Processes, and Agents, 1364 ed., vol. 1364, Springer, Berlin, 1998, pp. 99–119.
[14] G.F. Knolmayer, R. Endl, M. Pfahrer, Modeling processes and workflows by business rules, in: W.M.P. van der Aalst, J. Desel, A. Oberweis (Eds.), Business Process Management, Models, Techniques, and Empirical Studies, vol. 1806, Springer, London, 2000, pp. 16–29.
[15] A. Kovacic, Business renovation: business rules (still) the missing link, Business Process Management Journal 10 (2) (2004) 158.
[16] J. Krogstie, P. McBrien, R. Owens, A.H. Seltveit, Information systems development using a combination of process and rule-based approaches, in: R. Andersen, J.A. Bubenko Jr., A. Solvberg (Eds.), Third International Conference on Advanced Information Systems Engineering (CAiSE '91), Springer, Trondheim, Norway, 1991, pp. 319–335.
[17] R. Lu, S. Sadiq, A survey of comparative business process modeling approaches, in: Paper presented at the Proceedings of the 10th International Conference on Business Information Systems, Poznan, Poland, 2007.
[18] A. Maedche, S. Staab, Comparing Ontologies – Similarity Measures and a Comparison Study (Internal Report 408), Institute AIFB, University of Karlsruhe, 2001.
[19] R.J. Mayer, C.P. Menzel, P.S. Mayer, Idef3: A Methodology for Process Description, 1991.
[20] P. Mcbrien, A.H. Seltveit, Coupling process models and business rules, in: Proceedings of the IFIP 8.1 WG Conference on Information Systems Development for Decentralized Organizations, 1995.
[21] J. Meng, S.Y.W. Su, H. Lam, A. Helal, Achieving dynamic interorganizational workflow management by integrating business processes, events and rules, in: Paper presented at the Proceedings of the 35th Annual Hawaii International Conference on System Sciences System Sciences, HICSS 2002, Waikoloa, HI, 2002.
[22] OMG, Semantics of Business Vocabulary and Business Rules Specification, Second Interim Specification, dtc/06-08-05, 2006, p. 380.
[23] OMG, Production Rules Representation, Beta 1, Adopted Specification, dtc/07-11-04, 2007, p. 70.
[24] OMG, Business Process Modeling Notation, V1.1 OMG Available Specification, formal/2008-01-17, 2008, p. 294.
[25] J. Recker, Understanding process modelling grammar continuance: a study of the consequences of representational capabilities, Ph.D. Thesis, Queensland University of Technology, Brisbane, Australia, 2008.
[26] J. Recker, M. Indulska, M. Rosemann, P. Green, Do process modelling techniques get better? A comparative ontological analysis of BPMN, in: Paper presented at the Proceedings of the 16th Australasian Conference on Information Systems, Sydney, Australia, 2005.
[27] J. Recker, M. Indulska, M. Rosemann, P. Green, How good is BPMN really? Insights from theory and practice, in: Paper presented at the 14th European Conference on Information Systems, Goeteborg, Sweden, 2006.
[28] M. Rosemann, P.F. Green, M. Indulska, 2004. A reference methodology for conducting ontological analyses, in: Paper presented at the Proceedings of the 23rd International Conference on Conceptual Modeling, Shanghai, China.
[29] M. Rosemann, J. Recker, M. Indulska, P. Green, A study of the evolution of the representational capabilities of process modeling grammars, in: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering – CAiSE 2006, vol. 4001, Springer, Luxembourg, Grand-Duchy of Luxembourg, 2006, pp. 447–461.
[30] G.K. Royce, Integration of a business rules engine to manage frequently changing workflow: a case study of insurance underwriting workflow, in: Paper presented at the Proceedings of the 2007 Americas Conference on Information Systems, Keystone, CO, 2007.
[31] A. Scheer, O. Thomas, O. Adam, Process modeling using event-driven process chains, in: Process-Aware Information Systems: Bridging People and Software through Process Technology, Wiley, Hoboken, New Jersey, 2005, pp. 119–145.
[32] K. Schmidt, Of maps and scripts the status of formal constructs in cooperative work, Information and Software Technology 41 (6) (1999) 319–329.
[33] G. Steinke, C. Nikolette, Business rules as the basis of an organization's information systems, Industrial Management and Data Systems 103 (1/2) (2003) 52.
[34] M. Thorpe, C. Ke, Simple Rule Markup Language (SRML): a general XML rule representation for forward-chaining rules, XML coverpages, 2001, p. 1.
[35] M. Uschold, M. King, S. Moralee, Y. Zorgios, The enterprise ontology, Knowledge Engineering Review 13 (1998) 31–89.

[36] W.M.P. Van Der Aalst, A.H.M. Ter Hofstede, B. Kiepuszewski, A.P. Barros, Workflow patterns, Distributed and Parallel Databases 14 (1) (2003) 5–51.
[37] G. Wagner, Rule modeling and markup, in: N. Eisinger, J. Maluszynski (Eds.), Reasoning Web, 3564 ed., vol. 3564, Springer, Berlin, Msida, Malta, 2005, pp. 251–274.
[38] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, Information Systems Journal 3 (4) (1993) 217–237.
[39] R. Weber, Ontological Foundations of Information Systems, monograph ed., Coopers & Lybrand, Melbourne, Australia, 1997.
[40] B. Wyssusek, Ontology and ontologies in information systems analysis and design: a critique, in: Paper presented at the Proceedings of the 2004 Americas Conference on Information Systems, New York, USA, 2004.
[41] B. Wyssusek, On ontological foundations of conceptual modelling, Scandinavian Journal of Information Systems 18 (1) (2006) 63–80.