Representational Analysis of Business Process and Business Rule Languages

3 authors:

Vid Prezel
1 PUBLICATION 4 CITATIONS

Dragan Gasevic
The University of Edinburgh
447 PUBLICATIONS 5,528 CITATIONS

Milan Milanović
University of Belgrade
31 PUBLICATIONS 205 CITATIONS

Representational Analysis of Business Process and Business Rule Languages

Vid Prezel[1], Dragan Gašević[1], Milan Milanović[2]

[1] School of Computing and Information Systems
Athabasca University, Athabasca, AB, Canada
vprezel@gmail.com, dgasevic@acm.org
[2]GOOD OLD AI Research Network, University of Belgrade, Serbia
milan@milanovic.org

Abstract. We conducted a representational analysis of a set of business process and business rule modeling languages by using the well-known Bunge-Wand-Weber (BWW) representation model. Our paper comparatively assesses their ontological deficiencies and explores different possibilities of combining process modeling languages with rule modeling languages in order to achieve the highest ontological completeness. We demonstrate that a combination of BPMN and R2ML (rBPMN) offers the highest ontological completeness among the languages studied.

Keywords: Business process modeling, business rules, representational analysis, formal ontology

1 Introduction

With the growing complexity of today's information systems, business process modeling has gained a lot of attention by both academic and industry communities. In fact, research on business process modeling involves different areas such as software engineering, service-oriented architectures, business process management, formal reasoning, and the Semantic Web. To analyze business process modeling languages, typically, different types of usage patterns are leveraged to estimate suitability of a business process modeling language in addressing certain tasks [6]. For example, workflow patterns are used for general analysis of languages for commonly-used control-flows or service orchestrations, while server-interaction patterns are used to evaluate the support for modeling service choreographies.

Recently, aiming to support modeling and development of more agile business processes, the research community started investigating the integration of more declarative formalism with business process modeling languages. In fact, relations of business process modeling with business rules are one of the most interesting research directions [26]. On the one hand, rules are seen as an alternative to model and/or implement business processes. On the other hand, rules are seen as complements to the existing business process modeling languages (so-called hybrid languages). To evaluate both groups of contributions, traditional business process modeling methods based on different types of patterns (workflow and service interaction) have again been used [6]. While pattern-based evaluation is useful for indicating a level of support for solving different types of tasks, pattern-based analysis is not suitable to evaluate a general level of representational support to model information systems in general.

Borrowing from the formal ontology research, the research community has proposed the use of formal ontological models such as the Bunge-Wand-Weber (BWW) model [13]. In fact, the research community has developed methodologies that allow for representational analysis of modeling languages (so-called coverage analysis) of combinations of modeling languages (so-called overlap analysis) [23]. Adopting principles of these methodologies, works presented in [21-22] conducted a representational analysis of business process and rule languages. However, those efforts analyzed an older version of the Business Process Modeling Notation (BPMN). Moreover, that work did not include any existing proposals for a hybrid (rule-enhanced) business process modeling language. Adopting the methodology proposed in [23] and trying to take more recent business process modeling languages into consideration, this paper aims to provide:

- Representational analysis of the current and under-development versions (1.2 and 2.0) of the BPMN business process modeling language;
- Representational analysis of rule-based and hybrid modeling languages and their comparison with the current versions of the BPMN language;
- Representational analysis of pairing the current versions of the BPMN language with different business rule languages.

2 Background

This section describes a selected set of rule and process modeling languages, basic elements of the followed representational analysis, and related work on conducting representational analysis by following similar methodological principles.

2.1 Business process and business rules languages

We analyze the five modeling languages: BPMN [1-2], PRR [3], SWRL [4], R2ML [5], and rBPMN [6]. While we are aware that this is not a complete set of languages and we are working on the analysis of a few others (OCL, SBVR, and RIF), this selection might already provide some useful indicators comparing to the results reported in [21]. The choice of BPMN was due to its popularity and adoption by a wide range of process modeling tool vendors and organizations [1]. In the representational analysis conducted by Recker et al. [7], it was concluded that there is no representation for states in BPMN v1.0. So, one of our goals was to check what is the support for state modeling in the more recent versions of BPMN. In our analysis, we also wanted to include languages of four types of rules according to Wagner et al's classification cited in the PPR standard [3]. This coverage was assured by the inclusion of the following three rule languages: SWRL – supports integrity rules; PRR – production rules; and R2ML – integrity, derivation, production and reaction rules. We selected SWRL [4], as it is a widely-used (integrity) rule language for the Semantic Web; PRR [3], as it is the OMG's standard for modeling production rules; and R2ML [5], as it can represent all types of rules. Lastly, we chose rBPMN, because it represents a hybrid language, incorporating process (BPMN) and rule languages (R2ML).

2.1.1 BPMN

The Business Process Modeling Notation (BPMN) [1] is a graphical notation and a language for modeling business processes. It was developed by Business Process Management Initiative (BPMI) and is based on other notations such as IDEF, UML, LOVeM, RosettaNet, and Event-driven Process Chains [7]. The major goal that led to the development of BPMN was to introduce a business process modeling notation that is acceptable and usable not only by process developers responsible for technology implementation, but also by business analysts and business managers responsible for the design and management of business processes. The other goal that led to the development of BPMN was to allow BPMN instances to be the source of an executable process, which means that there would be a mapping from one or more BPMN notation instances to an execution level instance. This allows BPMN to map directly to languages that were designed for the execution of business processes. Since the merger of BPMI with OMG in 2005, BPMN is now maintained by OMG. The first version BPMN 1.0 was released to public in May 2004 and adopted by OMG in 2006. The current version is BPMN 1.2, with BPMN 2.0 Beta1 in a finalization phase. According to OMG's website [1], there are currently sixty-two implementations and four planned implementations in practice.

2.1.2 Production Rule Representation (PRR)

PRR [3] was defined by vendors of business rules engines such as ILOG, Fair Isaac, LibRT, IBM, Pega, Corticon, TIBCO, academic community (RuleML.org), and UML tool vendors [3]. The current version (1.0 from December 2009) is now an adopted OMG standard, and a formal model for production rules. It uses a UML style for rule representation. PRR includes two types of rules: Forward chaining inference rules and sequentially processed procedural rules. Forward chaining rules (e.g., Rete-model) are used for common production rule engines, which makes it dependant on the types of rules executed by rule engines. Sequentially processed procedural rules are used for tools that extract simple business logic as non inference production rules [3]. The PRR is defined at two levels. A core structure of PPR (referred to as PRR Core) includes general rule and production rule model. PRR OCL structure includes an extended OCL normative expression language to allow compatibility with non UML representations. The high level structure of PRR is similar to R2ML, however in PRR there is no consideration of correspondences to (Semantic) Web rule languages [8].

2.1.3 Semantic Web Rule Language (SWRL)

Semantic Web Rule Language (SWRL) is a submission to the W3C trying to combine the rules (RuleML) and ontologies (OWL-DL and Lite). Rules in SWRL are expressed in terms of OWL constructs such as individuals, properties, literals, and classes. Rules are written as antecedent-consequent pairs [9]: The antecedent as the rule body, and the consequent as the rule head. This means whenever the conditions specified in the body are true, than the conditions specified in the head must also be true. The head and body consist of zero or more atoms. If there are multiple atoms, they are treated as a conjunction and could be transformed into separate rules with atomic heads or consequents. If the body has zero atoms, it is satisfied by every interpretation, and thus, the head must also be satisfied by every interpretation. The same rule applies if the head has zero atoms. It is not satisfied by any interpretation, therefore the body is also not satisfied by any interpretation [4]. While SWRL is not standardized, it is a widely-used (or more modestly saying – widely-considered) language supported by a few commonly-used reasoners.

2.1.4 R2ML

A business rule is a statement that aims to influence or guide behavior and information in an organization [24]. REWERSE I1 Rule Markup Language (R2ML) is a general rule markup language [5]. It is originally designed to support rule interchange (thus, markup in its name), but it is also a comprehensive rule modeling language with a UML-based graphical concrete syntax. It can represent four types of rules, namely, integrity, derivation, reaction, and production. R2ML is built by using model-driven engineering principles, which include: metamodel, an XML-based textual concrete syntax, and a graphical concrete syntax (so-called URML [25]). A complete reference of R2ML can be found in [5]. All R2ML rule definitions are inherited from the Rule concept (class in meta-model). Each type of rule is defined over the R2ML vocabulary, where elements of the vocabulary are used in logical formulas (e.g., LogicalFormula – with no free variables) through the use of Atoms and Terms. An important aspect of R2ML is that it distinguishes between object and data atoms.

2.1.5 Rule-Based BPMN (rBPMN)

The rBPMN language is a product of integration of BPMN and R2ML, and it is defined by weaving the elements of the BPMN and R2ML abstract syntaxes (metamodels) [10]. The main element in the rBPMN language is a *RuleGateway*, which was added in the *Process* package of the BPMN metamodel (current submission for the BPMN 2.0 metamodel) and which actually relates to R2ML Rules. In this way, an R2ML Rule (i.e., reaction, derivation, production or integrity rule) can be placed into a process as a Gateway, but at the same time, the rule does not break the R2ML Rule syntax and semantics. rBPMN has been designed to support a rule-enhanced process-oriented modeling of service orchestrations and choreographies. More details about this language can be found in [10].

2.2 Representation Theory

According to Bunge [11], real-world systems in any domain can be explained by using an ontology. This can be done by defining structure, properties, and interaction between things of a domain under study [11]. Using a language L to describe topics in a domain D, an ontology provides a catalog of things (represented as concepts, relations, and predicates of language L) assumed to exist in domain D [12]. Any real world system can be explained in such a way (i.e., by an ontology). The application of ontology for the purpose of representation is known as representational analysis. Wand and Weber [13-15], adopted Bunge's ontology [11], and developed a theory that consists of state tracking, decomposition, and representation models. The last one, the representation model is used in information systems domain, and is now known as the Bunge-Wand-Weber (BWW) representation model. The BWW model defines a set of constructs necessary to provide a complete representation of all things and their interactions in a real world. For a detailed description of BWW representation model and constructs please see, for example [13].

2.3 Related work

The Bunge-Wand-Weber (BWW) representation model has been so far the most widely-used for the ontological analysis of language grammars for business system analysis. In 2005, Green and Rosemann evaluated over twenty research projects that used the BWW model in the area of conceptual modeling [16]. Several of those studies focused on process modeling. Keen et al. [17] for example, evaluated flowcharts and flow diagrams in order to determine their ontological completeness. Green et al. [18] analyzed event-driven process chain notation using the BWW model, and different modeling standards for enterprise system interoperability [19], to determine their ontological completeness and clarity. In terms of evaluating business process and business rule modeling languages specifically, zur Muehlen et al., conducted a BWW representational analysis for several different languages particularly relevant to compliance management [20-21]. Recker et al. conducted a representational analysis with a focus on BPMN 1.0 [7] and process modeling [27]. Opdahl et al. conducted an ontological evaluation of UML [28]. Our work complements the work of [20-22] by analyzing the current version of BPMN, and compares it with an additional set of rule languages aiming to determine a maximum ontological completeness.

3 Methodology

For our analysis, we selected five business process and business rule modeling languages. First, we obtained clear definitions of each of the languages from language specifications and their meta-models. We then selected relevant language constructs[1]. To analyze the selected languages, we followed a reference methodology for conducting ontological analysis given in [23]. To do so, we examined the BWW representation model constructs [13-15] and defined relevant ontology constructs which we than used in our reference model[2]. We than started the process of identifying correspond-

[1] Relevant constructs are the major language constructs relevant for presenting concrete problem domains. For example, for process languages (all BPMN versions) all constructs that were included in their visual notations (as per their specification) were considered relevant as this was also useful for evaluation of visual notation itself. For example, Opdahl et al selected 67 (out of 216) constructs when evaluating UML [28]. We performed similar selection, but for the languages in our scope.

[2] As indicated in the related work section, there are lots of BWW-based studies (and interpretations of the BWW model), and most of them use different numbers of *relevant* BWW constructs. In our case, we selected 28 main BWW constructs based on the original work from [13]. The original BWW work leaves some room for interpretation, and thus different ana-

ing constructs in the modeling language. Based on [13], [22], [18], [20], we divided our reference model into four main clusters: Thing, State, Event, and System. We then defined the subgroups of each cluster and relevant BWW constructs. We performed a representational analysis and compared each of the language constructs with constructs of our reference model and vice versa.

We first mapped the core set of the language constructs for BPMN 1.2 and noted the results. We then proceeded with the extended set of BPMN 1.2. This was then followed by mapping of the basic and extended sets of constructs for BPMN 2.0, Beta 1. After the mappings were completed for both versions (1.2 and 2.0) of BPMN, we marked results to determine ontological clarity. We looked for any differences between corresponding language constructs and BWW representation model constructs, as this provides us with an indication of a representational deficiency. We followed similar steps of our representational analysis for our selection of business rule languages. Finally, we compared the findings for rule languages with those for BPMN.

3.1 Ontological completeness

When analyzing the results, we looked at the extent to which a language has construct deficit comparing to the BWW representation model [13-15]. As such, this approach can be used as a measure of ontological completeness. Ontological completeness determines whether users of a given modeling language are able to represent all relevant real world scenarios when modeling with the given modeling language.

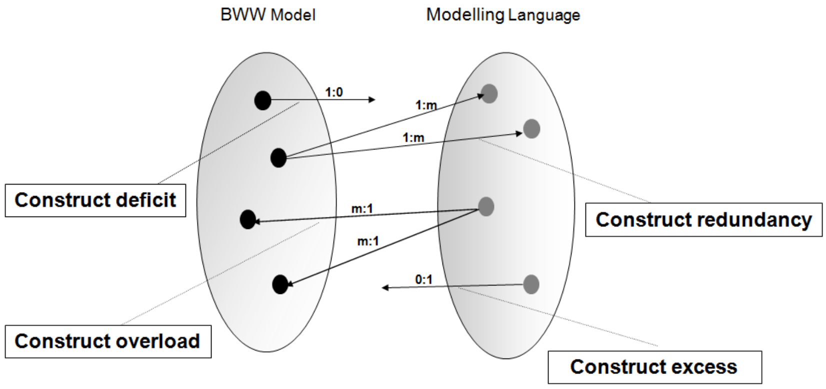

Fig. 1. Ontological completeness and clarity

Ontological clarity of a modeling language is determined by the extent to which language constructs are deemed to be overloaded, redundant, excessive or in deficit [23]. These metrics are illustrated in Fig. 1. Construct overload results when there are many to one (m:1) relationship mappings among constructs of a modeling language and the BWW model (i.e., one element of the modeling language can be used to

lyses might results in different number of BWW constructs. For example, zur Muehlen and Indulska used 29 main BWW constructs [21]. Their study lists the 'Acts on' BWW construct as a separate construct in the Event cluster, which is not the case in our study, as we could not find it in the original BWW model nor was the rationale for its inclusion given in [21].

represent many constructs of the BWW model); while redundancy occurs when there is one to many (1:m) relationship mappings (i.e., many elements of the modeling language can be used to represent one element of the BWW model). Construct excess represents zero to 1 (0:1) mapping, where at least one language construct does not map to any construct in the BWW ontological model. Construct deficit occurs when at least one construct in the BWW model does not map to any construct in an analyzed language. This can be described as a one to zero (1:0) mapping relationship.

3.2 Overlap analysis

In a scenario when none of the studied languages provides a complete representation capability, overlap analysis [19] is performed. This analysis combines a maximum ontological completeness and a minimum ontological overlap. This is useful for evaluating hybrid languages that already include combination of business process and business rule languages. It is also useful for exploring other language combinations or integrations which might offer a minimum overlap and a maximum completeness. With the overlap analysis, we are able to determine a symmetric difference and intersection of analyzed modeling languages [21]. By a symmetric difference, we determine a number of BWW constructs that are represented with no overlap for a given hybrid language or a language combination. With an intersection, we look at the number of concepts which can be represented additionally by a known hybrid language or by a newly proposed combination of languages. We are also able to determine a relative complement [21] for rule and process combinations of languages. This means how many BWW constructs were contributed by a rule modeling language to a process modeling language and vice versa.

4 Data Collection and Analysis

4.1 Representation Analysis of BPMN, versions 1.2 and 2.0

We started with the identification of the core constructs in BPMN 1.2 from the language specification. We then proceeded with the extended set of BPMN 1.2, which was followed by BPMN 2.0 Beta 1 core and extended sets. We performed a complete representational analysis of the core and extended sets of constructs for both versions of BPMN. We were interested in evaluating possible deficiencies in the above mentioned languages sets. As described in section 3.1, we examined four types of representational deficiencies: construct deficit, redundancy, overload, and excess. The lack of representation for particular BWW constructs means that users will have difficulties modeling certain scenarios in a real world domain. Table 1 shows our results for all four sets of BPMN constructs. The constructs columns show the number of constructs that exhibit a certain deficiency. The percentage columns indicate the percentage of constructs that reveal a particular deficiency. Both core versions of BPMN have 39.3% of deficit. The deficit is reduced in the extended sets of BPMN, with the extended set of BPMN 2.0 offering the lowest construct deficit of 32.1%.

Table 1. Construct Deficit, Redundancy, Overload, and Excess

	BPMN 1.2 Core		BPMN 2.0 Core		BPMN 1.2 Ext		BPMN 2.0 Ext	
	Constructs	Percentage	Constructs	Percentage	Constructs	Percentage	Constructs	Percentage
Deficit	11	39.3%	11	39.3%	10	35.7%	9	32.1%
Redundancy	11	39.3%	11	39.3%	16	57.1%	16	57.1%
Overload	5	17.9%	5	17.9%	29	103.6%	30	107.1%
Excess	4	14.3%	5	17.9%	16	57.1%	22	78.6%

In terms of construct redundancy both core sets offer lowest redundancy. This is due to the fact that core sets of elements include a smaller number of constructs overall, which is beneficial in terms of complexity. However, they offer a lower level of construct completeness and higher level of construct deficit. Both extended sets have a redundancy rate of 57.1%, comparing to 39.3% for the core sets of BPMN. Excessive redundancy can potentially cause some confusion to the languages users as to when to use a particular language construct. For example, all BPMN language sets include Pool and Lane language constructs. Both of these constructs map to BWW construct Thing which could cause confusion as to which construct to use, for example, to represent a department in an organization.

In terms of construct overload, both core sets of BPMN offer 17.9% overload, comparing to 103.6% for extended BPMN 1.2 set, and 107.1% for BPMN 2.0 set. This is again due to the number of constructs each sets offer and the number of constructs that actually map to the 28 selected BWW constructs. For example, core BPMN 2.0 has only 14 constructs comparing to 68 for the extended set. Out of 68 constructs, 30 constructs map to more than one BWW construct. As an example for BPMN 1.2, the Lane construct maps to system, subsystem, system composition, system environment, system decomposition, level structure, and thing. This means that users may be confused as to when to use this construct when modeling for example, a department in an organization, a seller/ buyer, or an application system.

In terms of excess, all of the evaluated language sets have excess, which means they have constructs that cannot be mapped to any BWW construct. For example, certain BPMN constructs such as Off-Page-Connector, Activity looping, and Association Flow, have no real world meaning from the BWW perspective and are included as excess. Those types of constructs may be useful for actual modeling activities, but not for capturing semantics of a real world domain. Large numbers of construct excess also contributes to the additional complexity. The core set of BPMN 1.2 has 14.3% excess, comparing to 17.9% for the core set of BPMN 2.0. The extended set of BPMN 1.2 has 57.1% excess, comparing to 78.6% for BPMN 2.0.

Based on the above analyses, we conclude that the extended set of BPMN 2.0 offers the lowest construct deficit, and therefore the highest ontological completeness. Ontological completeness is not 100%, as there are still nine BWW constructs that cannot be represented with the current version of BPMN. Despite that and its higher construct redundancy, overload, and excess, the extended set of BPMN 2.0 offers the most complete set of constructs to model scenarios in a real-world domain.

4.2 Representational Analysis of Rule Languages and Comparison with BPMN

Since we indentified that there are nine BWW constructs that cannot be represented with any of the constructs from the extended set of BPMN 2.0, we were further motivated to evaluate a few other languages that may offer representation for the remaining nine BWW constructs. BPMN 2.0 has almost no representation in the State cluster and no representation for a few of the other BWW constructs in the remaining three clusters. As this particular (State) cluster is important for modeling business rules, we further examined PRR, R2ML, SWRL business rule languages, and a hybrid language rBPMN, to determine if ontological completeness can be further improved. Table 2 shows the mappings of these languages compare to the extended set of BPMN 2.0.

Table 2. BPMN, PRR, R2ML, SWRL, rBPMN construct mapping

BWW Construct	BPMN 2.0 Ext	PRR 0.5	R2ML 0.5	SWRL 1.0	rBPMN
THING	+	-	+	+	+
PROPERTY	+	+	+	+	+
CLASS	+	+	+	+	+
KIND	+	-	-	-	+
STATE	-	-	+	-	+
CONCEIVABLE STATE SPACE	-	-	-	+	-
LAWFUL STATE SPACE	-	+	-	-	-
STATE LAW	-	+	+	-	+
STABLE STATE	-	-	-	-	-
UNSTABLE STATE	-	-	-	-	-
HISTORY	+	-	-	-	+
EVENT	+	-	+	-	+
CONCEIVABLE EVENT SPACE	-	-	+	-	+
LAWFUL EVENT SPACE	-	-	+	-	+
EXTERNAL EVENT	+	-	-	-	+
INTERNAL EVENT	+	-	-	-	+
WELL-DEFINED EVENT	+	-	-	-	+
POORLY-DEFINED EVENT	+	-	-	-	+
TRANSFORMATION	+	-	+	+	+
LAWFUL TRANSFORMATION	+	+	+	+	+
COUPLING	+	-	-	-	+
SYSTEM	+	+	-	-	+
SYSTEM ENVIRONMENT	+	-	-	-	+
SYSTEM COMPOSITION	+	+	-	-	+
SYSTEM DECOMPOSITION	+	-	-	-	+
SYSTEM STRUCTURE	-	-	-	-	-
SUBSYSTEM	+	-	-	-	+
LEVEL STRUCTURE	+	-	-	-	+
	19/28	7/28	10/28	6/28	23/28
	67.9%	25.0%	35.7%	21.4%	82.1%
EXCESS	+	+	+	+	+

For PRR 0.5 and SWRL 1.0, we included mapping results from zur Muehlen et al. [21], as they performed a similar comparison for these two particular language versions. We performed the mapping by following the methodology described in Section 3. From the overall ontological completeness perspective, as described earlier, BPMN offers 67.9% completeness, PRR 25%, R2ML 35.7%, SWRL 21.4%, and rBPMN 82.1%. rBPMN offers the highest completeness, that is, combing BPMN and R2ML is beneficial from the overall ontological completeness perspective.

In terms of construct deficit, rBPMN has the lowest deficit of 17.9%. As indicated in Table 3, there are only five out of 28 BWW constructs that cannot be represented with rBPMN, which gives a construct deficit rate of 17.9%.

Table 3. Construct Deficit

Construct Deficit	BPMN 2.0 Ext	PRR 0.5	R2ML 0.5	SWRL 1.0	rBPMN
# of Constructs	9	21	18	22	5
Percentage	32.1%	75.0%	64.3%	78.6%	17.9%

The extended set of BPMN 2.0 also offers a relatively low construct deficit of 32.1%. R2ML has a construct deficit of 64.3%, followed by PRR with 75% and SWRL with 78.6%. Fig. 2 illustrates this comparison. The y axis is the number of BWW constructs that cannot be represented by a given language listed on the x axis.

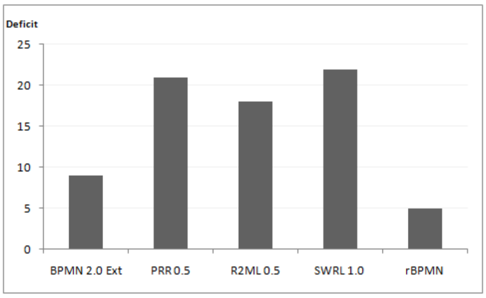

Fig. 2. Construct Deficit

In terms of construct excess, and as demonstrated in Table 2, all languages have construct excess, which means there is at least one language construct that does not map to any of the BWW constructs. From the cluster by cluster perspective, we notice that the cluster Thing is best represented with BPMN and rBPMN, which both offer a complete (100%) representation in this cluster. SWRL and R2ML have 75% representation, and PRR 50% representation. In the State cluster, BPMN and SWRL have 14.3% representation which is the lowest of all the analyzed languages. rBPMN offers the best representation of 42.9% followed by PRR and R2ML with 28.6% representation of the BWW state constructs. In the Event cluster, rBPMN is the only language that offers a complete (100%) representation of all event constructs, followed by BPMN, R2ML, SWRL, and PRR. In the System cluster, rBPMN and BPMN offer the highest representation of 85.7%, followed by PRR. Table 4 shows this comparison.

Overall, from the cluster-by-cluster perspective, rBPMN offers the highest representation in all four clusters, Thing, State, Event, and System among the five languages in comparison. Fig. 3 illustrates the results graphically. Y axis represents a percentage rate of representation for each of the four clusters per language studied.

Table 4. BWW Cluster Representation

Cluster	BPMN 2.0 Ext	PRR 0.5	R2ML 0.5	SWRL 1.0	rBPMN
Thing	100.0%	50.0%	75.0%	75.0%	100.0%
State	14.3%	28.6%	28.6%	14.3%	42.9%
Event	80.0%	10.0%	50.0%	20.0%	100.0%
System	85.7%	28.6%	0.0%	0.0%	85.7%

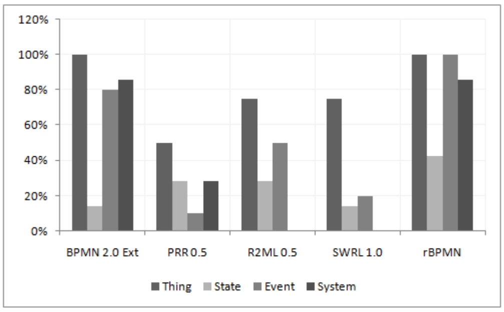

Fig. 3. Cluster by Cluster Comparison

4.3 Representational Analysis of Paired Business Process and Rule Languages

Based on the results described so far, we can conclude that rBPMN offers the highest degree of ontological completeness. Although rBPMN does not offer a complete representation capability, we can already conclude that combining business process and business rule modeling languages is beneficial – as demonstrated with rBMPN. Since rBPMN combines business process modeling language and business rule modeling language, namely BPMN 2.0 and R2ML 0.5, and we discovered that rBPMN offers better completeness than each of the two languages alone, we were interested to evaluate if this language pair also offers the best completeness among any given combination of business process and business rule modeling languages that we studied.

We further evaluated the following three language pairs: BPMN + PRR, BPMN + SWRL, and rBPMN (BPMN + R2ML). As per [21], we performed the overlap analysis and calculated symmetric difference, intersection, and relative compliment. With symmetric difference, we can determine the number of BWW constructs that are represented with no overlap for a given language combination (P∆R). With intersection, we look at the number of BWW constructs which can be represented additionally (with overlap) by a particular language pair (P∩R). To determine a relative complement, we look at how many non-overlapping BWW constructs were contributed by a business process modeling language to a business rule process modeling language (P\R) and how many non-overlapping constructs were contributed by the business rule language to the business process language (R\P). Table 5 shows the results.

Based on the results in Table 5, BPMN+R2ML (rBPMN) offers 17 distinctively represented BWW constructs free of overlap. This is the highest number of the three language combinations. The BPMN+PRR combination has 16, while BPMN+SWRL has 15 BWW constructs. The second column in Table 5 displays the intersection.

rBPMN has 6 constructs that can be additionally represented by both languages, comparing to 5 for BPMN+PRR and BPMN+SWRL. In terms of relative complement P\R and R\P, rBPMN represents 13 non-overlapping BWW constructs contributed by BPMN, and 4 constructs represented by R2ML. Fig. 4 illustrates this comparison.

Table 5. Overlap Analysis

Language Pair	PΔR	P∩R	P\R	R\P
BPMN 2.0 + PRR 0.5	16	5	14	2
BPMN 2.0 + SWRL 1.0	15	5	14	1
BPMN 2.0 + R2ML 0.5 (rBPMN)	17	6	13	4

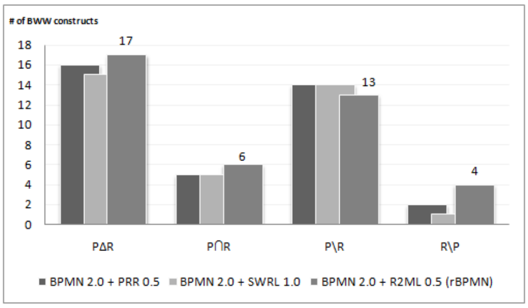

Fig. 4. Overlap Analysis

Based on our analysis, the language combination of BPMN and R2ML (rBPMN) is the most desirable language combination not only because it represents the highest number of distinct non overlapping constructs, but also the highest number of constructs that can be additionally represented with the overlap. In fact, this particular language combination is the best choice also because R2ML contributes the highest number of BWW constructs to BPMN comparing to any other language pair studied.

We cannot compare the results of our analysis for the best performing language combination (i.e., rBPMN) to the results of work presented in [21]. From the ontological completeness perspective, zur Muehlen and Indulska's best hypothetical pair, BPMN 1.0 and Simple Rule Markup Language (SRML) represents 23 constructs out of 29 used in their reference BWW model (79%). In our case, rBPMN represents 23 out of 28 included in our model (82%). zur Muehlen and Indulska's analysis also contained the "Acts on" concept in the BWW model, which we did not as already explained in footnote 2. If we consider only the 28 concepts covered in the BWW model that we used, it appears that the BPMN 1.0 + SRML combination covers 22 concepts. Two BWW concepts, "Conceivable State Space" and "Lawful State Space" are only represented in BPMN 1.0+SMRL, and three BWW concepts "History", "Conceivable Event Space', and "Lawful Event Space" are only represented by rBPMN. Out of last three, two are contributed by R2ML and one by BPMN 2.0.

The best performing combination of languages (R2ML + BPMN) analyzed in our study might have important implications for tool developers comparing to the case of BPMN 1.0 and SRML. Both constitutive languages of rBPMN are developed by using model-driven engineering principles. That fact already offers mechanisms for a solid basis of type safety and static semantic analysis needed for an effective language use. If BPMN were combined with a rule language which is designed to serve as a markup and only represented in an XML format, then there would a need to invest additional efforts in tool development to also overcome issues of different language definition mechanisms (e.g., Ecore and XML Scheme).

5 Conclusion and Future Work

A limitation of our study is that we have not included an analysis of the Rule Interchange Format (RIF) and Object Constraint Language (OCL), as they are important standards. Our on-going work takes these languages into consideration. Considering the types of language constructs supported in OCL and RIF, we can hypothesize that there will hardly be better coverage support comparing to rBPMN. This is due to the fact that RIF and OCL do not have a full support for state modeling, similar to the R2ML language. Yet, a thorough representation is to be conducted to investigate all other characteristics of RIF and OCL once it is combined with BPMN and/or other process modeling language. It will also be very important to analyze other rule standards such as Semantics for Business Vocabularies and Rules (SBVR). Furthermore, our future work will more thoroughly compare the results of our analysis with the results of representational analysis reported in previous reports [21] and [22].

From our analysis (see Table 2) as well as from the previous work [21], it appears that modeling state space is the major source of incompleteness w.r.t. the BWW model of the current process and rule modeling languages. While this might be a problem for developing some types of systems, this might not be the major point of concern for others such as service-oriented systems where statelessness is one of the main premises. Still, to have a complete support, the future languages might consider support for state space modeling. Some types of rules such as reactive and production might be a very good basis for this support in future research.

References

1. OMG, Business Process Modeling Notation, Final Adopted Specification, Online, Available: http://www.omg.org/docs/formal/09-01-03.pdf, (2009).
2. OMG, Business Process Model and Notation (BPMN), v2.0 - FTF Beta 1, Online, Available: http://www.omg.org/cgi-bin/doc?dtc/09-08-14.pdf, (2009).
3. OMG, Production Rule Representation (PRR), Specification, formal/2009-12-01, Online, Available: http://www.omg.org/spec/PRR/1.0/PDF/ (2009).
4. H. Ian, P. Peter F., B. Harold, T. Said, G. Benjamin, and D. Mike: SWRL: A Semantic Web Rule Language, Combining OWL and RuleML, W3C Member Submission, (2004).
5. Rewerse Working Group I1, "R2ML - The REWERSE I1 Rule Markup Language," Online, Available: http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=node/6.
6. Milanović, M., Gašević, D.: Towards a Language for Rule-enhanced Business Process Modeling, IEEE Int'l Enterprise Distributed Object Computing Conf., pp. 64-73. (2009).
7. Recker, J., Indulska, M., Rosemann, M., Green, P.: How good is bpmn really? Insights from theory and practice, 14th European Conference on Information Systems, pp. 1-12, (2006).

8. Nemuraite, L., Ceponiene, L., Vedrickas, G.: Representation of Business Rules in UML&OCL Models for Developing Information Systems, Proc. of 1st IFIP WG 8.1 Working Conf. Practice of Enterprise Modeling, pp. 182-196, (2008).
9. O`Connor, M., Knublauch, H., Tu, S., Musen, M.: Writing Rules for the Semantic Web Using SWRL and Jess, 8th Int'l Protégé Conference, Protege with Rules Workshop, (2005).
10. Milanović, M., Gašević, D., Wagner, G.: Combining Rules and Activities for Modeling Service-Based Business Processes, In Proc. EDOC 2008 International Workshop on Models and Model-driven Methods for Enterprise Computing, (2008).
11. Bunge, M.,: Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World, Springer, (1977).
12. Gašević, D., Djurić, D., Devedzić, V.: Model Driven Architecture and Ontology Development, Springer, (2006).
13. Wand Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars, Science And Technology, vol. 3, pp. 217-237, (1993).
14. Wand, Y., Weber, R.: On the deep structure of information systems, Information Systems Journal, vol. 5, pp. 203-223, (1995).
15. Weber, R.: Ontological Foundations of Information Systems, Melbourne, Australia: Coopers & Lybrand and the Accounting Association of Australia and New Zealand, (1997).
16. Green, P., Rosemann, M.: Ontological analysis of business systems analysis techniques: Experiences and proposals for an enhanced methodology, Business Systems Analysis with Ontologies, P. Green and M. Rosemann, Idea Group Publishing, pp. 1-27, (2005).
17. Keen, C., Lakos, C.: An Analysis of the Design Constructs Required in Process Modelling, the 1996 Int'l Conference on Software Engineering: Education and Practice , (1996).
18. Green, P, Rosemann, M., Indulska, M: Ontological Evaluation of Enterprise Systems Interoperability Using ebXML, IEEE Trans. on Knowledge and Data Eng., vol. 17, (2005).
19. Green, P., Rosemann, M., Indulska, M., Manning, C.: Candidate interoperability standards: An ontological overlap analysis, Data & Knowledge Eng., vol. 62, pp. 274-291, (2007).
20. zur Muehlen, M., Indulska, M., Kamp, G.: Business Process and Business Rule Modeling Languages for Compliance Management: A Representational Analysis, In Tutorials, At the 26th international Conference on Conceptual Modeling, pp. 127-132, (2007).
21. zur Muehlen, M., Indulska, M.: Modeling languages for business processes and business rules: A representational analysis, Information Systems, vol. 35, pp. 379-390, (2009).
22. Recker, J. Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN, 16th Australasian Conference on Information Systems, Sydney, Australia, (2005).
23. Rosemann, M., Green, P., Indulska, M.: A Reference Methodology for Conducting Ontological Analyses, 23rd International Conference on Conceptual Modeling, S.Z. H. Lu, W. Chu, P. Atzeni, Shanghai, China: Springer Verlag Berlin Heidelberg, pp. 110-121, (2004).
24. Steinke, G., Nikolette, C.: Business rules as the basis of an organization's information system, Industrial management + Data Systems, vol. 103, p. 52, (2003).
25. UML-based Rule Modeling Language, Online, Available: http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=node/7.
26. Graml, T., Bracht, R., Spies, M.: Patterns of business rules to enable agile business processes. Enterprise IS, vol. 2, no, 4, pp. 385-402 (2008).
27. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business process modeling : a comparative analysis. J.the Association for Information Systems, 10(4). pp. 333-363 (2009).
28. Opdahl, A. L., Henderson-Sellers, B.: Ontological Evaluation of the UML Using the Bunge–Wand–Weber Model. Weber Model, Soft. & Sys. Modeling, 1(1), pp. 43-67 (2002).