How Complete Are BPM Languages for Software Requirements Elicitation? A BABOK Insight

Carlos Monsalve[1, 2], Alain April[2], Alain Abran[2]

[1] CIDIS-FIEC, Escuela Superior Politécnica del Litoral, Apartado 09-01-5863, Campus Gustavo Galindo, Km. 30.5 via Perimetral, Guayaquil, Ecuador
monsalve@espol.edu.ec
http://www.espol.edu.ec
[2] GELOG, École de Technologie Supérieure, H3C 1K3, 1100 rue Notre-Dame Ouest, Montréal, Canada
{alain.april, alain.abran}@etsmtl.ca
http://www.gelog.etsmtl.ca

Abstract. Business process models (BPM) have proved to be useful for software requirements elicitation. Software development depends on the quality of the requirements specifications, and so generating a high-quality BPM is critical. A key factor in achieving this is the right choice of modeling language. A good one should be complete, and allow all the relevant phenomena in the domain to be modeled. One ontological approach to measure the completeness of BPM languages is the Bunge-Wand-Weber (BWW) representational analysis. The BWW representation model, being an upper-level ontology, often needs to be refined for a given domain. This paper presents a number of such refinements based on an analysis of the Guide to the Business Analysis Body of Knowledge (BABOK) recommendations for software requirements elicitation. Using the BWW, the completeness of two BPM languages is assessed to see whether or not the relevant concepts for software requirements elicitation are well represented.

Keywords: Business process modeling, BPM languages, requirements elicitation, BWW representation model, ontological analysis, BABOK, BPMN, Qualigram.

1 Introduction

Business process improvement has been consistently identified as one of the top "business expectations of IT" during the last five years [1]. Recently, there has been increasing interest in business process modeling (BPM) in academia and among vendors and practitioners [2-5]. Various modeling languages, such as: Petri Nets [6], IDEF [7], UML [8], Event-driven Process Chains (EPC) [9], Business Process Modeling Notation (BPMN) [10], and Qualigram [11], to name a few, have been proposed over the years to model business processes.

Business Process Modeling (BPM) theory was created to help document, communicate, or improve an organization's business processes. Business process models are also used by software engineers and business analysts for software requirements elicitation [3, 12-14]. As software development depends on the quality of the requirements elicitation activities [15], high-quality modeling of business processes is critical.

Key to a high-quality BPM is choosing the right modeling language for the task. A good modeling language needs to be complete and capable of representing all the relevant phenomena in the business domain being modeled [2]. Wand and Weber [16] state that "completeness parallels the notion of expressive power in programming languages", where "the expressive power of a modeling technique is a measurement of what can be described."

The objective of this paper is to present our research progress towards developing an analysis technique to measure and compare the completeness of a given BPM language to adequately represent the relevant concepts for software requirements elicitation. To show the usefulness of such an analysis technique, this paper analyzes the completeness of two different BPM languages used to represent software requirements elicitation concepts.

One analysis technique proposed for measuring and comparing the completeness of a BPM language [2, 17] is Bunge-Wand-Weber (BWW) representational analysis [18]. This technique is based on an ontological model (*i.e.* representation model) and is designed to study the capacity of a language to represent the various concepts of the ontological model. Since the BWW representation model was originally designed to perform ontological analysis of all types of information systems [2], it is necessary to select a set of constructs from its content that is relevant to the business domain studied [2, 19].

With this in mind, our paper addresses the following research questions: 1) What are the relevant BWW ontological concepts associated with software requirements elicitation? and 2) Can we assess the completeness of a given BPM language, and thus its capacity to faithfully support software requirements elicitation concepts?

The software requirements elicitation activity is not unique to software engineers [15]. Business analysts also interact with business stakeholders and software developers for this purpose [14, 20], and certified business analysts often use the recommendations of the Guide to the Business Analysis Body of Knowledge (BABOK) [19] to conduct their work. This is why we chose to use the BABOK recommendations in this research to select, from the BWW representation model, a relevant set of ontological constructs for software requirements elicitation.

We propose that the use of the selected constructs from the BWW representation model will make it possible to assess the completeness of any BPM language, and thus its capacity to represent the software requirements elicitation concepts. The first question we must answer is: Which BPM language should we analyze? Recent publications have revealed that different stakeholders are accustomed to using different BPM languages and techniques to represent their business processes [3, 20]. Some authors have also reported that it can be very difficult to find a common BPM language that allows effective communication between management, business analysts, and IT stakeholders [3, 20]. For the case study reported here, two different BPM languages were selected: the Qualigram language [11], which is management-oriented and its modeling tool is based on Microsoft's Visio, and BPMN [10], which is very popular with IT personnel. We wanted to study two very different points of view.

The structure of this paper is as follows. Section 2 presents a review of the BWW representation model and the justification for its use for analyzing the completeness of the two BPM languages we have chosen for software requirements elicitation. Section 3 describes the research methodology used in this study. Section 4 debates the choice of those two languages for this study, and provides a brief overview of each. Section 5 presents the results of using the proposed analysis technique. Finally, section 6 concludes with a review of the contributions of this research, its limitations, and future work.

2 The BWW Representation Model

An ontology typically attempts to represent real-world concepts, as well as the structure of the real world and the relationships between the things that make up the real world [21, 22]. Since the questions to be answered in this paper aim to identify the constructs of a BPM language required to satisfy the modeling needs for software requirements elicitation purposes, an ontology can be a useful tool.

Wand and Weber have proposed a set of ontological models [18, 22] based on Mario Bunge's work [23, 24]. These models focus on representing the basic concepts that allow any type of information system, its structure, and its behavior to be described. They believe that one of these models, named the representation model, can be useful for evaluating the expressiveness of any analysis and design grammar [16].

Basically, the representation model can be used to determine whether or not a BPM language grammar is complete (that is, does it have all the constructs needed to represent the domain being modeled?). It also helps to determine whether or not a BPM language grammar is clear (that is, does it have sufficient constructs to allow an unambiguous interpretation of the models generated?). Wand and Weber argue that a grammar is ontologically complete only if it is possible to map each of the concepts of the representation model to the grammar constructs [16] (see *(1)* and *(4)* in Fig. 1). They also argue that a grammar is ontologically clear if there is no construct overload, redundancy, or excess. Construct overload exists when one specific construct of the grammar can be mapped to two or more concepts of the representation model (see *(2)* in Fig. 1). Construct redundancy exists when one specific concept of the representation model can be mapped to two or more constructs of the grammar (see *(3)* in Fig. 1). Finally, there is construct excess when the grammar includes constructs that cannot be mapped to any of the concepts of the representation model (see *(5)* in Fig. 1).

In addition, Wand and Weber propose two types of mappings between the representation model and the language being evaluated: 1) the representation mapping; and 2) the interpretation mapping [16] (see Fig. 1). The former describes how each of the ontological concepts can be represented by the constructs of the language being evaluated. The latter describes what each of the language constructs represents in the real world. Wand and Weber maintain that these two types of mappings should be used for assessing the completeness of a language.

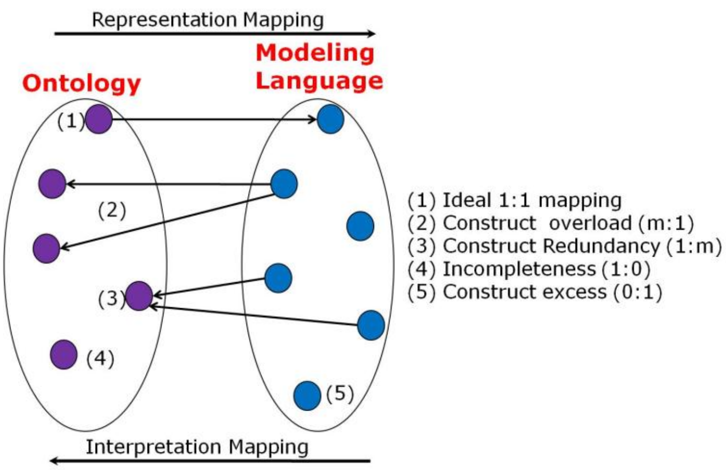

Fig. 1. Elements of a representational analysis (adapted from Wand and Weber [16])

This representation model is called the Bunge-Wand-Weber (BWW) representation model [25, 26], and has been considered "the most popular reference ontology used for representational analyses" today [27]. Over the years, the BWW representation model has been used to evaluate BPMN [28], EPC [25], UML [26], and many other languages used for BPM [28].

In today's literature, it is possible to find other, similar ontology proposals, but most of them have not achieved the popularity of the BWW representation model for the representational analysis of BPM languages [2, 17], either in theoretical publications or in empirical studies [29]. Moreover, the BWW representation model is based on sound and well-established theory [23, 24], and it has been designed for the study of any type of information system [16, 18, 22]. For these reasons, the BWW representation model has been chosen for this research.

3 Research Methodology

Although popular, representational analysis has also been subject to criticisms related to its lack of understandability, objectivism, and formalism [17, 27]. To address some of these criticisms, Green *et al*. [27] have proposed: 1) to produce the mappings with the aid of the meta-models of the ontology and the languages being compared; 2) to involve more than one researcher in the mapping process, where each researcher would produce a draft of the mapping; and 3) to require multiple iterations leading to a consensus of the mapping results among the various researchers.

In addition, Gehlert and Esswein [17] have proposed to address some of these weaknesses by: 1) performing language comparisons based on the same set of ontological concepts; 2) specifying the version of the languages being compared; 3) being explicit about whether the result of a mapping is an equivalence, a similarity, or a difference; and 4) providing the induced criteria to infer a similarity.

All these improvement recommendations have been taken into consideration and included in our research methodology (see Fig. 2). For each research activity, two researchers were involved, and they were required to conduct multiple iterations followed by reviews until a consensus was reached. The BPMN representational analysis of Recker *et al*. [30, 31] was used, which means that the same versions of the BWW representation model and BPMN were used in all the research activities presented here. It has not been necessary to provide any similarity criteria, since all the representation mappings used or performed during this research considered either an equivalence or a difference between the constructs being compared [27]. The specifications of the Qualigram language are textual [11], so the research team had to produce a meta-model of this language before performing its representational analysis. The meta-model was validated with the members of the technical staff of Globalliance who developed the Qualigram toolset.

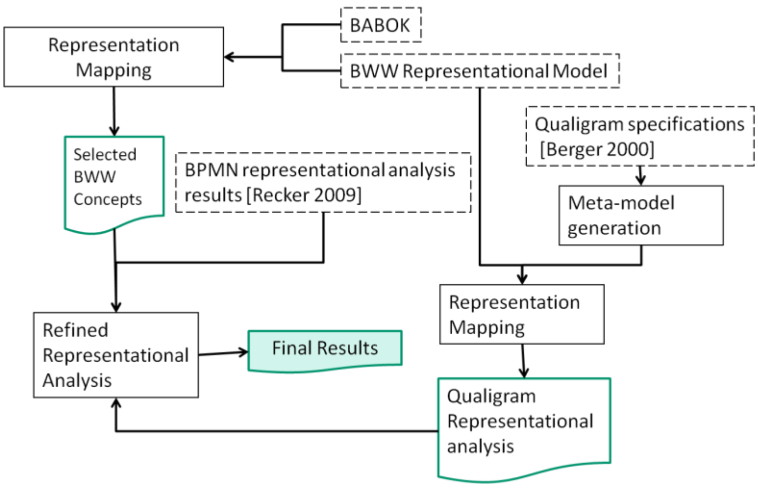

Fig. 2. Research methodology

The first main research activity (see Fig. 2) consisted of performing a representation mapping of the BABOK (*i.e.* the domain to be represented) as the reference for the mapping, and the BWW representation model for the grammar to be evaluated. This mapping allowed us to select a set of BWW constructs which, according to the BABOK, represent some relevant software requirements elicitation concepts, thus addressing the first research question.

The second representation mapping performed in this research evaluated the capability of the Qualigram constructs to represent the concepts of the BWW representation model. This step was not necessary for BPMN, as the results of the BPMN representational analysis performed by Recked *et al*. were used instead [30, 31] for this case study.

The last step of our analysis activity was to compare the two representational analyses, Qualigram's and BPMN's, with the BWW set of constructs revealed during the first step. These comparisons allowed us to answer the following question: How well do Qualigram and BPMN represent the BABOK's software requirements elicitation concepts? Thus, the second research question is addressed.

4 Review of the Languages Selected

Ko *et al*. [32] have categorized many of the most popular BPM languages according to proposed characteristics and functionalities, reflecting four typical phases of a business process management project: process design, system configuration, process implementation, and diagnosis [33]. They propose four categories of BPM languages: 1) graphical languages (*e.g.* BPMN); 2) execution languages (*e.g.* BPEL); 3) interchange languages (*e.g.* XPDL); and 4) diagnosis languages (*e.g.* BPQL). Ko *et al*. also devised an algorithm to help researchers and practitioners categorize other BPM languages they might use. Of these four categories of BPM languages, this paper focuses only on the graphical languages category, because this is typically the category of BPM language that will allow a stakeholder to represent and communicate the business processes in diagrammatic form.

Then, from the BPM graphical languages, this paper reviews and uses only two of them: BPMN and Qualigram. BPMN has already been categorized by Ko *et al*. as a graphical language. How can we categorize a Visio-based BPM language? Following the algorithm proposed by Ko *et al*., we can conclude that Qualigram is also a graphical language. In the subsections below, a high-level introduction to these two BPM languages is presented.

4.1 BPMN

The Business Process Modeling Notation (BPMN) is currently an Object Management Group (OMG) standard [10]. It was initially developed by the Business Process Management Initiative (BPMI), the BPMN 1.0 specification having been released in 2004 [34]. Later, in 2005, there was a merger between BPMI and OMG, and the latter adopted BPMN.

BPMN was designed with the aim of providing a unified language to be used by both IT and management stakeholders. It claims to be easy to understand, but at the same time having a formal basis [30, 34, 35]. To achieve this goal, the standard includes a basic set of constructs called the "Business Process Diagram (BPD) Core Element Set" (Core Set), complemented by a more complete set, "BPD Extended Set" (Extended Set). The first set is intended for documentation and communication purposes, and the second set for developing more detailed models that are appropriate for the implementation and analysis of business processes. According to recent publications, the adoption rate of BPMN is increasing in industry [5, 30].

BPMN is a modeling language rich in modeling constructs for representing various types of control flow and events. Rosemann *et al*. [2] have presented a study of how the various BPM languages have evolved to become more complete over time, using representational analysis as a basis for measuring completeness. Their results show that BPMN is the most complete of all the BPM graphical languages studied in [2].

BPMN has a high degree of expressiveness, but at the same time is highly complex [28]. According to a recent study [36], of all the modeling constructs offered by BPMN, a typical BPM created in industry uses only nine. The selection of the nine constructs varies in an arbitrary way. However, only four modeling constructs were always observed in all the BPMs studied, and some of the BPMN modeling constructs have never been used in practice.

In summary, BPMN is selected in our study because: 1) its popularity is growing; 2) it is considered as a standard by the OMG; and 3) it has a high degree of completeness, according to the representation model. However, evidence shows that BPMN is highly complex and that management stakeholders often look for simpler features in a BPM language [37]. To address this concern, a management-oriented BPM language is presented in the next subsection.

4.2 Qualigram

Qualigram is a management-oriented modeling language intended for the documentation and communication of business processes. Qualigram's modeling tool, developed by Globalliance, is currently based on Microsoft's Visio, which has been identified as the most popular tool for BPM used in industry [4, 23].

Qualigram proposes three levels of abstraction for representing processes. The top level of abstraction (the strategic level) models the core business processes and their main objectives at a high level, aiming to represent why the organization needs to perform the business processes modeled, and where to go from the organization's strategic point of view. The intermediate level of abstraction (the tactical level) models procedures, and aims to represent who is responsible for what activity in the organization, and what is accomplished, describing how to achieve the objectives of the organization. Finally, the lowest level of abstraction (the operational level) models the work instructions, aiming to represent how somebody in the organization performs a specific activity and what he uses to do so. This level also deals with the control of some specific tasks. "A process is constituted by a set of procedures; a procedure is constituted by a set of work instructions; and an instruction is constituted by a set of elementary operations" [11]. These concepts are depicted in Fig. 3.

Qualigram was designed to satisfy the requirements of the standards of the International Organization for Standardization for describing business processes (ISO 9001) [38]. Moreover, Qualigram's levels of abstraction fit with the traditional strategic, tactical, and operational levels of managerial activities found in an organization. A recent case study [37] shows that the use of such a hierarchy for establishing the levels of abstraction in a BPM initiative might be well accepted by an organization's various stakeholders. In particular, the case study showed that Qualigram's strategic level is found to be valuable for introducing the business processes to customers and to the new personnel of an organization.

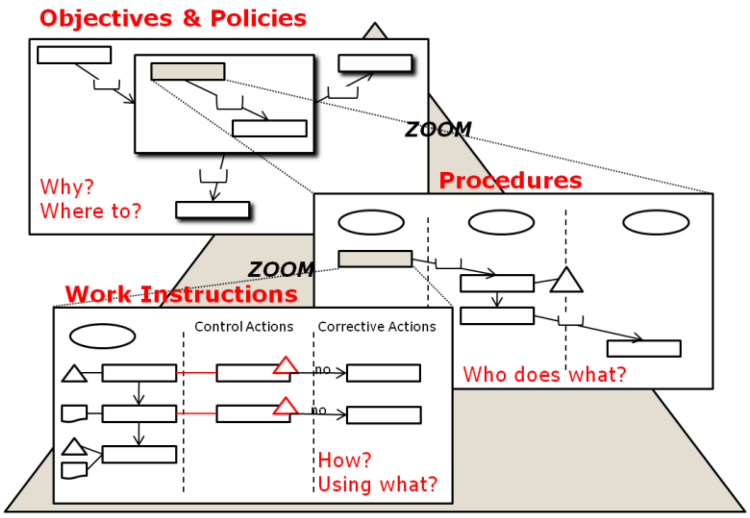

Fig. 3. Qualigram pyramid (adapted from Berger and Guillard [11])

Another characteristic of Qualigram is its simplicity. The modeling constructs for each level of abstraction are based on a set of four basic concepts, along with their corresponding graphical forms: 1) action; 2) entity; 3) tool; and 4) information (see Fig. 4). Variations of the *action* form are used to represent processes, procedures, work instructions, and elementary operations. Variations of the *entity* form are used to represent roles (internal and external), units, and external entities. The *tool* form is used to represent any kind of physical tool or equipment, as well as any kind of document produced or used by an action. The *information* form is used to represent the input and output flows of information between the various elements modeled. Qualigram's simplicity makes it clear enough to be understood by any stakeholder in the organization [11].

To summarize, the Qualigram language has been selected for this case study because: 1) it is management oriented; 2) its tool is based on Microsoft's Visio, which is currently the most popular in industry; 3) its simplicity; 4) its ISO-based approach; and 5) the presence of a hierarchy of abstraction levels.

5 Results of the Analysis

This section presents the results of the representation mapping of the BABOK, the outcome of which is the subset of constructs from the BWW representation model that have been selected for describing the relevant concepts of software requirements elicitation. The results of the representation mappings of Qualigram and BPMN are presented next using the selected subset of the representation model as a reference. Finally, some conclusions derived from the analysis of the results obtained are presented.

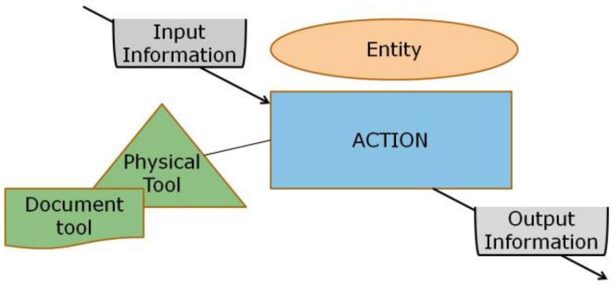

Fig. 4. Graphical forms of the Qualigram notation (adapted from Berger and Guillard [11])

5.1 A BABOK Insight into the Representation Model

The version of the representation model used for this study is the same as that used by Recker *et al*. [31]. The basic construct represented is a *thing*, as in this sentence from Wand and Weber [16]: "The world is made up of things." Recker *et al*. have grouped the BWW constructs into four clusters: "1) things, including properties and types of things; 2) states assumed by things; 3) events and transformations occurring on things; and 4) systems structured around things." The definitions of the constructs used in the BWW representation model can be found in the Green and Rosemann article [25].

In our case study, two independent researchers were asked to review the BABOK to identify the concepts relevant for software requirements elicitation. Several iterations were necessary to reach a consensus. Then, the two researchers were asked to classify the concepts into two groups: 1) those that should be represented in a BPM; and 2) those that should be supported by a BPM tool (*e.g*. the attributes and priorities of the requirements). Again, it required several iterations to reach a consensus. From the two resulting sets of concepts (*i.e*. those attributed to a BPM and those attributed to a BPM tool), the former was taken as the reference for mapping with the representation model. The final list of the BABOK concepts chosen is presented[1] in the left-hand column of Table 1 shown below. Finally, two researchers were asked to independently perform the representation mapping with the BABOK concepts. Once again, several iterations were needed to reach a consensus.

5.2 Representation Mappings of BPMN and Qualigram

Based on the results presented in Table 1, the research team elaborated a new and reduced list of BWW constructs. The resulting constructs are considered those that

[1] There are some exceptional concepts that were selected and do not appear in Table 1. The rationale for this is provided in the analysis conclusions.

should be able to describe the relevant software requirements concepts found in the BABOK. The list of final BWW constructs chosen appears in the column "BWW constructs" in Table 2 below.

Table 1. Representation mapping between selected BABOK concepts and the BWW representation model

BABOK concepts	BWW constructs
Requirement structure	Level structure
Stakeholders, users, and roles	Thing
Classes	Thing, Class
Objects	Thing
Business domain	System
Sub-domain	Subsystem
Boundaries for business domain and sub-domains	Lawful state space
External events	Event
Internal events	Event
Scheduled events	Event
Rules	State law
Business contraints	State law
Triggers	Event
Outcomes	Transformation
Responsibilities	State law
Interactions between roles and stakeholders	Acts on, Coupling
High-level requirements	Subsystem
Specific stakeholder requirements	Subsystem
Levels of abstraction: from high to low	System decomposition, level structure
Actions or Tasks	Transformation
Activities (may be broken into tasks)	Transformation
Dependencies between tasks and activities (action flows)	Lawful transformation
Milestones	Event
Process	Transformation
Subprocess	Transformation
Behavioral characteristics of a solution (features and functions)	Conceivable state space
Business units, departments	Thing
Interactions people-solution-system; IT responses	Conceivable event space, lawful event space
Constraints	State law

From the BPMN representation mapping performed by Recker *et al*. [31], only those results that correspond to our set of BWW constructs have been selected for this case study. In the same way, we present the results of the representation mapping of Qualigram that correspond to the same set of BWW constructs. For performing Qualigram's representation mapping, two researchers independently elaborated a draft of the mapping, and needed several meetings before they could reach a consensus. Qualigram's representation mapping was subsequently improved after discussing the results with the various mapping groups involved with the research team. Table 2 presents the resulting representation mappings found for BPMN and Qualigram.

Table 2. Representation mappings of BPMN and Qualigram based on a subset of the BWW representation model

Cluster	BWW constructs	BPMN	Qualigram
Things including properties and types of things	Thing	√	√
	Class	√	
States assumed by things	Conceivable state space		
	State law		
	Lawful state space		
Events and transformations occurring on things	Event	√	√
	Conceivable event space		
	Lawful event space		
	Transformation	√	√
	Lawful transformation	√	√
	Acts on	√	√
	Coupling	√	√
Systems structured around things	System	√	√
	Subsystem	√	√
	System decomposition	√	
	Level structure	√	√

5.3 Discussion of the Case Study Results

The following requirements elicitation concepts found in the BABOK could not be mapped to the BWW representation model: objectives; information managed by the solution; task ID number; relationships between objectives, requirements, stakeholders, solution, and deliverables; and metrics.

In spite of the fact that some of the unmapped BABOK concepts can be represented by the grammar of some commercial BPM languages (*e.g.* Qualigram's grammar is capable of representing objectives), it cannot be concluded, at this time, that the BWW representation model is incomplete in terms of describing the domain of software requirements elicitation. For instance, it appears that there is another point of view that can be considered, which is the possibility that some of the unmapped BABOK concepts should be included in the list of concepts attributed to a typical BPM tool. Therefore, further research, including empirical work, is needed to formulate a conclusion.

In terms of the clarity of the selected subset of BWW constructs for representing software requirements concepts (not an objective of this paper, but an interesting finding of this case study nonetheless), it can be observed from Table 1 that there is both overload and redundancy in the set of BWW constructs. The overload findings are discussed below.

The most overloaded BWW constructs revealed in this case study are: thing, event, transformation, and state law. Concerning the possible overload of *thing*, this finding is inconclusive and further research is needed to arrive at any conclusion. Recker *et al*. [30, 31] have stated that BPMN presents a number of redundant constructs (*i.e.* lane and pool) to represent things. They argue that this might cause confusion among modelers. Their empirical study confirmed that some confusion does result when those BPMN constructs are used. However, this confusion might originate from other factors, such as: 1) the apparent similarity between the constructs *lane* and *pool*; and 2) the poor definitions of these two constructs. The latter has been confirmed as a potential cause of confusion by the empirical results of Recker *et al*. Our case study reveals that Qualigram has a richer – and well differentiated – set of constructs than those of BPMN for representing different types of things. The initial findings of empirical research seem to suggest that there is some use in having such a specialization for things. Concerning the findings about the overload of *event* and *transformation*, these coincide with the observed results already published by previous theoretical and empirical studies that evaluated various BPM languages using the BWW representation model [2, 28]. These previous results have suggested a potential need to specialize the BWW constructs *event* and *transformation*, since various BPM languages have been designed with specialized constructs to represent these two BWW constructs, and some participants in the empirical studies have confirmed the usefulness of such specialization. Finally, concerning the overload of *state law* an interesting result has been found. Note in the results shown in Table 2 that this BWW construct is not represented either by BPMN or by Qualigram. Furthermore, recent studies have shown that most of the popular BPM languages lack a construct that represents a state law [2, 28]. This has been observed, and confirmed through empirical research, as a potential difficulty for modelers attempting to represent business rules in a BPM [2, 29, 30]. As a consequence, several researchers are currently working on the representation of business rules in a BPM [39, 40].

From the results shown in Table 2, it can be concluded that neither BPMN nor Qualigram are able to represent all the BWW constructs that have been selected to describe the software requirements concepts proposed in the BABOK. Of all the graphical BPM languages, the completeness of which has been studied over the years, BPMN appears to be the most complete [2]. Then, it is plausible to argue that graphical BPM languages are not complete enough to represent all the relevant software requirements concepts. In some ways, this statement is supported by Table 2, which shows the difficulty of representing the BWW constructs *state law*, *lawful state space*, *conceivable event space*, and *lawful event space*. This means that, according to [2, 29, 30], modelers have greater difficulty representing business rules in a BPM.

It is not possible to conclude, from the results presented in Table 2, which of the two BPM languages studied (*i.e.* Qualigram or BPMN) is more complete. However, this table presents evidence that BPMN might be more complete in terms of representing software requirements concepts. In particular, BPMN has an advantage over Qualigram, in that it represents the BWW constructs *class* and *system decomposition*. However, Qualigram's rich set of constructs for representing things might compensate for its inability to represent a class. Moreover, Recker *et al*. [30, 31] found that BPMN is capable of representing *system decomposition* through the *lane* and *pool* constructs, which are greatly overloaded (they represent eight different BWW constructs). Consequently, Recker *et al*. [30, 31] have stated that modelers might experience problems trying to represent a system decomposition using either a lane or a pool. This observation was clearly supported by their published empirical results. Therefore, further empirical research will be needed to support the theoretical conclusion that BPMN is more complete than Qualigram when representing the relevant software requirements concepts found in the BABOK.

6 Conclusions, Limitations, and Future Work

The representational analysis is considered an upper-level ontology. Therefore, in order to produce more precise results, the representation model will need to be refined to better fit a given business domain. One of the main contributions of the research reported here is to provide such a refinement for the software requirements elicitation domain. The result is a subset of constructs from the BWW representation model which can be used by BPM practitioners to better select a specific language for representing software requirements in a BPM [2, 28]. The subset can also be used by industry to create or improve BPM languages to support software requirements elicitation [2, 28].

Another important contribution is the assessment of the capability of two BPM languages to represent the relevant software requirements concepts. One of the BPM languages evaluated is management oriented (*i.e.* Qualigram), while the other is more IT oriented (*i.e.* BPMN). It was concluded that the difference between the representation capabilities of these two BPM languages is rather small, in terms of software requirements representation. We have also concluded that it is very likely that there is no graphical BPM language capable of representing all the BWW constructs selected for the description of software requirements.

The results of this article also support previous findings by other research groups, in particular: 1) the findings of Rosemann *et al*. [2] about the potential need to specialize the BWW constructs *transformation* and *event*; and 2) the claim of several authors [2, 29, 39, 40] that there is a need to provide BPM languages with the capability of representing business rules.

The potential shortcomings of the representational analysis, as identified in the literature, have been reported, and the recommendations suggested by Green *et al*. [27] and Gehlert and Esswein [17] to reduce the impact of these shortcomings have been adopted in the work reported here.

Further empirical research is needed to assess: 1) the potential incompleteness of the BWW representation model to describe the relevant software requirements concepts found in the BABOK; 2) the impact of an overloaded *thing* BWW construct; and 3) the differences in the representation capabilities of BPMN and Qualigram, in terms of the *class* and *system decomposition* BWW constructs.

Finally, software requirements elicitation can be performed by various types of professionals, including software engineers [15]. Therefore, this study should be complemented with a similar analysis based on the guides to the bodies of knowledge of other related professions, including the Guide to the Software Engineering Body of Knowledge (SWEBOK) [15].

Acknowledgments. Our thanks to Linda Kang and Tarik Ben Jillali for their support in the execution of this study.

References

1. McDonald, M., Aron, D.: Reimagining IT: The 2011 CIO Agenda. Gartner (2011)
2. Rosemann, M., Green, P., Indulska, M., Recker, J.C.: Using ontology for the representational analysis of process modelling techniques. International Journal of Business Process Integration and Management 4, 251-265 (2009)
3. Monsalve, C., April, A., Abran, A.: Representing Unique Stakeholder Perspectives in BPM Notations. In: 8th ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2010, pp. 42-49. IEEE Computer Society, Washington, DC, USA (2010)
4. Mili, H., Leshob, A., Lefebvre, E., Lévesque, G., El-Boussaidi, G.: Towards a Methodology for Representing and Classifying Business Processes. E-Technologies: Innovation in an Open World, 4th International Conference, MCETECH 2009, vol. 26, pp. 196-211. Springer Berlin Heidelberg, Ottawa (2009)
5. Harmon, P., Wolf, C.: The State of Business Process Management 2010. Business Process Trends (2010)
6. Desel, J.: Process modelling using Petri nets. Process-Aware Information Systems-Bridging People and Software through Process Technology, pp. 147-176. Wiley, Chichester, Hoboken, N.J. (2005)
7. IEEE: IEEE standard for conceptual modeling language syntax and semantics for IDEF1X97 (IDEFobject). (1998)
8. Russell, N., van der Aalst, W., Ter Hofstede, A., Wohed, P.: On the suitability of UML 2.0 activity diagrams for business process modelling. In: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling pp. 95-104. Australian Computer Society, Inc. Darlinghurst, Australia, Darlinghurst, Australia, Australia (2006)
9. Scheer, A., Thomas, O., Adam, O.: Process Modeling Using Event-driven Process Chains. Process-aware Information Systems: Bridging People and Software through Process Technology pp. 119-145. Wiley-Interscience, Hoboken, New Jersey (2005)
10. OMG: OMG Business Process Model and Notation (BPMN), Version 1.2. Object Management Group (2009)
11. Berger, C., Guillard, S.: La rédaction graphique des procédures : démarche et techniques de description des processus. Association Française de Normalisation, AFNOR, Paris (2000)
12. Heinrich, C.M., Christian, K., Daniela Esberger, L.M.: Business process modeling and requirements modeling. In: First International Conference on the Digital Society, ICDS'07, pp. 8. IEEE Computer Society, Washington, DC, USA (2007)
13. Mayr, H.C., Kop, C., Esberger, D.: Business Process Modeling and Requirements Modeling. In: First International Conference on the Digital Society, ICDS '07, pp. 8-8. IEEE Computer Society, Washington, DC, USA (2007)
14. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. In: Business Process Management, pp. 246-261. (2007)
15. Abran, A., Moore, J., Bourque, P., Dupuis, R.: SWEBOK: Guide to the Software Engineering Body of Knowledge 2004 Version. IEEE Computer Society, Los Alamitos, California (2004)

16. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information Systems Journal 3, 217-237 (1993)
17. Gehlert, A., Esswein, W.: Toward a formal research framework for ontological analyses. Adv. Eng. Inform. 21, 119-131 (2007)
18. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 5, 203-223 (1995)
19. Rosemann, M., Green, P.: Integrating multi-perspective views into ontological analysis. In: Proceedings of the twenty first international conference on Information systems, ICIS'00, pp. 618-627. Association for Information Systems, (2000)
20. Dreiling, A., Rosemann, M., van der Aalst, W.M.P., Sadiq, W.: From conceptual process models to running systems: A holistic approach for the configuration of enterprise system processes. Decision Support Systems 45, 189-207 (2008)
21. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D.: Selecting Empirical Methods for Software Engineering Research. Guide to Advanced Empirical Software Engineering, pp. 285-311 (2008)
22. Wand, Y., Weber, R.: An Ontological Model of an Information System. IEEE Trans. Softw. Eng. 16, 1282-1292 (1990)
23. Bunge, M.: Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World. Springer (1977)
24. Bunge, M.: Treatise on Basic Philosophy: Volume 4: Ontology II: A World of Systems. Springer (1979)
25. Green, P., Rosemann, M.: Integrated process modeling: an ontological evaluation. Information Systems 25, 73-87 (2000)
26. Opdahl, A., Henderson-Sellers, B.: Ontological Evaluation of the UML Using the Bunge–Wand–Weber Model. Software and Systems Modeling 1, 43-67 (2002)
27. Green, P., Rosemann, M., Indulska, M., Recker, J.: Improving Representational Analysis: An Example from the Enterprise Systems Interoperability Domain. In: 17th Australasian Conference on Information Systems. (2006)
28. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business Process Modeling: A Comparative Analysis. Journal of the Association for Information Systems 10, 333-363 (2009)
29. Green, P., Rosemann, M., Ipswich, A.: Ontological analysis of integrated process models: testing hypotheses. AJIS 9, (2001)
30. Recker, J., Indulska, M., Rosemann, M., Green, P.: How good is BPMN really? Insights from theory and practice. In: 14th European Conference on Information Systems: ECIS' 06. IT University of Gotteborg, (2006)
31. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do process modelling techniques get better? A comparative ontological analysis of BPMN. In: 16th Australasian Conference on Information Systems. Australasian Chapter of the Association for Information Systems (2005)
32. Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business Process Management (BPM) standards: A survey. Business Process Management Journal 15, 744 - 791 (2009)
33. Dumas, M., van der Aalst, W., Ter Hofstede, A.: Process-aware information systems: Bridging people and software through process technology. Wiley-Interscience, Hoboken, N.J. (2005)
34. White, S.: Introduction to BPMN. Business Process Trends (2004)
35. Smith, H., Fingar, P.: Business process management: the third wave. Meghan-Kiffer Press, Tampa, Fla. (2007)

36. Muehlen, M.Z., Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: CAiSE '08: 20th international conference on Advanced Information Systems Engineering, pp. 465-479. Springer-Verlag, (2008)
37. Monsalve, C., April, A., Abran, A.: Requirements Elicitation Using BPM Notations: Focusing on the Strategic Level Representation. In: 10th International Conference on Applied Computer and Applied Computational Science (ACACOS '11), In Press. WSEAS, Stevens Point, Wisconsin, USA (2011)
38. ISO: ISO 9000 Introduction and Support Package: Guidance on the Documentation Requirements of ISO 9001:2008 International Standardization Organization (2009)
39. Meersman, R., Tari, Z., Herrero, P., Goedertier, S., Vanthienen, J.: Declarative Process Modeling with Business Vocabulary and Business Rules. On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, vol. 4805, pp. 603-612. Springer Berlin / Heidelberg (2007)
40. zur Muehlen, M., Indulska, M.: Modeling languages for business processes and business rules: A representational analysis. Information Systems 35, 379-390 (2010)