Representing Unique Stakeholder Perspectives in BPM Notations

Carlos Monsalve\textsuperscript{1,2}  
\textsuperscript{1}Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Apartado 09-01-5863. Guayaquil, Ecuador  
email: monsalve@espol.edu.ec

Alain April\textsuperscript{3}, Phd, Alain Abran, Phd\textsuperscript{2}  
\textsuperscript{2}Department of Software Engineering  
École de Technologie Supérieure  
1100 rue Notre-Dame Ouest. Montréal, Canada  
email: {alain.april, alain.abran}@etsmtl.ca

Abstract—Evidence shows that proposals for new modeling notations emerge and evolution of current ones are becoming more complex, often in an attempt to satisfy the many different modeling perspectives required by each stakeholder. This paper presents a method to identify the specific notation construct requirements, at multiple levels of abstraction, which satisfy the needs of a stakeholder when performing a specific task. Initially the focus is on two different stakeholders: software engineers (SE) and business analysts (BA), and one specific software engineering activity: requirements eliciting and analysis. The specific body of knowledge of the two stakeholders (Software Engineering Book of Knowledge (SWEBOK) for the SE, and Business Analysis Body of Knowledge (BABOK) for the BA) are used to identify each stakeholder specific notation construct requirements, at multiple levels of abstraction, in order to propose a simplification of their notation and constructs set. This paper presents solution avenues to simplify business process modeling notations by identifying the specific constructs preferred by different stakeholders.

Keywords- business process modeling, software requirements, system requirements, representational analysis, levels of abstraction.

I. INTRODUCTION

At the center of Business Process Management (BPM) are the business processes and their modeling. Business process models were designed to be useful to document, communicate, or improve organizational business processes. Business process models are also used, by software engineers (SE) and business analysts (BA), for gathering the software and system requirements of information systems \cite{1-8}. Software development success is often dependent on the quality of the requirements activities \cite{9, 10}; therefore, it is crucial to successfully model the business processes. One key factor for achieving this is the active participation of all the stakeholders and sharing a common vision of business processes \cite{9, 11, 12}. Unfortunately, evidence shows that business processes are not consistently documented \cite{13} and that the different stakeholders tend to use different notations, conventions and techniques to represent them \cite{14-21}. Despite the growing complexity of business process modeling notations over the years \cite{19, 22} the goal of having one notation to satisfy all is still further ahead. Maybe this goal is unachievable? This growing complexity hinders the use of the notations and the possibility to reach a common understanding of the resulting models. It creates inefficiencies/duplications when each stakeholder uses his or her own notation and results in numerous communication problems among stakeholders \cite{10, 19, 23}. In the literature, there is a scarcity of proposed solutions to formally identify the specific set of constructs for business process modeling that satisfies the needs of a stakeholder when performing a specific task; i.e. the appropriate set of constructs for a specific perspective of business process modeling. Some authors propose new business process modeling notations, others elaborate new constructs for available business process modeling notations; some others propose to use a business process modeling notation in combination with other techniques. It is argued in this paper that, before proposing more solutions to this problem, a focused ontological analysis \cite{24} complemented by a multiple levels of abstraction analysis may be helpful in finding new avenues of solutions. This proposed approach: 1) would avoid proposing a solution that is only based on the “practical wisdom” of the authors “rather than on a scientific theory” \cite{25}; 2) the ontological analysis could help to systematically identify “the basic things in the real world” that should be modeled \cite{26}; and 3) the use of different levels of abstraction would effectively select the information to be provided to different types of users with different types of needs \cite{16, 21}. Initially the proposed focus would be on two different stakeholder perspectives: SE performing software requirements activities and BA performing system requirements activities, to show potential solutions to this complex problem. Initial analysis would be based on current requirements standards, and the guides to the bodies of knowledge used by the SE (Software Engineering Book of Knowledge or SWEBOK) and the BA (Business Analysis Book of Knowledge or BABOK). For the multiple levels of abstraction analysis three management-oriented approaches would also be considered: the Anthony’s model \cite{27}, the Berger’s approach \cite{21}, and the ISO standardized guidelines for process description \cite{28}.

The structure of this paper is as follows. Section II summarizes some basic concepts related to BPM that will help the reader understand the specific research objectives and the relevance of the topic for the industry. Section III introduces some basic requirements definitions from the perspective of both SE and BA. Section IV summarizes related work in the literature. Section V introduces the theoretical principles of an ontological analysis (using multiple levels of abstraction). Section VI presents the proposed approach. Finally, section VII concludes with a review of the contributions of this research, its limitations and future work.

II. BPM AND BUSINESS PROCESS MODELING

Early \cite{29-31} and recent \cite{20, 32} publications recognize BPM as a promising domain to bring business processes efficiencies into organizations. Many BPM methodologies, notations and tools proposing systematic analysis, design, monitoring and improvement of business processes have arisen
during the last decade. Not only is there growing academic enthusiasm about these topics, but vendors and consultants are also proposing BPM solutions to address the opportunities of this growing market. Recent industrial studies show that 93% of organizations see BPM as crucial for their organizations [33], and 26% of organizations are considering BPM in the near future to be used as a strategic component of their organizations [13]. Both studies also show a growing trend in adopting some sort of BPM within organizations. Moreover, a recent Gartner study [34] shows that business process improvement is still the number one “business expectation of IT” in 2009.

At the center of BPM are the business processes. Many definitions of business process can be found in the literature [3, 16, 35-38]: each definition varies depending on the viewpoint of the author, and depending on the perspective and the focus of the publication. From these definitions, a business process involves many elements. To be effective, a business process should aim at a goal. It typically includes a series of structured activities that transform inputs into outputs bringing some value to a customer or to the market. An activity might be triggered by an internal or external event, and it is executed by actors (employees) playing specific roles and using resources of the organization. A business process often crosses the functional boundaries of a specific corporate function often covering organizations end-to-end. As a result, business-relevant objects are transformed and value can be assessed.

Business processes are often informal and part of an employees’ experience and competencies. It has been discovered, over the years, that business processes need to be represented formally for many reasons. It may be required to document them, understand them, share the understanding with others, automate them, or try to improve them. To represent a business process, business process models are often preferred to textual descriptions. A business process model is an abstraction of the real business process [16]: it represents only those elements that the modeler considers important for a specific perspective. Business process models are also used, by SE and BA, for gathering the software and system requirements of information systems. Since a model corresponds to the point of view of the modeler, then it is plausible that SE and BA may require different abstractions of the business processes for their own specific perspective.

Modeling business processes that can be successfully shared requires the active participation of all the stakeholders, thereby improving the communication among all of them. The next paragraphs describe the current obstacles to successfully achieving the unification of business process perspectives.

A recent survey [13], shows that 51% of the time, the business processes are occasionally consistent with the information systems designed to support them. Only 2% are assessed as very consistent. Considering that 24% of the time business process management systems have been acquired by the organizations to support the business processes, this is even more significant.

Many authors report on the difficulty of choosing a single modeling notation that allows for the effective communication and participation of all the stakeholders during the requirements definition activities [9, 15, 16]. It has been also reported [14-21] that different stakeholders ask for different perspectives of the business processes being modeled: to address this need, for different notations, particular conventions and techniques have been designed. Having many notations can cause miscommunications, rework and can also be a cause of software engineering project delays, costs overruns and failure.

Other authors refer to current modeling notations as highly complex, in an attempt to satisfy the different modeling purposes required by different stakeholders [19, 22]. This complexity has been reported as one of the key reasons why a modeling notation might not produce effective models [10]; this presumption has been corroborated empirically [19, 23]. In spite of their complexity, the most popular current BPM notations lack the constructs to appropriately represent all the different requirements of an information system [18, 39-41].

Solutions to this problem have to provide the means for a consistent way of modeling the business processes and the requirements to both SE and BA. Ideally, the solution should be simple and should not significantly increase the complexity of the BPM notations, thereby allowing business process models to be easily understood by different stakeholders.

III. STAKEHOLDER PERSPECTIVES AND BODY OF KNOWLEDGE

This study focuses on two different stakeholders. This section of the paper presents their unique perspective on requirements eliciting and analysis.

The SWEBOK [9] is the guide to the body of knowledge of SE. It defines a requirement as “a property that must be exhibited in order to solve some real-world problem”, and makes a clear difference between “system requirements” and “software requirements”. System requirements involve not only software components but also other components such as hardware, firmware, and people. Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem and “are derived from system requirements”. According to the SWEBOK, SE mainly deal with the software requirements specifications; they are “concerned with the elicitation, analysis, specification, and validation of software requirements”. The more general system requirements are recognized as a responsibility of other professionals.

The BABOK [42] is the guide to the body of knowledge of BA. It adopts a more general view of requirements. BA are concerned with the understanding of “the structure, policies, and operations of an organization”. Hence, a requirement is defined by the BABOK as: 1) “A condition or capability needed by a stakeholder to solve a problem”; 2) “A condition or capability that must be met … by a solution … to satisfy a … formally imposed document”; and 3) ”A documented representation of a condition or capability as in 1) or 2)”. Since not all of the requirements are found in an explicit way, an important objective for BA is to document and communicate them by means that are “understood by all stakeholders”.

According to the SWEBOK, functional requirements “describe the functions that the software is to execute” and non-functional requirements “act to constrain the solution”. The SWEBOK also points out the importance of understanding the business processes of the organization during the requirements activities, something that is considered a complex task.

Conceptual modeling is considered by the SWEBOK and many authors as one of the main techniques for eliciting requirements (to depict scenarios), as well as for requirements analysis (to model and understand the problem, its context and
and optional ways for achieving one main goal. The goal-oriented models are intended to depict all feasible business processes than with their control flow or behavior. The goal-oriented models are more concerned with the goals of the strategic-rationale model of the BABOK [40]. This newer proposal is based on the strategic-dependency model (initially called actor-dependency model) and the strategic-rationale model. Their proposal is that the external entities are also shown. The business process, their goals, and the main dependencies among them, are modeled. From this high-level model, an i* model is generated with the particularity that the interfaces of the external entities are also shown. The i* model includes the messages to be transmitted/received to/from the external entities. The model is then translated into FT language, and the T-Tool is used for a set of verifications. Finally, a skeleton code is generated from the FT description. Kazhamiakin and Pistore claim that this method allows an early verification of the requirements and the business processes (before the final coding).

B. UML Based Approaches

Other approaches use UML as a foundation. The following authors also try to include the capability of representing goals in a business process model. Eriksson and Penker [6] using the standard extensions mechanisms of UML proposed a modeling approach consisting of four different views they refer to as approach consisting of four different views they refer to as

A. i* Based Approaches

According to the SWEBOK, one key source of software requirements are the goals (or objectives) of the software to be developed. From a BPM perspective, these goals are related to the goals of the underlying business processes. Some authors have suggested that one of the weaknesses of current business process modeling notations is their scarce support for representing goals [39, 40, 46, 51].

For instance, the early work of Yu and Mylopoulos [45, 46] proposed a new framework for modeling business processes, the i* framework, which includes two models: the strategic-dependency model (initially called actor-dependency model) and the strategic-rationale model. Their proposal is that business process models typically represent only the what? (e.g. what activity has to be executed by a role?); however, to represent the non-functional requirements the business process models also have to answer the why? (e.g. why does the role needs to execute this activity?), and also the what-if? (e.g. what if this activity is eliminated?).

Recently, they proposed a notation for developing goal-oriented models [40]. This newer proposal is based on the strategic-rationale model of the i* framework. The goal-oriented models are more concerned with the goals of the business processes than with their control flow or behavior. The goal-oriented models are intended to depict all feasible and optional ways for achieving one main goal. The goal-oriented models also include the representation of functional goals (also known as hard goals) and qualitative goals (also known as soft goals), as well as a punctuation notation for describing the control flow and the impact of the functional goals over the soft goals. This last characteristic is useful for the dynamic configuration of the business processes to fit the users and customers requirements; i.e. business process variability management. This proposal includes a method for going from the preliminary goal models to code generation. Prototype tools for supporting the business process variability, and the generation of the corresponding code are available.

The proposed notation is relatively easy to understand from the goals perspective; however, the control flow notation is not intuitive, and the roles involved in the business process are not represented. It is not clear, in the publications, if the proposed method has been validated.

Another i* based proposal, is the Tropos software development methodology, which is a requirements-oriented methodology [54]. Decreus and Poels [47] combine the i* strategical-rationale model with the formal specification language of Tropos (Formal Tropos or FT), and the Semantic Business Process Management (SBPM) ontology. They propose a method for going from the requirements elicitation to coding the associated business processes in Business Process Modeling Ontology (BPMO) which is a superset of BPMN and Event-driven Process Chain (EPC). The goals are first elicited using the i* model. Decreus and Poels propose that, for a detailed requirements analysis, a more formal description is needed. Thus, they translate the graphical model into FT language. This language has an analysis, verification and validation tool called the T-Tool, which is used for analyzing the requirements expressed in FT. Then, Decreus and Poels use SBPM to enrich the description in FT. They argue that this step eases the translation of the FT description into BPMO, which is the last step of their proposed method. No evidence of an empirical validation of the proposed method is provided.

Finally, Kazhamiakin and Pistore [48] propose a framework also based on Tropos, for going from a high-level requirements model to skeleton coding. This proposal is oriented towards the requirements orchestration and choreography for organizations that might interconnect their business processes to the ones of other organizations. First, the different entities (internal and external) that will interact in the business process, their goals, and the main dependencies among them, are modeled. From this high-level model, an i* model is generated with the particularity that the interfaces of the external entities are also shown. The i* model includes the messages to be transmitted/received to/from the external entities. The model is then translated into FT language, and the T-Tool is used for a set of verifications. Finally, a skeleton code is generated from the FT description. Kazhamiakin and Pistore claim that this method allows an early verification of the requirements and the business processes (before the final coding).
“Eriksson-Penker Business Extensions”. The views not only included the traditional aspects represented by a business process model (behavioral and process views), but also the functional structure of an organization (structural view) and the vision view. The latter, the vision view, was designed for depicting the goals of the organization and of its business processes. Some vendors use the Ericksomn-Penker extensions in their products and services [55, 56], and an open source project also uses them [57].

The List and Kotherr [5] proposal creates another UML profile extending the meta-classes actor, property and class from the UML meta-model. The specifications for each of the created stereotypes are provided, as well as an example of their use. The profile includes extensions for representing goals, quantitative and qualitative measures, and the relationships of the business processes with their owners and the customers, and a classification of the business processes as core, management and support business processes. There is no evidence of empirical validation of this profile.

Vasconcelos et al. [52] propose a three-tier UML profile for modeling business processes; it includes the strategic goals associated to them, and the information system components that support the business processes. The top tier, business strategy, models the goals and it is inspired in the Balanced Score Card approach [58]; thus, goals are classified as strategic and operational, and each of these classes are sub-classified as qualitative and quantitative. When modeling a goal it is suggested to describe its specific perspective: financial, customer, internal business processes, or learning and growth. The business process tier is the intermediate layer; its modeling is inspired in the value chain model [59]. Business processes are classified as core (inbound, operations, outbound, and sales) and support business processes (management, human resources, technology, and procurement). Resources used or produced by the business processes are also modeled as part of the middle tier. Finally, the information systems tier models the “functional building blocks of the system” (extended components). The authors state that this notation is currently used in industrial projects.

C. BPMN Based Approaches

Pavlovski and Zou [41] propose to extend BPMN to enhance it with the capabilities of representing the “operational behavior and the associated process constraints”. They argue that representing these characteristics at an early stage of the software development process will allow the software development team to determine all the detailed non-functional requirements. They propose two extensions to the current BPMN notation. One, named “operating condition” for representing the business process constraints; and a second one, named “control case”, for describing the risks associated to the constraints and the mechanisms to reduce their impact. The former one, operating condition, is graphically attached to any activity that presents a constraint. The control case is considered as optional, more likely to be used at a lower level of abstraction, and besides the graphical symbol suggested to represent it; it includes a text table with the description of the associated constraint, its risks, and means to reduce the impact. No publication could be identified indicating that these extensions have been validated empirically.

De la Vara, Sánchez, and Pastor [18] combine BPMN and a goal-oriented notation called MAP to propose a method for requirements analysis. They are also concerned with business process variability management. BPMN is used for modeling the business processes, and MAP is used for modeling the goals and strategies that lead to fulfill the users’ requirements. Initial BPMN models (as-is) are updated by the results of the analysis of the MAP model to get the to-be business process models. The BPMN notation is extended with a set of three labels that can be attached to different elements of the BPMN models to indicate if the elements are not part of the information system, are executed by a human actor, or are under the control of the information system. For every activity to be included in the information system a text table that describes the requirements associated to the activity has to be created. The method has been empirically validated with a software development company. End-users indicated that the combination of the two notations is a better approach that using typical UML class diagrams. However, not all the software development team agreed that the approach is worth using.

D. Approaches Based on Other Modeling Notations

Mayr, Kop and Esberger [7] argue that “the same basic modeling notions [of business process modeling] should be used” for requirements modeling. The basic modeling notions are: 1) tasks are executed in a workflow; 2) tasks are executed by roles; 3) resources are used by tasks; 4) tasks have pre and post conditions. The use of an inter-lingua language (KCPM) is proposed considering that stakeholders prefer to use natural language to specify their requirements. The KCPM language and its use for depicting the business processes and their requirements are described. Finally, the mapping of the KCPM generated models into available business process modeling notations is explained. The authors claim that KCPM has been empirically tested in one case study, and that the models were well accepted by the stakeholders.

E. Methodological Approaches

Kueng and Kawalek [51] point out the importance of considering goals when modeling business processes. They propose a methodology for effectively modeling business processes (activities, roles and objects) having as a starting point the goals of the software to be developed. As Yu and Mylopoulos [46], Kueng and Kawalek believe that modeling business processes should consider the whys?. They also propose to use the requirements derived from their methodology to evaluate business process models; however, they do not provide a solution for the graphical representation of the goals and software requirements.

Business process models have also been proposed as an effective mechanism for identifying system requirements from a software acquirer point of view [52]. Demirors, Gencel, and Tarhan argued that for contracting the acquisition of “software intensive system”, it is not enough to consider the customer needs, but it is imperative to clearly “understand the concept, the domain as a whole, the technology to be utilized and technical and management constraints”; furthermore, they argue that the process of going from the concept to the detailed system requirements might be done using the “notations and tools developed primarily for business process reengineering”. The authors propose a methodology to be followed by means of describing a software acquisition case. The methodology follows four stages. The first one defines the concept of the
system to be acquired, then the as-is business processes are modeled and analyzed; EPC is used among other modeling notations. The third stage consists of modeling the to-be business processes. Finally, the system requirements are defined. The methodology also includes a quality assurance activity, which performs a verification and validation of all the requirements, looking for their consistency and traceability to the modeled business processes.

Finally, Pichler and Rumetschofer [53] describe the importance of using adequate visualizing tools for describing the requirements and the business processes along the life cycle of a software development project. Their work explains the experience of using different notations, tools and techniques, ranging from post-it papers to professional requirements tools, in a three years project.

V. THEORETICAL PRINCIPLES OF THE PROPOSED METHOD

The previous section provided an overview of the different proposals considered by their authors as a contribution to model the software and system requirements with a business process orientation. It is argued here that, before proposing solutions, a focused ontological analysis with multiple levels of abstraction is necessary. The ontological analysis is performed based on a theory of the representational capabilities of a modeling notation [26, 60, 61]. For the multiple levels of abstraction analysis the Anthony’s model [27] and the Berger’s approach [21] will be considered as the theoretical foundation. The underlying theoretical principles of the proposed approach are described in the next subsections.

A. An Ontology Based Theory of Representation

Ontology studies the real world, its structure, and the relations between the things that conform to the real world [25, 26, 62]. Since the proposed method in this article aims to identify the required constructs of a business process-modeling notation to satisfy the modeling needs of a stakeholder for performing a specific task, then an ontological basis is necessary. Wand and Weber developed a set of ontological models [26, 61] that initially were focused on identifying the basic concepts that allow describing any kind of information system, its structure, and its behavior. From these models, Wand and Weber have mentioned that the “representational model” is useful to evaluate the expressiveness of any analysis and design grammar [60]. Basically, the representational model is used to determine if the evaluated grammar is complete (it has all the constructs needed to represent the real world) and clear (it has the sufficient constructs to allow an unambiguous interpretation of the generated models). Wand and Weber argue that a grammar is ontologically complete if it is possible to map each of the concepts of the representational model into the grammar constructs. A grammar is ontologically clear if there is no construct overload, redundancy and excess. Construct overload exists when one specific construct of the evaluated grammar can be mapped into two or more concepts of the representational model. Construct redundancy exists when one specific concept of the representational model can be mapped into two or more constructs of the evaluated grammar. Finally, there is construct excess when the evaluated grammar includes constructs that are not possible to be mapped to any of the concepts of the representational model. This representational model has been identified in the literature as the Bunge-Wand-Weber (BWW) representational model [3, 22, 24, 43, 63, 64], and has been considered as “the most popular reference ontology used for representational analyses” [63]. The BWW representational model has been used to evaluate BPMN [22], EPC [3], UML [64], among others notations used for modeling business processes [22]. For these reasons, the BWW representational model is adopted for the ontological analysis component of the method proposed in this article.

B. Foundations for Multiple Levels of Abstraction Approach

Curtis, Kellner and Ovner [16] have argued that a business process model “must support multiple levels of abstraction to serve the various needs”. The use of different levels of abstraction helps to select the effective information to be provided to different types of users with different types of needs [21]. From a management point of view, information systems must support, and not push, the activities and decisions taken within an organization [20, 27]. Therefore, the design of an information system should consider, in a holistic way, the needs of the different managerial activities of the organization [27, 65]. One popular approach for classifying the managerial activities is described by Anthony’s model [27, 65] which contemplates three levels of abstraction: strategic planning, management control, and operational control. Each type of managerial activity has enough particular properties that demand different details of information, and different support from the information systems. Nowadays it is possible to find several kinds of organizational structures that do not follow the traditional organizational pyramid. Even then, it can be argued that organizations host actors with different levels of information necessities that respond to the different activities being performed (e.g. planning, tactical, operations) at a specific moment of time. Even more, it is possible to find in the literature recent research work proposing modeling and design solutions trying to match the multiple levels of abstraction demanded by the organization [50, 66-68].

Berger and Guillard [21] have already developed a business process modeling method with multiple levels of abstraction. Their method is based on the results of an extensive international research project, as part of the European Strategic Program for Research and Development in Information Technology (ESPRIT) [69, 70]. The levels of abstraction of this method follow principles similar to those expressed by Anthony’s model. Berger and Guillard argue that the International Organization for Standardization (ISO) standards reflect the three levels of processes and documentation found in organizations (strategy, organization, and operation); therefore, their modeling method proposes three levels of abstraction. The top level models the business processes; the intermediate level deals with the procedures; and the lowest level models the work instructions. The close relationship between business process modeling and the ISO 9000 standards [71, 72] is well documented in the literature [13, 15, 21, 29, 30, 71, 73]. Moreover, there is evidence that organizations with an ISO 9000 familiarity are more likely to better exploit BPM initiatives [13, 29]. For these reasons, Anthony’s model, Berger and Guillard method, and the related ISO standards (ISO 9000 [71, 72] and ISO/IEC TR 24774 [28]) are considered as inputs for the multiple levels of abstraction analysis component of the method proposed in this article.
VI. IDENTIFYING THE CONSTRUCTS FOR THE REPRESENTATION OF MULTIPLE PERSPECTIVES

It has been discussed how current modeling notations tend to be highly complex (in an attempt to improve expressiveness) to satisfy the different modeling purposes required by different stakeholders. It has also been presented that there is empirical evidence that the stakeholders prefer a “careful selection of subsets of a modeling method can significantly reduce its complexity” of use and understanding [19]. An ontological analysis could help identifying the concepts and constructs needed to represent the real world, and the ontology-based representational analysis could particularly apply to the study of notation grammars. To obtain relevant results of the representational analysis, it is necessary to understand the needs of the potential users and to conceive propositions that might be of interest for targeted audiences [24, 25].

Different authors have suggested extending the usefulness of the representational analysis by incorporating novel approaches and complementing the analysis with other theories [22, 24, 25]. Rosemann and Green [24] proposed the extension of the representational analysis by integrating the characteristics of the user and the purpose of use of the business process model into the analysis. Their work also described how to use the proposed extension for developing a specific ontology for activity-based costing. Recker, Rosemann, Green and Indulska [25] argued that it is possible to produce relevant propositions as the result of a representational analysis if different independent variables that influence the act of modeling (purpose, modeler experience, modeler role, tool, conventions) are taken into consideration. They proposed to extend the representational analysis and empirically study the impact of the resulting propositions on predefined dependent variables. As an example they proposed the study of the impact of the expressiveness of a modeling notation on the continued acceptance of the modeling notation.

Inspired by these research proposals, this paper presents a novel approach to extend the representational analysis using multiple levels of abstraction, taking into consideration the analysis of: 1) formal specifications of the activities performed by the stakeholders; and, 2) the body of knowledge (including international standards) that regulates these activities. Fig. 1 depicts a general view of the proposed approach.

First, a selection analysis of the representational model is performed to generate an inventory of the modeling concepts that appropriately represent the predefined activity. For the selection analysis, the formal specifications of the activity and the related international standards are used as inputs. The selected modeling concepts serve as an input for the multiple levels of abstraction analysis. The final result is a set of modeling concepts, for a specific activity, at the different levels of abstraction.

The proposal will be initially validated by focusing on two different perspectives: SE performing software requirements activities and BA performing system requirements activities. Therefore, we need to adapt the method shown in Fig. 1 to produce and validate a set of modeling constructs for these specific perspectives.

First, a set of the BWW representational model concepts are selected taking into consideration the SWEBOK, the BABOK, and the relevant international standards. The international standards associated with these perspectives are: 1) The IEEE recommended practice for software requirements specifications [74]; 2) The IEEE guide for developing system requirements specifications [75]; and 3) The IEEE standard adoption of the international standard ISO/IEC 12119 [76]. The selected BWW concepts at the different levels of abstraction serve as a basis for performing a focused representational analysis of a predefined business process modeling notation, resulting in the set of constructs of the modeling notation that theoretically are capable of effectively representing the intended perspectives. These sets of modeling constructs have to be further validated. Two different cases of study will be conducted to validate the results. The adapted approach is shown in Fig. 2

![Figure 1. Method for selecting specific modeling concepts](image)

Figure 1. Method for selecting specific modeling concepts

![Figure 2. Method for identifying a set of modeling constructs for software and systems requirements](image)

Figure 2. Method for identifying a set of modeling constructs for software and systems requirements

VIII. CONCLUSIONS AND FUTURE WORK

A new approach for identifying the specific modeling notation constructs which satisfy the needs of a stakeholder when performing a specific task has been presented. This is a novel research proposal that will complement the representational analysis. It will use multiple levels of abstraction analysis while considering international standards. How the proposed approach can be used to identify a set of appropriate modeling constructs has been illustrated. This article presents a comprehensive review of the topic describing different solution trends for supporting the representation of goals and requirements in a business process model. In future...
work the different modeling proposals found in the literature will be analyzed and evaluated using the BWW modeling concepts.

REFERENCES


[57] StarUML, "StarUML - The Open Source UML/MDA Platform."


