

## Extracting Business Vocabularies from Business Process Models: SBVR and BPMN Standards-based Approach

Tomas Skersys, Kestutis Kapocius, Rimantas Butleris, and Tomas Danikauskas

Kaunas University of Technology, Center of Information Systems Design Technologies,  
Kaunas University of Technology, Department of Information Systems,  
Studentu str. 50-313a, Kaunas, Lithuania  
{tomas.skersys, kestutis.kapocius, rimantas.butleris, tomas.danikauskas}@ktu.lt

**Abstract.** Approaches for the analysis and specification of business vocabularies and rules are relevant topics in both Business Process Management and Information Systems Development disciplines. However, in common practice of Information Systems Development, the Business modeling activities still are of mostly empiric nature. In this paper, aspects of the approach for semi-automatic extraction of business vocabularies (BV) from business process models (BPM) are presented. The approach is based on novel business modeling-level OMG standards “Business Process Model and Notation” (BPMN) and “Semantics for Business Vocabularies and Business Rules” (SBVR), thus contributing to OMG’s vision of Model-Driven Architecture (MDA) and to model-driven development in general. The discussed extraction approach is evaluated against fully-automatic BPMN BPM  $\rightarrow$  SBVR BV transformation that has been developed in parallel to the presented work.

**Keywords:** SBVR, BPMN, business vocabulary, business process model, model-to-model transformation.

### 1. Introduction

Recent trends in the areas of Business Process Management (BPMgmt), Information Systems Development (ISD) and Business Rules Management (BRMgmt) show that these three should not be treated as competing but rather as complementary and equally important disciplines. Business vocabularies and business rules (BV&BR) can contribute greatly to intra- and inter-organizational communication, and other business knowledge exchange and transformations-oriented activities [14], [15], [31].

To support its vision of Model-Driven Architecture (MDA [18]), Object Management Group (OMG) has also contributed to the standardization of business modeling discipline (in the context of ISD, but also BPMgmt) by providing such business modeling-level standards as “Business Process Model and Notation” (BPMN [16]) and “Semantics for Business Vocabularies and Rules” (SBVR [19]). Nevertheless, integration among different interrelated concepts (aspects) of the whole Business model itself remains quite loose and empiric. None of the current OMG developments define how these standards interrelate and link to each other. Any process-related concepts are out of scope in SBVR specification; at the same time, BPMN has very poor support for the concepts related to business vocabularies and business rules (BV&BR). Indeed, today,

one of the main concerns of the professionals working in the areas of BPMgmt and ISD is the lack of efficient, standards-based approaches that allow the development of business process models and conceptual IS design models synchronized with formalized, well-structured BV&BR specifications.

We have attempted to address this issue in an ongoing three yearlong project titled “Integration of Business Processes and Business Rules on the Base of Business Semantics”. Within it, we have been defining relations between BPMN, UML and SBVR standards, at the same time investigating different techniques for the implementation of individual solutions. The scope of this paper is limited to one aspect of bi-directional  $BPM \leftrightarrow BV \& BR$  integration, i.e. semi-automatic one-way extraction of SBVR business vocabularies from BPMN business process models ( $BPM \rightarrow BV$ ).

We build on our previous work on this subject ([26]) by further refining the approach and by attempting to answer the following research question: can SBVR business vocabulary be extracted from BPMN business process models using semi-automatic approach and how efficient is such extraction compared to the fully automatic  $BPM \rightarrow BV$  transformation implemented using dedicated transformation languages (DTLs).

The paper is organized in five sections. First, state-of-the art in the analyzed field is overviewed and argumentation behind the choice of SBVR and BPMN is given. In Section 3, we cover the issue of choice between different levels of extraction as well as take a look at the available model-to-model (M2M) transformation implementation techniques. The approach itself, including the generalized transformation algorithm, is overviewed in Section 4. The section ends with the look at the key features of the actual implementation of the developed semi-automatic  $BPM \rightarrow BV$  extraction approach. Finally, in Section 5, we present key results of experimental comparison of the described semi-automatic business vocabulary extraction approach with the automatic one that is based on the use of transformation languages. Conclusions are drawn in Section 6.

## 2. Stat-of-the-Art and Argumentation

### 2.1. Related Work

As it was already mentioned, the work presented here was the direct continuation of efforts described by us in [26]. It has to be noted that since the first results of  $BPM \rightarrow BV$  transformation research have been published, important changes to the approach have been made. Therefore, before tackling the main research question of this particular paper, refined and updated view at the key approach aspects, including prototype implementation, had to be overviewed. We also devote special attention to the issue of implementation of the approach, including available automation levels and actual implementation techniques, because understanding them is critical in order to grasp the context of the performed experimental evaluation.

Before speaking of the progress made by other researchers in the area of  $BPM \leftrightarrow BV \& BR$  interrelation, it is important to distinguish between integration and transformation. Integration, although involving mappings between the concept types of different models, deals with relating two or more metamodels by either merging them, enriching one of them with the required elements, or by introducing supplementary

mapping data structure [29]. Models that are integrated this way remain cohesive at all times. Transformation, on the other hand, ensures cohesiveness only at the time of performing the transformation (term “generation” may also be applied). After that, both models can be edited independently. Furthermore, transformation does not assure bipolar connections between the elements of different models as would be the case with integration. Here, one must analyze transformation opportunities from both directions separately (and develop two separate transformations), because certain transformations that are possible from Element A to Element B, may not be possible the other way around or may require a different algorithm.

As was mentioned in the Introduction, the scope of this paper is the extraction of business vocabularies from the business process models. Analysis of related efforts indicates that majority of such research deals with the issues of the opposite transformations, i.e. the development of business process models from BV&BR specifications. Apparently, at this time only Malik and Bajwa [9] have proposed an initial framework of a BPMN BPM  $\rightarrow$  BV&BR generation approach. According to their findings, no other work in this specific area of research has been done, which correlates with our experience.

Speaking of research carried out in the field of BV&BR $\rightarrow$ BPM interrelation, it was also highly relevant to us in helping understand this complex area of research as well as providing insights and ideas that let to our contribution. Automatic transformations from SBVR-based BV&BR to business process models in BPMN have been the subject of work by Raj et al. [23] and Steen et al. [32]. The aim there has been facilitation of correct modeling of activity sequences in business process models. “IF <condition> THEN <action>” template is used by the authors to express rules. Although it makes sense when modeling sequences of activities, the expressiveness of SBVR does not get fully utilized. Steen et al. [32] also addressed business process model optimization issues. At around the same time, a similar proposal was developed within the OPAALS Network of Excellence which, among other things, delivered a solution for automatic code structure and workflow generation from natural language specifications [2]. Six scenarios of model-to-model transformation for the workflow and code structure generation out of SBVR specifications were presented. The scenario describing the development of BPMN/XPDL models using SBVR business vocabularies and rules was of the most interest to us. However, only the very basic transformations have been described in the available documentation.

EM-BrA2CE (Enterprise Modeling using Business Rules, Agents, Activities, Concepts and Events) project [4] was another notable initiative analyzing the issue of business processes’ specification using business vocabularies and rules. Here, the key result was the declarative process modeling framework unifying vocabulary and execution model. SBVR has been used as a vocabulary specification standard, while the execution model was presented as a Colored Petri Net (CP-Net). Authors emphasized the declarative nature of their framework, which meant that the functional and operation perspectives were not supported. Furthermore, no business process models visualization tools (in the form of a graphical modeling language and an appropriate interpreter) were included in the proposal.

Considerable amount of research is available on the topics of SBVR specifications application during the development of other IT artifacts, namely, web services [10], SQL queries [12] or SPARQL queries [33]. Progress has also been made in such areas as SBVR rules extraction from the software code [21], and natural language text [5]

[11]. Although these developments are only loosely related to the aims of the work presented in this paper, they are interesting as confirmations of the relevance of SBVR application research.

The above analysis has revealed that the issue of extraction of SBVR specifications from the existing business process models has not been properly researched. The popular opinion that vocabularies (in a form of fact models, conceptual data models or textual specifications) appear before business processes are specified may be one of the reasons for that. We argue that in the real world there are cases when business process models exist without any formal business vocabularies. As soon as such models have to be used in larger ISD projects, the need for a vocabulary may arise. Not to mention the fact that BPM $\rightarrow$ BV transformation is an essential step towards the overall BPMN-SBVR integration.

## 2.2. BPMN and SBVR: Definitions and Argumentation

**Business Process Model and Notation (BPMN).** The main diagram of BPMN is a Business Process Diagram (BPD), which is a representation of the whole or a part of Business Process Model (BPM). There are four categories of core elements in BPM: Swim lanes, Flow Objects, Connecting Objects and Artifacts. These core elements are the basic source of knowledge in the process of SBVR BV extraction.

Analysis of Bunge-Wand-Weber ontology [24] showed that compared to other business process modeling languages BPMN covers the largest amount of real world concepts and it is well-understood and accepted by business experts [3], [13]. Judging from these results one can conclude that BPMN models provide the most of the formalized business knowledge, from which BV (and later, BR) may be extracted. Wahl and Sindre [35] also indicated that BPMN is clearly understandable and well-suited for business process modeling. Compared to BPMN, software systems modeling-oriented standards, like UML and IDEF, lack sufficient semantic expressiveness; also, such languages hold a number of concepts, constructions and rules that can be treated as excessive for business process modeling needs [22].

Another argumentation to use BPMN is the fact that this standard formally integrates into OMG's Model Driven Architecture (MDA [18]), which is widely accepted standard by CASE tools developers and other R&D communities. From MDA perspective, BPMN BPM is a part of a Business model.

**Semantics for Business Vocabularies and Rules (SBVR).** Business vocabulary (BV) is defined to contain a set of "specialized terms and definitions of concepts that a given organization or community uses in their talking and writing in the course of doing business" [19]. According to so called business rules "mantra" (followed from "Business Rules Manifesto" [20]), business rules (BR) are built on facts and facts are built on terms. Terms (*Noun Concepts*) and Facts (*Verb Concepts*) are the ones that form the basis of any business vocabulary. In its turn, one cannot properly specify and manage business rules without having the support of a proper business vocabulary.

Among other approaches of BR specification using natural language expressions [25], [30], [34], arguably, the most significant one is the "Semantics of Business Vocabulary and Business Rules" (SBVR) standard [19]. The vision of SBVR is to express business knowledge in a controlled natural language, which would be unambiguous and understandable to business and IT people as well as computer systems.

The complete SBVR business vocabulary metamodel contains over a hundred of concepts defining various aspects of a BV. Due to the scope of this paper, this large structure can be scaled down to the following core elements:

- *Noun Concept*, which can be specialized to General Concept and Individual Concept (and also, Role, which is out of scope in this paper). General Concept is a noun concept that classifies things on the basis of their common properties. Individual concept is a noun concept that corresponds to only one object (thing).
- *Verb concept* is a concept that denotes some type of relationship between two or more noun concepts or a characteristic of the noun concept. Following the definition, verb concepts are defined using the existing noun concepts, which have been already defined in BV. A Verb Concept has a final set of specializations.

Four types of font styles with concrete formal meaning are used to represent noun concepts, verb concepts and business rules in SBVR-based Structured English or any other chosen language (e.g. Lithuanian):

- ‘*term*’ font is used to represent general concepts (object types) and roles, e.g. ‘*customer*’;
- ‘*Name*’ designates individual concepts that usually are proper nouns, e.g. ‘*Lithuania*’, ‘*IBM*’;
- ‘*verb*’ font represents a verb, a preposition, or a combination of these two, e.g. ‘*customer provides order*’;
- ‘*keyword*’ font represents linguistic symbols that are used to construct statements and definitions, e.g. ‘*each*’, ‘*It is obligatory that*’.

SBVR BV has glossary-like entries, which specify concepts having representations in the vocabulary. Each entry is for a single concept. It should be mentioned that even though the primary form of representation of SBVR specifications is structured natural language (e.g. Structured English as in [19], Structured Lithuanian or any other world language), graphical representation could also be used for various model-driven developments (see a fragment of fact diagram in Figure 3 (tag D) as an example).

As of yet, SBVR is probably the first initiative to formalize and standardize the definition of a business vocabulary suitable for wide-range practical applications. Being an OMG standard, SBVR initially had a strong support from the world’s BPMgmt community, CASE tools developers and other R&D parties. From the MDA perspective, BV and BR have to be defined at Business modeling level of MDA, i.e. in parallel with business process modeling. SBVR is fully integrated into the OMG’s MDA via MOF or Eclipse Metamodeling Framework (EMF).

### 3. Choosing Automation Level and Implementation Technique for the Extraction of SBVR BV from BPMN BPM

The problem of extraction of any kind of specific knowledge from a certain model in order to use this knowledge in developing a different type of model could, in most cases, be referred to as the model-to-model (M2M) transformation task. M2M transformations are at the very core of the MDA vision. Both BPMN and SBVR were developed as part of this vision.

### 3.1. Levels of Automation of M2M Transformation

When undertaking any M2M transformation task, one must decide on the level of transformation automation, i.e. would it be manual, semi-automatic, or fully-automatic. The following must be kept in mind when making this choice:

- Distinctive feature of a *manual* M2M transformation is that it is done exclusively by the user, who might use his own empiric knowledge (“know-how”) and/or formally defined transformation rules and algorithms to perform this task. Due to the lack of any automation, this kind of transformation is of no particular interest to us and will not be discussed any further.
- In case of a *semi-automatic* M2M transformation, the transformation algorithm requires certain degree of user interaction to complete the task. Transformations of this kind are usually implemented as *hard-coded* applications.
- *Automatic* M2M transformations are capable to perform a fully-automated transformation of source models to target models without the involvement of a user in the transformation process. Automatic transformations may be implemented as *hard-coded* applications or by using *dedicated transformation languages* (DTLs). Provided both source and target models are based on formal metamodels, the latter implementation technique is recommended.

As mentioned earlier, semi-automatic and automatic M2M transformation approaches may be implemented using different implementation techniques. The most popular techniques are hard-coding and DTL implementation, which involves the use of specialized M2M transformation engines. We will briefly discuss these techniques in the following section.

### 3.2. Implementation Techniques of M2M Transformation

In general, the choice of the implementation technique and its implementation language could be conditioned by many factors. We distinguish the two basic ones:

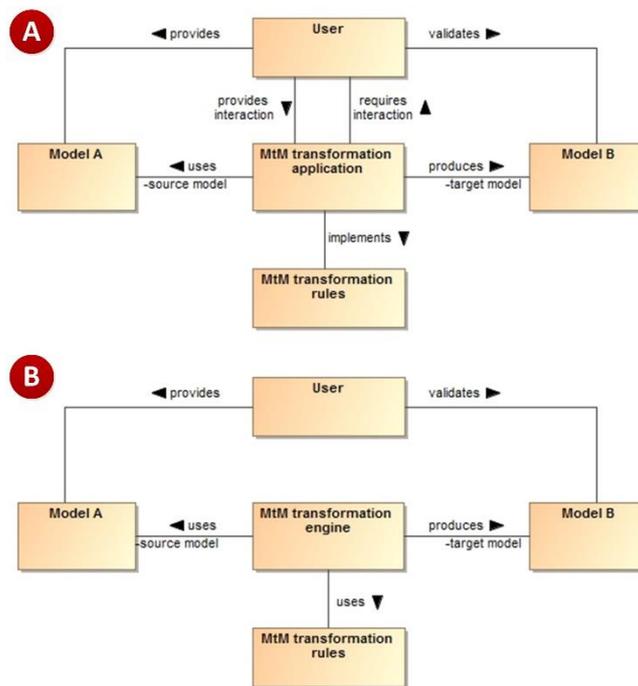
- Required architectural features of the developed system and the specifics of its implementation environment;
- Required levels of system’s flexibility and user’s interaction with the system.

*Hard-coded* technique does not require any intermediate technology to implement transformation. The transformation itself follows the fixed algorithm, which is embedded in an application (Figure 1). Due to its algorithmic nature, this technique might be considered as an optimal solution for algorithm-based approaches involving user assistants (wizards), forms and other graphical user interface tools. Wizard-assisted M2M transformation applications provide high level of flexibility and customization for the user. One of the main drawbacks of such implementation (and thus, the approach itself) lies in the management of changes at a metamodels level – these changes must be implemented by altering the source code, which might become a time and effort consuming undertaking.

The latter drawback of hard-coding technique has much less impact on the *DTL-based* technique. This technique uses dedicated transformation languages to realize M2M transformations on metamodels level (Figure 1), therefore it assures higher stability of transformation rules and higher compatibility with modeling standards as well as

CASE tools that support them. The DTL-based technique is well-suited for the fully-automated M2M transformation approaches, because DTL transformations do not provide the possibility of user interaction – in some cases this might be considered as a drawback of this technique and of fully-automatic approach as a whole. In general, approaches with high level of automation rely heavily on so called best modeling practices, which may vary depending on a modeling language used (e.g. modeling business processes with UML and BPMN).

In our ongoing research project VEPSEM (see Acknowledgements), we are investigating different BPMN-SBVR transformation approaches and implementation techniques. Next to the *semi-automatic* BPM→BV transformation approach (which is the main scope of this paper), the approach of *automatic* BPM→BV transformation and its implementation are also being researched and developed (the latter implementation is based on QVTO DTL).



**Fig. 1.** Principles of (A) hard-coded semi-automatic and (B) DTL-based automatic M2M transformation

### 3.3. Common Implementation Languages of M2M Transformation

The analysis of available options has revealed that the most prominent implementation languages for M2M transformations are ATL / QVT, and Java. The first two are dedicated for automatic M2M transformations, and Java is the option for the implementation of the semi-automatic transformation approach.

**ATL Transformation Language.** ATLAS Transformation Language – or ATL – is of a hybrid nature, allowing for both declarative and imperative constructs to be used in transformation rules [7]. Declarative format is recommended, but sometimes imperative statements are also necessary. At the basis of an ATL statement lays a module that consists of header, libraries import part, unlimited number of functions called helpers, and, finally, transformation rules.

The core part of an ATL transformation is a transformation rule. Declarative rules, also known as matched rules, consist of descriptions of source (*from* part) and target (*to* part) structures. *From* part has to include description of concept type of source metamodel as well as required constraints in OCL. In the *to* part, concept types of a target metamodel have to be specified, including operations required to assign values to the newly created concepts of the defined concept types.

When executing an ATL transformation rule, the source metamodel is analyzed looking for structure matching the description given in the rule. When such structure is found, elements of a target model are created following the logics of target structure of the rule.

Architecturally ATL is similar to Java Virtual Machine. Prior to execution, each ATL transformation code is converted into XMI format, which is compiled into an Assembler file. The latter can be executed in ATL Virtual Machine.

**QVT Transformation Language.** QVT (Query/View/Transformation) is an OMG standard tailored for M2M transformations in the context of MOF. QVT specification includes three languages: declarative *Relations* and *Core* languages as well as imperative *Operational Mappings* language [17]. There is also support for non-standard *Black-box MOF Operation* implementations that allows for an external code and programming libraries to be used during transformations. QVT Operational Mappings language (QVTO) is of biggest interest to us, because it is the only subset of QVT that is fully supported in advanced tools like Eclipse M2M and has been documented in detail.

QVTO architecture is not defined strictly in the standard, thus allowing for some flexibility when implementing QVT transformation engines. In *MagicDraw* tool, which is our actual implementation environment, QVTO is implemented with JSR-223 script engine reusing the QVTO implementation from Eclipse M2M project.

It should be mentioned that reasonable interoperability is also possible between ATL and QVT languages. Jouault and Kurtev [6] have shown that in theory ATL programs could be executed on QVT engines, and QVT languages – on ATL virtual machine. Laarman [8] has successfully developed a compiler allowing the specification of QVTO programs and then running them in ATL environment. Because ATL and QVT can extend each other using available mechanisms, such implementations could be quite valuable.

**Java-based Transformation (hard-coded).** Quite often the DTLs overviewed above may be inefficient simply because of the lack of know-how. In such cases, writing transformations in much more popular Java programming language could serve as an alternative – even for fully-automatic M2M transformations.

Specialized frameworks like Eclipse EMF can be used to generate application programming interface (API) methods for manipulation of metamodels created in Ecore format. The source model can be viewed as a tree-like structure, so the transformation program code must enable traversing this structure from top to bottom. More specifically, the source model should be passed to the main method via its parameters, after

which it is decomposed hierarchically into smaller components that are transformed into a target – new model – by specific API methods.

Here it is important to note that the complexity of such implementation can rise dramatically, as the complexity of the source metamodel rises. But the key shortcoming of such transformation is the need to completely rewrite software code, when at least one metamodel changes, e.g. the new version is released. On the positive side, such transformations can be interactive, thus enabling greater flexibility of M2M transformations.

## 4. Semi-automatic Extraction of SBVR BV from BPMN BPM

On a conceptual level, the approach consists of two interrelated parts: a mapping matrix and an M2M transformation algorithm. In case of the extraction of SBVR business vocabularies from various source models (e.g. BPMN BPM, UML Use Case Model, UML Class Model), the mapping matrix is very dependent on a source metamodel, i.e. different matrixes must be defined for each M2M transformation. The initial BPM-to-BV transformation algorithm was discussed in [26], but since it was published we came to the realization that the core transformation algorithm is of universal nature. Therefore, the algorithm presented here is constructed in such a way that its basic steps remain stable and source model-independent throughout various M2M transformations; this is also reflected on the implementation level of the approach. Beside the approach presented in this paper, the above mentioned principle was also applied in the approach of semi-automatic extraction of SBVR business vocabularies and business rules from UML use case models [27].

### 4.1. The Algorithm

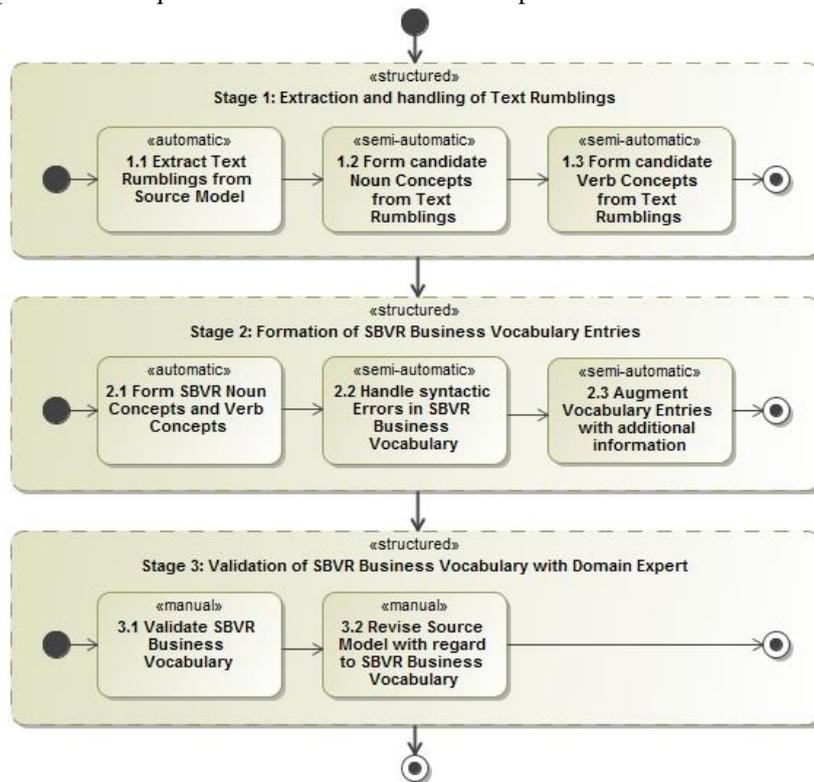
On the highest level of abstraction, the semi-automatic BV extraction algorithm is composed of three successive stages (Figure 2):

- *Stage 1: Extraction and handling of text rumblings*<sup>1</sup>. In the first step of this stage, text rumblings are extracted from the identified source model concepts. In the second and third steps, *candidate* noun concepts and verb concepts are formed from the extracted text rumblings. Here, the algorithm follows the basic principle of Business Rules “mantra”, which, once again, states that facts are built on terms – therefore, one must specify a set of noun concept before specifying verb concepts. On the implementation level, *Stage 1* represents the most volatile part of the M2M transformation application because it encapsulates the extraction rules defined in the mapping matrix. The next two stages of the algorithm remain stable throughout all implemented approaches of semi-automatic M2M transformation (e.g. [27]).

---

<sup>1</sup> A term “text rumbling” represents an unstructured piece of textual information in a problem domain. This term is derived from a term “business rumbling”, which was first introduced by B. von Halle in her paper “Back to Business Rule Basics” (in Database Programming & Design, 1994) and had more specific scope than a “text rumbling”, i.e. *business* domain.

- *Stage 2: Formation of SBVR business vocabulary entries.* SBVR BV entries are formed from the candidate concepts. After the entries are formed, automatic validation of syntactic errors of the specification can be (this is a recommended step because the *candidate* concepts may have been formed syntactically incorrect). The formed SBVR business vocabulary entries may be also augmented with additional information as needed.
- *Stage 3: Validation of SBVR business vocabulary with domain expert.* This is more or less a straight forward process of business vocabulary validation with a domain expert and consequent revision of BPMN business process model.



**Fig. 2.** The algorithm of semi-automatic SBVR BV extraction from a source model

More details of the algorithm are presented in Section 4.3, where the implementation aspect of the approach is discussed.

It is important to note that the higher the level of automation of M2M transformation, the more strictly the rules of best modeling practices have to be followed when developing a source model. Therefore, it is highly recommended that BPMN business process models are created, or, in the cases when models are already available, – refactored, in accordance to best practices prior to extraction of business vocabulary concepts. Obviously, in some cases such refactoring is not possible (the models cannot be changed due to authorship or other restrictions) or inefficient (due to time or other constraints). This is why one of the aims of this particular research was to see how does the quality of

business process models influences the business vocabulary extraction (more on that in Section 5).

Three basic recommendations have been formulated to facilitate creation or refactoring of business process models (for more detailed elaboration please see [26]):

1. *Use consistent naming scheme for each type of concepts in models.* Transformation rules (independently of their implementation) work only with the predefined naming patterns; therefore, inconsistent concept names will be most likely interpreted incorrectly by M2M transformation application.
2. *Use pools and lanes in all BPM diagrams.* The absence of these concepts in BPM will drastically limit the number of automatically extracted candidate concepts and formation of business vocabulary entries.
3. *Strive towards BPM diagrams that consist of no more than 7(+2) activities* by creating hierarchical business process models. This may be especially useful in case of semi-automated extraction as smaller diagrams are easier to analyze.

Some more details about the extraction process itself are presented in Section 4.3.

#### 4.2. The Mapping Matrix

In order to develop M2M transformation rules (both hard-coded and DTL-based) that may be passed to the transformation algorithm, a set of mappings between the concept types of corresponding metamodels have to be identified. The defined mappings are then deployed in the mapping matrix (Table 1). Table 1 presents a revised and improved version of the mapping matrix from our previous work [26].

**Table 1.** Mapping pairs of core concept types of SBVR and BPMN metamodels for BV extraction

BPMN BPM		SBVR BV	
Group	Concept Type	Noun Concept	Verb Concept
Swimlanes	Lane	A	A
	Pool	A	-
Flow Objects	Event	M	A*
	Activity	A*	A*
	Gateway	M	M
Connecting Objects	Sequence Flow	-	-
	Sequence Flow (with Condition)	M	M
	Message Flow	-	-
	Message Flow (with ref. Message)	A	-
	Association	-	-
Artifacts	Data Object	A	-
	Data Object (with State)	A	A
	Data Store	A	-
	Group	A	-
	Text Annotation	M	M

The meaning of markings “A”, “A\*” and “M” used in the matrix is as follows:

- “A” implies that a corresponding SBVR noun concept or verb concept can be *automatically* identified and extracted from one or more BPM concepts;
- “M” implies *manual (semi-automatic)* identification and extraction of a corresponding SBVR general concept or verb concept from one or more BPM concepts;
- “A\*” implies that a corresponding SBVR noun concept or verb concept can be *automatically* identified and extracted from one or more BPM concepts, if best modeling practices are applied for naming concepts in a source BPM; otherwise, “A\*” should be interpreted as an “M”.

Referring to the algorithm described in Section 4.1, during the extraction of text rumblings, markings “A”, “A\*” and “M” have the same meaning – they all mean that the marked BPM elements are used as sources of knowledge for *automatic* extraction of certain textual expressions. When all the rumblings are extracted, the defined meaning of the markings becomes important and comes into play. For example, in Table 1, the intersection of the BPMN concept type *Lane* with the SBVR concept type *Noun Concept* is marked with “A”, which means that all concepts of the type *Lane* in any BPMN BPM will be a subject for *automatic* identification and extraction of certain candidate noun concept in order to form *Noun Concept* entries in SBVR business vocabulary (e.g. ‘[delivery boy](#)’ and ‘[clerk](#)’ from the presented BPM in Figure 3).

Note that in the case of fully-automatic transformation, the transformation algorithm would not include mappings marked with “M” due to the required user-interacted decision making. Furthermore, automatic transformation depends heavily on the best modeling practices applied to a source model (mappings marked with “A\*”); though, this also influences the performance of semi-automatic transformation as well. Our experience has revealed that if the quality of the source model is known to vary significantly, automatic transformations may become quite problematic. On the other hand, the models built following a rigid set of rules would be easier to interpret automatically. This knowledge has been reflected in the developed fully-automatic transformation approach, which is used as an experimental reference here, but will not be described in detail as it is out of scope of this particular paper.

### 4.3. The Implementation

The main goal of the chosen implementation strategy was to supplement an advanced CASE tool with capability of developing and managing business process models and business vocabularies under the same working environment. CASE tool *MagicDraw* was used as a base implementation platform

On the implementation level, basic process of the proposed BPM→BV approach could be described as follows (Figure 3):

- BPMN BPM is developed and managed using *CAMEO Business Modeler*, which is an official vendor’s plug-in for a CASE tool *MagicDraw* (tag A).
- After the BPM is developed, the formation of SBVR BV can be initiated. The process starts from the extraction of text rumblings from the BPM (tag 1). Main steps of the approach, starting with the extraction of text rumblings and finishing with the formation of SBVR BV entries, are performed using a hard-coded JAVA-based semi-automatic M2M transformation tool (tag B), which is implemented as a plug-in

of *MagicDraw*. The hard-coded M2M transformation rules are based on the mapping matrix (Table 1).

- After the extraction of text rumblings, the tool guides a user through the steps of SBVR BV formation, which is based on the principles of the business rules “mantra”. Here, it all starts with the formation of candidate general concepts in the first tab window; then the next tab window provides functionality to work with candidate verb concepts; finally, candidate business rules are formed in the third tab window. Please note though, that the formation of business rules is not within the scope of this paper. In Figure 3, the second tab window of the transformation tool is opened where the candidate verb concepts are being formed (tag B); an applied font style marks the candidate general concepts that have already been identified in the previous step.

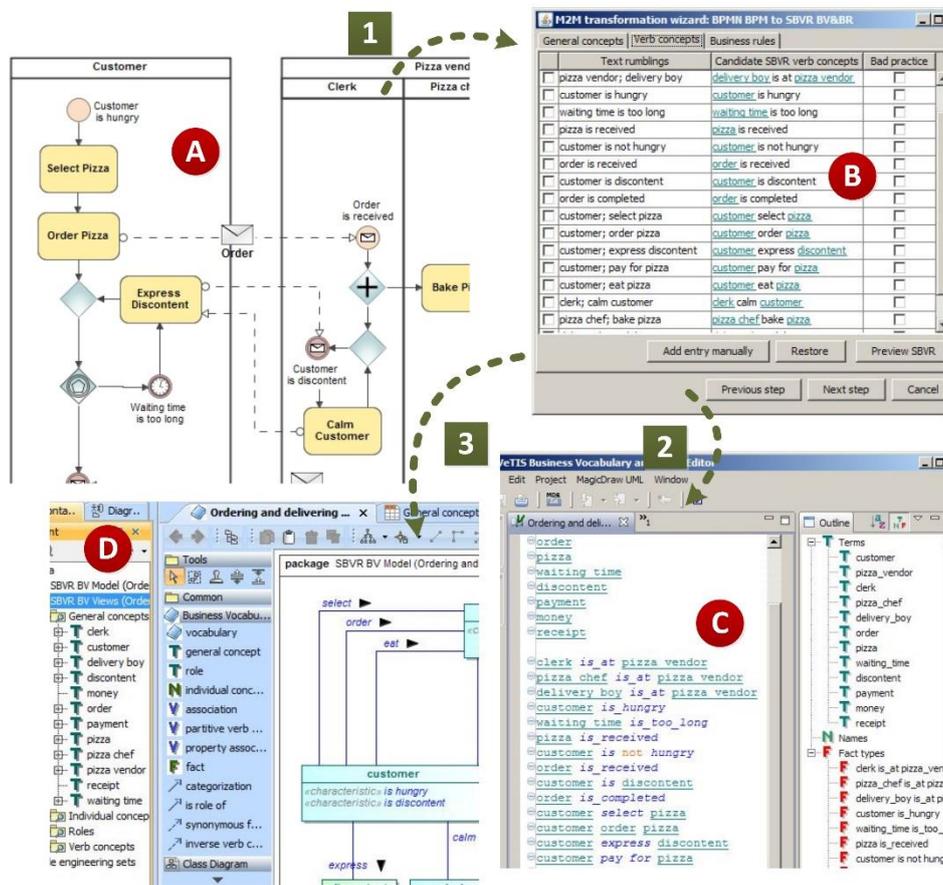


Fig. 3. Interaction of MagicDraw, BPM→BV tool (*MagicDraw* plug-in) and *VeTIS* editor (*Eclipse*)

- After the candidate concepts were formed and validated, the actual formation of SBVR BV entries is performed. The transformation tool can form SBVR business vocabulary in two output formats: (1) structured text document (tag 2), which may be interpreted by our research group’s earlier development – *VeTIS* tool [14] (tag C); (2)

SBVR model (tag 3), which is supported by *MagicDraw* (tag D). *VeTIS* tool is an SBVR editor with syntactic validation of SBVR specifications. *VeTIS* also has a *MagicDraw* plug-in, which enables the generation of UML class models from SBVR business vocabularies and business rules [14]. The second output format is SBVR model. At the moment of writing, *MagicDraw* is the only UML CASE tool known to us, which supports model-driven specification of SBVR business vocabularies – this capability was enabled after the UML Profile for SBVR and a specific DSL was developed and implemented in the CASE tool by the researchers of VEPSEM project [28].

Figure 4 presents basic elements of the implementation architecture. Note that the colored elements in Figure 4 represent the main parts that were directly related with the implementation of semi-automatic BPM→BV approach. Other elements were also mentioned in certain parts of the paper; however, they fall out of the scope of this particular research.

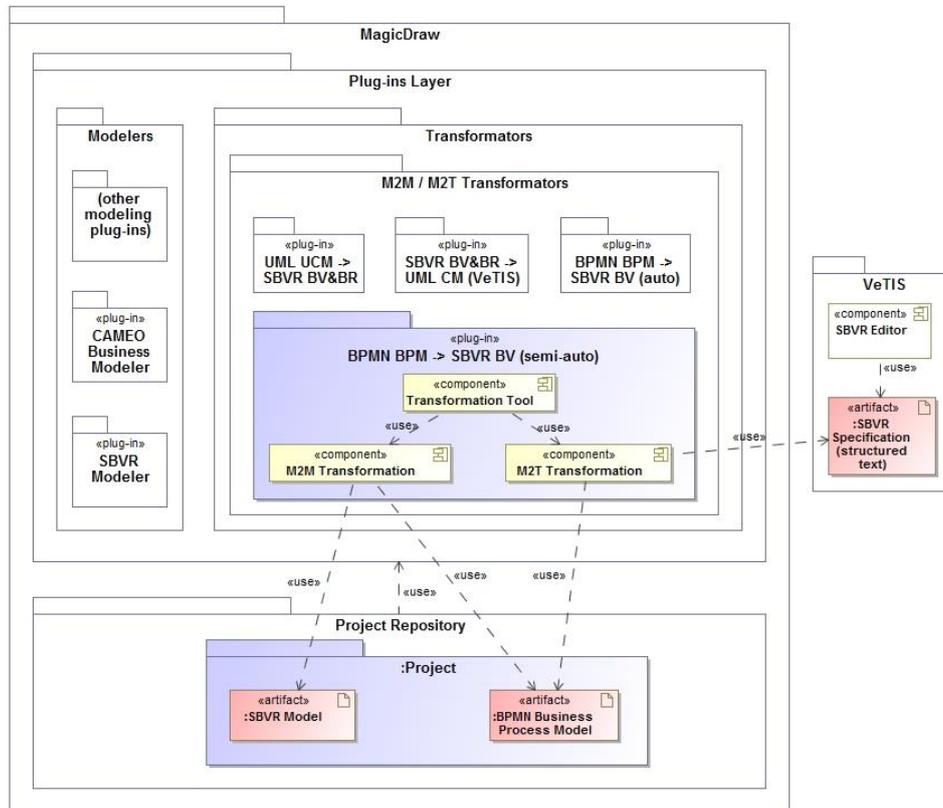


Fig. 4. Conceptual schema of the implementation architecture

## 5. Evaluation of the Approach

Experimental evaluation of the proposed semi-automatic BP-BV extraction approach consisted of the following steps:

1. Refactoring of the given BPMN BPM according to the outlined recommendations.
2. Extraction of SBVR business vocabularies from the BPMN BPM using the presented semi-automatic approach:
  - 2.1. using original business process models as a source (scenario  $S_{1.1}$ );
  - 2.2. using refactored business process models as a source ( $S_{1.2}$ ).
3. Extraction of SBVR business vocabularies from BPMN BPM using automatic DTL-based transformation:
  - 3.1. using original business process models as a source ( $S_{2.1}$ );
  - 3.2. using refactored business process models as a source ( $S_{2.2}$ ).
4. Analysis of results.

As a source for the experiment, the business process models from OMG's non-normative document "BPMN by Example" were used, namely:

- D1. Shipment Process of a Hardware Retailer ([1], p. 3),
- D2. Ordering and Delivering Pizza ([1], p. 4),
- D3. Incident Management as detailed collaboration ([1], p. 9),
- D4. Incident Management with human-driven and system-driven pools ([1], p. 11),
- D5. Nobel Prize Process ([1], p. 26),
- D6. Travel Booking ([1], p. 28),
- D7. E-Mail Voting ([1], p. 36).

The size of source models varied from small to quite large ones (see Table 2). Using official exemplary document as a source ensured that the nature and structural characteristics of each business process model varied sufficiently as each of them represented certain BPMN application areas and/or scenarios.

For the purposes of the experiment, all seven models were refactored following the formulated recommendations. An example of a refactored diagram – D2 – is given in Figure 5.

**Table 2.** The size of the experimental sample

BPMN BPM concept type	The amount of concepts in diagrams							
	D1	D2	D3	D4	D5	D6	D7	
Swimlanes	4	5	5	8	5		1	
Flow objects	Event	2	7	11	11	8	25	14
	Activity	8	9	12	17	17	20	24
	Gateway	6	2	5	4	1	6	10
Connecting objects	18	24	38	43	38	46	93	
Data objects, data stores					5		15	
Other artifacts	1		2	1	3	3	3	
Total amount of con- cepts	39	47	73	84	77	100	160	

The experiment was first carried out individually by our research group members – the main experimental group. Then, a selection of experts not involved in the develop-

ment of the approaches used also performed the extraction, but with 2-3 randomly assigned models each, and focusing on the semi-automatic extraction. This second phase was carried out aiming to evaluate the usability characteristics, namely, to see if the approach is clear enough to be used by the newcomers.

The summary of results from the experimental extraction of business vocabulary using all four scenarios is given in Table 3. Three criteria ( $C_{1,2,3}$ ) were considered and measured:

- $C_1$ : *amount of extracted noun and verb concepts*. A combined total amount of SBVR business vocabulary elements extracted from the given set of business process models. Generally the higher value – the better. As with other criteria, for the purposes of this paper, we skip concrete numbers presenting instead the key relative fuzzy evaluations that are of interest with regards to the research question formulated in the Introduction. In Table 3, value *Low* means that the extraction rate was below 65% of the maximum theoretically possible estimate, *Medium* – 65-85%, and *High* – more than 85%. Estimate *Highest* has also been used to indicate the subtle difference between results under scenarios  $S_{1,2}$  and  $S_{2,2}$ . In the case of the latter, the extraction rate was above 90%.
- $C_2$ : *time requirements*. The time it takes for an analyst to get the extraction results for all models, not counting the time it takes to find erroneous ones. Lower value is better. Again, relative generalized evaluations have been used. *High* stands for times over 1 hour, while *Medium* – below that mark. Finally, *Very low* means that the extraction was more or less instantaneous, which was the case with automatic transformations.
- $C_3$ : *percentage of erroneous extractions*. The relative amount of extracted BV elements that are redundant (duplicates, synonyms, different grammatical forms of the same concept), must be broken up / renamed / rephrased (homonyms), do not belong to the vocabulary, et al. The value must be as low as possible. The relative generalized values are *High* (more than 20% of all extracted BV elements were identified as erroneous), *Medium* (8-20% were erroneous), and *Low* (less than 8% were erroneous).

**Table 3.** Summary of experimental evaluation results achieved by the main experimental group

Approach:	Source/scenario	Evaluation criteria		
		$C_1$	$C_2$	$C_3$
Presented semi-automatic approach and hard-coded implementation	$S_{1,1}$ : Original BPMN business process models	Medium	High	Medium
	$S_{1,2}$ : Refactored BPMN business process models	Highest	Medium	Low
Automatic approach and QVT-based implementation	$S_{2,1}$ : Original BPMN business process models	Low	Very low	High
	$S_{2,2}$ : Refactored BPMN business process models	High	Very low	Low

Note that in the case of the proposed semi-automatic extraction, search for erroneous extractions is somewhat simplified by the *VeTIS* tool. It finds syntactical errors and, maybe more importantly, duplicates, automatically and presents the user with the opportunity to resolve them.

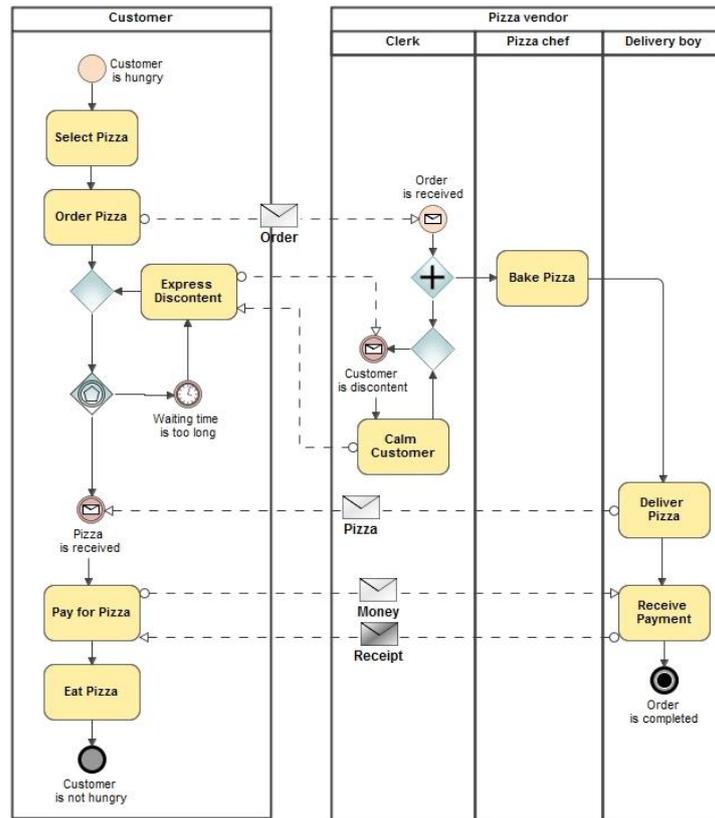


Fig. 5. Refactored BPMN business process diagram “Ordering and delivering pizza” ([1])

Based on the experimental findings given above, the following key observations were made:

- Semi-automatic extraction is less sensitive to the quality of the source BPMN BPM, whereas the success rate of automatic transformation depends heavily on how the source model meets the business process modeling rules. Obviously, it is difficult to assure that business process analysts will be willing to follow a specific business process modeling practice at all times.
- Refactoring business process models improves extraction times and increases the success rates significantly enough to be recommended as a common practice when using semi-automatic extraction.
- If the probability of erroneous results is high (criterion  $C_3$ ), additional time is likely to be spent by the analyst on correcting the extracted vocabulary. Therefore, it is essential that the tool used to work with the vocabulary is capable of finding syntactical

errors, duplicates, etc. In our case, *VeTIS* tool was of the use here, but opportunities for further improvement of vocabulary validation automation should be investigated.

The experiences of the second group of experts who got to evaluate the approach were generally positive. Actual results proved to be within the same margins as the ones achieved by the authors of this approach. However, we must note here that these experts did not have to refactor business process models, and only evaluated the vocabulary extraction process.

## 6. Conclusions

There is little doubt in the necessity of integration and transformations between BPMN and SBVR models. Technically, the preconditions for making it come true are good, and the choice of programming tools is sufficient to implement both semi- and fully-automatic transformations.

In this paper, an approach for semi-automatic extraction of SBVR business vocabularies from BPMN business process models is presented. The prototyping was based on CASE tool *MagicDraw* and proved to be promising.

The experimental findings confirmed that SBVR business vocabulary can indeed be extracted from BPMN business process models. However, the extraction efficiency and success rate depend on whether the source models meet the defined modeling practices. This is especially notable in the case of fully-automatic transformation. Here, one must acknowledge that the need to follow certain business process modeling practices may be a strong off-factor for many analysts. This shortcoming could be addressed by introducing presented modeling practices directly into CASE tools, in the form of non-critical recommendations.

Experiment also revealed that semi-automatic interactive extraction delivered generally better results in terms of the quality of the extracted vocabulary than fully-automatic transformation did. This came with no surprise because automatic extraction is less flexible due to lacking interactivity and errors can only be corrected after the actual transformation. However, using dedicated transformation language should make it easier to update the implementation in case any of the metamodels are updated. Hard-coded solution could be more difficult to update when the metamodels change.

Overall we conclude, that, when fully implemented, BPMN BPM  $\rightarrow$  SBVR BV approach would provide certain benefits, such as: faster development of well-structured, formalized business vocabularies, ready for various areas of application, including model-driven ISD; increased quality and completeness of the business model; possibility to track changes in business model and conceptual IS models. Further research could result in the development and implementation of BPM $\leftrightarrow$ BV&BR two-way synchronization approaches.

**Acknowledgements.** The work is supported by the project VP1-3.1-ŠMM-10-V-02-008 „Integration of Business Processes and Business Rules on the Base of Business Semantics (VEPSEM)" (2013-2015), which is funded by the European Social Fund (ESF).

## References

1. BPMN 2.0 by Example. OMG, Version 1.0 (non-normative), Document Number: dtc/2010-06-02. (2010). [Online]. Available: <http://www.omg.org/spec/BPMN/2.0/examples/PDF> (current December 2013)
2. Eder, R., Kurz, T., Heistracher, T.J., Filieri, A., Russo, M., et al.: D2.2 - Automatic code structure and workflow generation from natural language models, OPAALS Project. (2008)
3. Glassey, O.: A case study on process modelling - Three questions and three techniques. *Decision Support Systems*. Vol. 44, No.4, 842-853. (2008)
4. Goedertier, S., Haesen, R., Vanthienen, J.: EM-Bra2CE v0.1: A vocabulary and execution model for declarative business process modeling. FETEW Research Report KBI-0728, Katholieke Universiteit Leuven. (2007)
5. Gomez, T. H., Franco A. F. J.: Business Rules Extraction from Business Process Specifications Written in Natural Language. *Business Rules Journal*, Vol. 11, No. 7. (2010). [Online]. Available: <http://www.BRCommunity.com/a2010/b543.html> (current June 2013)
6. Jouault, F., Kurtev, I.: On the Architectural Alignment of ATL and QVT. In: *Proceedings of the 2006 ACM Symposium on Applied Computing*. ACM New York, 1188-1195. (2006)
7. Jouault, F., Kurtev, I.: Transforming Models with ATL. In: Bruel, J.-M. (ed.): *MoDELS 2005 Workshops*, 128-138. (2006)
8. Laarman, A.: Achieving QVTO & ATL Interoperability: An Experience Report on the Realization of a QVTO to ATL Compiler. In: *1st International Workshop on Model Transformation with ATL, MtATL 2009: CEUR Workshop Proceedings*, 119-133. (2009)
9. Malik, S., Bajwa, I. S.: Back to Origin: Transformation of Business Process Models to Business Rules. In: *BPM 2012 International Workshops, Revised papers*. Estonia, ISBN 978-3-642-36284-2, 611-622. (2012)
10. Marinos, A., Krause, P.: An SBVR Framework for RESTful Web Applications. In: *International Symposium on Rule Representation, Interchange and Reasoning on the Web*, Springer - Verlag Berlin, Heidelberg, 144-158. (2009)
11. Martinez-Fernandez, J. L., Gonzalez, J. C., Villena, J., Martinez, P.: A Preliminary Approach to the Automatic Extraction of Business Rules from Unrestricted Text in the Banking Industry. In: *13th International Conference on Natural Language and Information Systems*. Springer, 299-310. (2008)
12. Moschoyiannis, S., Marinos, A., Krause, P.: Generating SQL Queries from SBVR Rules. In: *International Symposium RuleML 2010, Lecture Notes in Computer Science*, vol. 6403, 128-143. (2010)
13. Muehlen, M., Indulska, M., Kamp, G.: Business process and business rule modeling languages for compliance management: a representational analysis. In: *ACM International Conference*. Vol. 334, 127-132. (2007)
14. Nemuraite, L., Skersys, T., Sukys, A., Sinkevicius, E., Ablonskis, L.: VeTIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models. In: *16th International Conference on Information and Software Technologies (IT 2010)*, Kaunas Lithuania, 377-384. (2010)
15. Nenortaite, J., Butleris, R.: Improving Business Rules Management through the Application of Adaptive Business Intelligence Technique. *Information Technology and Control*. Vol. 38, No. 1, 21-28. (2009)
16. Object Management Group: Business Process Model and Notation (BPMN) v.2.0. OMG Doc. No.: formal/2011-01-03 (2011). [Online]. Available: <http://www.omg.org/spec/BPMN/2.0> (current December 2013)
17. Object Management Group: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Version 1.1. (2011) [Online]. Available: <http://www.omg.org/spec/QVT/1.1/> (current December 2013)
18. Object Management Group: Model-Driven Architecture (MDA) v.2.0 (2003). [Online]. Available: <http://www.omg.org/mda/index.htm> (current December 2013)

19. Object Management Group: Semantics of Business Vocabulary and Business Rules (SBVR) v.1.0 (2008). [Online]. Available: <http://www.omg.org/docs/formal/08-01-02.pdf> (current December 2013)
20. Object Management Group: The Business Rules Manifesto (2003). [Online]. Available: [www.businessrulesgroup.org/brmanifesto.htm](http://www.businessrulesgroup.org/brmanifesto.htm) (current December 2013)
21. Osrof, M., Zaghal, R.: Ontology Based Business Rules Extraction Model and Algorithm (OBBREMA). In: 13th International Arab Conference on Information Technology ACIT2012, Lebanon, ISSN 1812-0857, 66-73. (2012)
22. Paiano, R., Guido, A. L., Pandurino, A.: Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering. Information Science Reference. (2009)
23. Raj, A., Prabhakar, T. V., Hendryx S.: Transformation of SBVR business design to UML models. In: 1st conference on India software engineering conference Hyderabad, India, 29-38. (2008)
24. Recker, J. C., 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, Sydney, Australia (2005)
25. Ross, R.: Principles of the Business Rule Approach. USA: Addison Wesley. (2003)
26. Skersys, T., Butleris, R., Kapocius, K., Vileiniskis, T.: An Approach for Extracting Business Vocabularies from Business Process Models. Information technology and control. Vol. 42, No. 2, 178-190. (2013)
27. Skersys, T., Danenas, P., Butleris, R.: Approach for Semi-automatic Extraction of Business Vocabularies and Rules from Use Case Diagrams. Advances in Enterprise Engineering VIII, LNCS, vol. 174, pp. 182-196 (2014)
28. Skersys, T., Pavalkis S., Nemuraite L.: Implementing Semantically Rich Business Vocabularies in CASE tools. In: 12th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2014). AIP Conference Proceedings (2014) (*to be published*)
29. Skersys, T., Tutkute, L., Butleris, R., Butkiene, R.: Extending BPMN business process model with SBVR business vocabulary and rules. Information technology and control. Vol. 41, No. 4, 356-367. (2012)
30. Skersys, T., Peciulis, V., Simutis, R.: Business rules specification using natural language-based templates: approach and implementation. In: 14th International Conference on Information and Software Technologies, Kaunas, Lithuania, 353-360. (2008)
31. Skersys, T.: Business knowledge-based generation of the system class model. Information technology and control. Vol. 37, No. 2, 145-153. (2008)
32. Steen, B., Ferreira Pires, P., Iacob, M. E.: Automatic Generation of Optimal Business Processes from Business Rules. In: 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, 117-126. (2010)
33. Sukys A., Nemuraite L., Paradauskas B., Sinkevicius, E.: SBVR Based Representation of SPARQL Queries and SWRL Rules for Analyzing Semantic Relations. In: 1st International Conference on Business Intelligence and Technology, Rome, Italy, 1-6. (2011)
34. Von Halle, B.: Business Rules Applied - Business Better Systems Using the Business Rules Approach. USA: John Wiley & Sons. (2002)
35. Wahl, T., Sindre, G.: An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. Proceedings of the CAiSE'05 Workshops. Vol. 1, 533-544. (2005)

**Tomas Skersys**, PhD, is a researcher at the Center of Information Systems Design Technologies and an Associate Professor at the Department of Information Systems (Kaunas University of Technology, Lithuania). His research interests and practical experience cover various aspects of business process management and model-driven information systems development. On these topics, he published several articles in high-rated academic journals and in a number of international conferences. He is also a co-editor of three books of international conferences published by Springer Verlag.

**Kestutis Kapocius**, PhD, is a researcher at the Center of Information Systems Design Technologies and an Associate Professor at the Department of Information Systems (Kaunas University of Technology, Lithuania). His research covers the topics of information systems modeling, business rules driven IS development, and systems usability. Since 2005, Dr. Kapocius has been involved in no less than 6 multi-year research & development projects.

**Rimantas Butleris**, PhD, is the Director of the Center of Information Systems Design Technologies and a Full Professor at the Department of Information Systems (Kaunas University of Technology, Lithuania). His main area of research is requirements engineering. During his career, he has authored or co-authored more than seventy research papers and participated in more than fifteen national and international research projects; in many of those he was a leading researcher. Up till 2013, Rimantas Butleris was the program chair or co-chair of the International Conference on Information and Software Technologies (ICIST).

**Tomas Danikauskas**, M.Sc., is a junior researcher at the Center of Information Systems Design Technologies and a Lecturer at the Department of Information Systems (Kaunas University of Technology, Lithuania). His research and development activities center around the topics of requirements engineering and information systems design. Since 2000, Tomas Danikauskas has been involved in several large scale projects, including creation and ongoing development of the Integrated Information System for Lithuanian Forestry Cadastre.

*Received: January 06, 2014; Accepted: June 27, 2014*

