



## **Modeling Business Rule Parallelism by Introducing Inclusive and Complex Gateways in Semantics of Business Vocabulary and Rules**

**Irene RHT Tangkawarow<sup>1,2</sup>**

**Riyanarto Sarno<sup>1\*</sup>**

**Daniel Siahaan<sup>1</sup>**

<sup>1</sup>*Informatics Department, Faculty of Intelligent Electrical and Informatics Technology,  
Institut Teknologi Sepuluh Nopember, Indonesia*

<sup>2</sup>*Education of Information and Communication Technology Department, Faculty of Engineering,  
Universitas Negeri Manado, Indonesia*

\* Corresponding author's Email: [riyanarto@if.its.ac.id](mailto:riyanarto@if.its.ac.id)

---

**Abstract:** The Semantics of Business Vocabulary and Rules (SBVR) standard was developed by the Object Management Group (OMG) for business purposes. SBVR is used for transformation of business vocabulary and business rules into business processes. Gateways are used for regulating the divergence and convergence of flow objects in the business process. The existing business rules in SBVR do not support all gateways in BPMN, whereas there are conditions where branching situations in business rules occur. This article introduces parallelism rules (OR rules) and complex rules to increase 50.6% usage of the existing AND rules and XOR rules in SBVR. The main contribution of this research is to introduce new formal model of inclusive gateway (OR) and complex gateway that allow parallelism and branching to be modeled using SBVR. Thus, this study increases coverage of the usage gateway in SBVR achieved 66.7%. The authors provide branching cases with various levels of complexity, i.e. nested conditions and non-free choice conditions, using the formal description of SBVR.

**Keywords:** BPMN, Business rules, Business vocabulary, Complex rules, Inclusive rules, SBVR.

---

### **1. Introduction**

The concept of business rules has been defined in many different ways. A business rule is defined as a logical statement that determines some aspects of a business in a particular situation [1]. Business process management (BPM), information system development, semantic technologies, artificial intelligence, etc. are areas where business rules are applied. Business rules can be expressed in several different forms and languages depending on the application area to become information system executable code or database triggers [1]. They can also be expressed in the executable rule language of a business rules management system (such as FICO Blaze Advisor or IBM Ilog) [2] or in natural language text.

Business rules must be specified in a language that is based on a formal model and is well structured and unambiguous to specify expressions in model

transformation, rule exchange, execution, and other similar activities. The business actors who are the actual owners of the business rules expect them to be comprehensible and easy to use.

Business people and even information system developers tend to express business rules in unstructured natural language statements ('text rumbling'), followed by a free list of standard terms and definitions [3]. In 2008, the Object Management Group (OMG) released the Semantics of Business Vocabulary and Rules (SBVR) standard [4] and revised it by adding subject transformation using XMI (XML Metadata Interchange) in 2017 [5]. The purpose of SBVR is to express business knowledge formally in a controlled natural language. SBVR can bridge the gap between machines language and human language. Humans can understand simple sentences in natural language, while a controlled natural language has the unambiguous properties of a

machine language [6]. SBVR is understandable and unambiguous to humans as well as computer systems.

In S. Arshad, I. S. Bajwa, and R. Kazmi [6] propose an approach to translating software requirements using SBVR into XML schema. In their calculation average recall is 92.50% while average precision is 94.87%. The SBVR2XML tool performed highly accurate and exact results as compared with other natural language based tools.

G. Aiello, R. Di Bernardo, M. Maggio, D. Di Bona, and G. Lo Re [8] proposed a mapping procedure for automatically translating rules formulated into a format based on natural language. Their report describes the action of formulating business rules in a natural language that is executable by using well-consolidated techniques.

Mohanani, M. and Samuel, P [9] conducted experiments to generalize SBVR using natural language as the initial data. In the experiment they used the SBeVeaR tool to verify the syntax of the business vocabulary rules. The experiment was conducted by dividing the SBVR into three levels of complexity, with ten sentences each. O. Tantan and J. Akoka [10] executed a model-to-model transformation of SBVR Structured English (SE) into BPMN. Those researches developed the idea of transforming SBVR into simple BPMN notation but did not discuss gateway notation.

Rodrigues, R. D. A., Azevedo, L. G., Revoredo, K. C. [11] propose a language-independent framework for generating natural language texts automatically from business process models using Java standard technology. The framework conducted to evaluate the generated text quality. As the result, the framework claims the quality of textual description is 86.6%.

In previous research translation from natural language to SBVR have been done [6-10]. Then, translation from BPMN to natural language has been done by Rodrigues, R. D. A., Azevedo, L. G., Revoredo, K. C. [11].

Rules in SBVR specified in two kinds, i.e. structural and operational. In structural rules use such modal operators as necessary or possible / impossible. Operational rules use such modal operators as obligatory, permitted / forbidden [13].

No previous studies provide a detailed discussion of SBVR's operational rules about situations that contain branching or forking in a business process. Branching of the object flow uses gateways in BPMN; parallel disjunction in SBVR uses inclusive gateways and exclusive gateways in BPMN.

This study aimed to find an SBVR operational rule that is suitable as a solution for inclusive gateways (OR gateways) and complex gateways,

which have not been investigated previously. In usage of gateway in several sampling process models, we found that the use of inclusive gateway (OR) and complex gateway was 11.16%. Then, the usage of the existing AND gateways and XOR gateways was 50.6%. The main contribution of this research is to introduce new formal model of inclusive gateway (OR) and complex gateway that allow parallelism and branching to be modeled using SBVR. This research fulfill branching rules for transformation from BPMN to SBVR. A formal process model was developed for their integration as well as a model that applies to the representation of special rules. Thus, the authors provide the formal description rules for gateways in BPMN using the SBVR approach.

## 2. Related work

### 2.1 Business process modeling

Business process modeling is an activity where several elements of an organization or company are represented to produce a cohesive model of the operating process [14]. Business processes can be modeled with various tools and techniques selected by the engineer or modeler [15]. The model of a business process based on standard operational procedures (SOP) can have several objectives, namely: resource planning, identifying bottlenecks, calculating time durations, and calculating costs [16]. Business process development that begins with modeling can be related to fraud detection based on a process model obtained from the event log [15 - 17].

### 2.2 Business process modeling semantics of business vocabulary and rules (SBVR)

SBVR is a so-called controlled natural language. A controlled language is an intermediary between natural language and formal language. The Business Rules Mantra is an SBVR methodology. It states that business rules are built on fact types, while the fact types are built on concept types [5]. According to A. Raj, A. Agrawal, and T. Prabhakar [18] the advantages of SBVR are its declarative nature, rule-based modeling approach, natural language representation, and backbone of formal logic. O. Chaparro, J. Aponte, F. Ortega, and A. Marcus [19] define the key concepts of BRs in SBVR: (1) business vocabulary: common business vocabulary, built from concepts, terms, and types of fact; (2) business rules (BR): statements/sentences based on types of fact that guide the business operation or business structure; (3) semantic formulation: the way of structuring the meaning rules through several logical formulations,

e.g. quantification states (each, at least, at most, etc.), logical operators (and, xor, or, if-then, etc.), and modal formulations (It is obligatory or It is necessary); (4) notation: the language for writing and expressing BR. The SBVR standard uses three reference notations, namely: SBVR Structured English, RuleSpeak, and Object-Role Modeling.

In Fig.1 shows an overview of SBVR divided into business vocabulary models and business rule models. The business vocabulary model is divided into concept types and fact types. Fact types are arranged based on concept types. The business rule model in which these rules are arranged is based on fact types.

### 2.2.1 Business vocabulary (BV)

In SBVR, the business vocabulary has concept type and fact type as the two main elements [19]. Concepts are critical terms that represent a business entity in a particular domain. There are several basic concepts types, namely noun concepts, individual concepts, and verb concepts [10].

Usually, general nouns are classified as noun concepts while proper nouns and quantified nouns are classified as individual concepts. Verb concepts can be either auxiliary verbs or action verbs, or both. A fact type is a combination of a verb concept and a noun concept. The fact type determines the relationship between different concepts in business rules in business process models: the name of the actor is represented by the noun concept and the process is represented by the action verb concept.

### 2.2.2 Business rules (BR)

In SBVR, business rules represent business logic in specific contexts. Every SBVR business rule is based on at least one fact type. SBVR rules can consist of rules of definition and rules of conduct [4]. Rules of definition, or structural rules, are used to

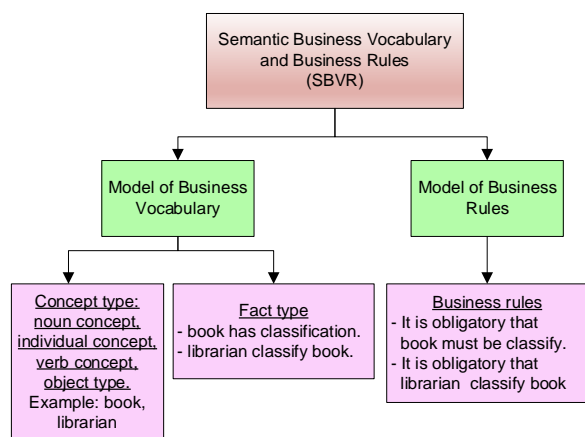


Figure. 1 SBVR methodology

Table 1. SBVR notation

| Type        | Explanation  |
|-------------|--|
| <u>term</u> | Term for a noun concept that is part of the vocabulary used or defined; for example, <u>person</u>   |
| <u>name</u> | Term for individual concepts and numerical values; for example, <u>white car</u> , <u>California</u> , <u>the number 26</u>  |
| <i>verb</i> | Term for a fact type that is usually a verb or preposition, or a combination of both; for example, <i>sort</i> , <i>has</i> , <i>determine</i>   |
| keyword     | Keywords that accompany designations or expressions; for example, each, at least, at most, obligatory.<br>In a sentence:<br>It is obligatory that ...<br>It is necessary that ...<br>It is possible that ...<br>It is permitted that ... |

define or ganizational settings. For example, It is necessary that each customer has at least one bank account. Behavior rules, or operation rules, state the entity’s behavior. For example, It is obligatory that librarian sorts book if a librarian receives a book. In Table shows an explanation of four types of SBVR notation.

### 2.3 Overview of SBVR parallelism rule

The purpose of the research by R. D. A. Rodrigues, L. G. Azevedo, and K. C. Revoredo [11] was to translate natural language specifications into SBVR business rules. The final result of the study was the NL2SBVR tool. The input is divided into three levels of complexity: simple, compound, and complex SBVR rules. This research used structured sentences and the experiment was carried out by dividing the SBVR rules into three levels of complexity, with 10 sentences each. However, only one sentence each was given in the report, so it is difficult to understand the difference with a sentence that has another type of complexity.

In T. Skersys, L. Tutkute, and R. Butleris [1] there are three primary stages for integration, namely: (1) development and integration of BP diagrams and business vocabulary; (2) augmentation of BP diagrams with business rules; (3) validation of the overall business model by domain experts. The principle of simple integration was used to integrate SBVR into business processes. The implementation used the developed VeTIS solution approach, which consists of a full-featured SBVR editor and an SBVR-to-UML transformation plug-in from the MagicDraw UML CASE tool. Unfortunately, the

report does not provide a detailed description of the process integration between SBVR expression or grammar and its transformation into the structure of BPMN.

The follow-up research by T. Skersys, K. Kapocius, R. Butleris, and T. Danikauskas [19] proposed the semi-automatic business vocabulary extraction algorithm, which consists of three steps: (1) extracting and handling 'text rumbling' (unstructured text information that is still in the problem domain); (2) establishing SBVR business vocabulary entries; and (3) validating SBVR business vocabulary with domain experts. This research conducted semi-automatic business vocabulary extraction from a business process model. It concludes that the extraction of BPMN into SBVR provides certain benefits, such as: the development of business vocabularies that are faster, well structured, formal, ready to be used in various fields of application, including information systems development (ISD) driven by the model; improving the quality and completeness of business models; the possibility to track changes in business models and conceptual information system (IS) models. Overall, this study presents an extraction method using a one-way synchronization approach.

O. C. Tantan and J. Akoka [10] conducted a model-to-model transformation from SBVR Structured English (SE) to BPMN. The main purpose of this research was to help business experts in the requirement validation stage when the business rules are stated in natural language. This research did not discuss text-to-model transformation, assuming that SBVR Structured English is consistent and complete. The document used was a text in a structured document. The research was still in the form of an idea with an explanation of the transformation of SBVR into a simple BPMN notation, which did not include parallel gateways.

G. Aiello, R. Di Bernardo, M. Maggio, D [8] proposed an automatic mapping technique to translate the rules stated in SBVR. The production rules are processed by a rule engine in Java beans, translating the SBVR rule set into JBoss Drools production rules. The main objective of this method is to overcome some of the weaknesses in the software development process, which results in inconsistencies between the identification of the domain requirements and the functionality of the applied software. An analysis of mapping techniques and case studies were carried out. This research was an important step forward in making business rules written in natural language executable by adopting well-consolidated techniques. Besides, the case studies proved the validity of this approach and made

it possible to reduce the communication gap between non-IT and IT personnel involved in developing the MyOpenGov research project's eHealth scenario. However, the SBVR grammar format used is still complicated and the case study did not cover all grammatical expressions.

The method of K. Kluza and K. Honkisz [13] was initially aimed at determining operational rules for mapping SBVR to BPMN and structural rules for transforming SBVR into Decision Model and Notation (DMN). They developed a translation algorithm that translates the vocabulary rules, structural rules, and operational rules into elements of BPMN and DMN. At the end of the study, the relationship between BPMN and DMN was described. The case study did not cover all notational elements in BPMN. This research was in the form of algorithm translation and no tools were developed to solve existing problems. The existing business rules did not involve all the gateway notations available in BPMN.

E. Mickeviciute, R. Butleris, S. Gudas, and E. Karciauskas [20] compared four solutions to transform BPMN into SBVR using 11 criteria. The purpose of this study was to determine business vocabulary and business rules in the form of SBVR obtained from a process model used for validation purposes by domain experts. The contributions of this research were: (1) automatic transformation of BPMN 2.0 into formally defined SBVR. The rules that were set included all elements in BPMN 2.0; (2) determining the requirements for transforming a BPMN 2.0 business process model into a consistent and complete SBVR. This research was conducted using simple cases. Five complex cases were not discussed in detail, namely: (1) successive decision points when there are more than two points at the BPMN gateway; (2) sub-processes in the same diagram (the sub-processes must be represented in separate activity diagrams); (3) the process specified for the BPMN 2.0 Call Activity element; (4) the name of the element that must be declared. There is no check that all elements of the rule are present and complete; and (5) the transformation does not include all parameters in BPMN 2.0 except the element name and the time duration parameter.

The result of the study of S. Arshad, I. S. Bajwa, and R. Kazmi [6] was the SBVR2XML tool. The trial conducted in the case study was in the form of SBVR from a cafeteria ordering system. In the end, an evaluation of this tool resulted in an average recall of 92.50% and an average precision of 94.87%. The case study did not give a detailed explanation of the stages of mapping SBVR to XSD to the XML generated results. No explanation was given for the

Table 2. Comparison of the previous studies on business rules with branching in SBVR

| Research  | SBVR Rule | Rules |   |   |   | SBVR Editor  | Research  | SBVR Rule | Rules |   |   |   | SBVR Editor |
|---|-----------|-------|---|---|---|--------------|---|-----------|-------|---|---|---|-------------|
|   |           | a     | b | c | d |              |   |           | a     | b | c | d |             |
| B. Akhtar, A. Mehmood, A. Mehmood, and W. Noor, 2019 [7]            | +         | -     | - | - | - | SBeVeaR      | Kluza & Honkisz, 2016 [13]  | +         | +     | - | + | - | unknown     |
| Skersys, Tutkute & Butleris, 2012 [1]                               | -         | -     | - | - | - | Vetis        | E. Mickeviciute, R. Butleris, S. Gudas, E. Karciauskas, 2017 [21] | +         | +     | - | + | - | QVT         |
| T. Skersys, K. Kapocius, R. Butleris, and T. Danikauskas, 2014 [20] | -         | +     | - | + | - | VeTIS        | Kluza & Nalepa, 2019 [22]   | +         | +     | - | + | - | unknown     |
| Tantan & Akoka, 2014 [10]   | +         | -     | - | - | - | unknown      | Arshad, Bajwa & Kazmi, 2019 [6]                                   | +         | -     | - | - | - | SBVR2 XML   |
| G. Aiello, R. Di Bernardo, M. Maggio, D, 2014 [8]                   | +         | +     | - | + | - | JBoss Drools | K. Anand, S. Mitra, and P. K. Chittimalli, 2019 [28]              | +         | +     | - | + | - | Burrito     |

Note: Rule a for AND condition; rule b for OR condition; rule c for XOR condition; rule d for Complex condition

Table 1. Definition variable for BPMN and SBVR formal model

| Variables       | Description  | Variables       | Description                               |
|-----------------|--|-----------------|---|
| $\mathcal{P}$   | a process  | $\mathbb{G}_x$  | exclusive gateway notation                |
| $\mathcal{O}$   | the set of flow objects  | $\mathbb{G}_o$  | inclusive gateway notation                |
| $\mathcal{M}$   | the set of the model attributes  | $\mathbb{G}_*$  | complex gateway notation                  |
| $\mathcal{F}$   | the set of sequence flows  | $\mathcal{S}$   | a set of SBVR                             |
| $\mathcal{M}_F$ | a subset of attributes that used in sequence flow  | $\mathcal{C}$   | a concept type                            |
| $\mathcal{A}$   | a set of activities  | $\mathcal{C}_n$ | a noun concept                            |
| $\mathcal{T}$   | a set of tasks   | $\mathcal{C}_i$ | an individual concept                     |
| $\mathcal{S}$   | a set of sub-processes   | $\mathcal{C}_v$ | a verb concept                            |
| $\mathcal{E}$   | the set of events  | $\mathcal{F}$   | a fact type                               |
| $\mathbb{G}$    | the set of gateways  | $\mathcal{K}$   | a keyword for designations or expressions |
| $F_g^{in}$      | sequence flow sets represent input flow  | $\mathcal{B}$   | a business rule                           |
| $F_g^{out}$     | sequence flow sets represent output flow   | $\mathcal{K}_1$ | Keyword “it is obligatory that”           |
| $\mathcal{M}_g$ | set of gateway attributes  | $\mathcal{K}_2$ | Keyword “after”                           |
| $g$             | Type of gateway (parallel, inclusive, exclusive, complex, event-based, parallel event-based) | $\mathcal{K}_3$ | Keyword “if”                              |
| $\mathbb{G}_+$  | parallel gateway notation  | $\mathcal{K}_4$ | Keyword “then”                            |

calculation of the average recall and average precision evaluation.

K. Kluza and G. J. Nalepa [21] compared previous research on formalizing business processes. In general, the primary purpose of formalization is to enable soundness checking by defining semantic execution. This research presents a formal description of business process integration using

business rules with the semantic knowledge engineering (SKE) approach.

In Table shows comparison between these previous researches by comparing whether the research discusses SBVR, logical rules in the form of AND, OR, XOR and complex rules, and the tool used as SBVR editor. From the comparison, the authors found that several studies discussed the transformation from BPMN to SBVR. SBVR rules

and parallelism rules related to all conditions of the AND, OR, XOR, and complex gateways in BPMN were not the main topics of these previous studies, which were aimed at further development of prior research. The present study discusses parallelism rules to complement previous studies related to the transformation of SBVR into BPMN.

### 3. Material and method

The method will use variables as listed in Table 1. All variables equipped with description to define variables for BPMN and SBVR formal model.

#### 3.1 SBVR and BPMN process model

SBVR are formed from a business vocabulary model and a business rule model. The results of natural language processing (NLP) from informal documents in the previous stages are input from this stage. In other words, the results of NLP are already in the form of concepts and business rules, which will be extracted into SBVR and then become BPMN. In Table 4 shows the transformation from SBVR to BPMN. The input is in the form of fact types (t1, t2, and t3). Based on the three fact types, SBVR 1 and SBVR2 are determined. SBVR1 is a combination of fact type t1 and fact type t2, and SBVR2 is a combination of fact type t2 and fact type t3. Both fact types are combined using the keywords ‘after’ or ‘if’ to indicate the order of the sentences related to the order of the task.

#### 3.2 Formal description of process model and SBVR

Based on process model in, the authors define a BPMN process model that describes the artifacts in BPMN notation. In general, the BPMN process

model consists of object flow and sequence flow. Flow objects are activities, events, and gateways. K. Kluza and G. J. Nalepa [21] show a formal model of the process model that is integrated with business rules; the process model used BPMN 2.0. Based on this research, the authors formulated a formal model of the process model with the XTT2 method approach.

#### 3.2.1 BPMN formal odel

Let  $O$  denote the set of flow object in process model,  $O = \{o_1, o_2, o_3, \dots, o_n\}$  where  $o_i \in O, i = 1..n$ . The set of model attributes denoted with  $M = (m_1, m, m_3, \dots, m_n)$  where  $m_j \in M, j = 1..n$ . The set of sequence flow form in Eq. (1). An  $F$  is proper subject of flow object  $O \times O \times 2^{M_F}$ :

$$F \subset O \times O \times 2^{M_F}, \tag{1}$$

where,  $M_F \subset M$  is a subset of attributes that used in sequence flow.

Furthermore, the set of flow objects  $O$  is denoted with  $A, E, \mathbb{G}$ . The set of activity  $A$  consists set of tasks  $T$  and a set of sub-processes  $S$  show in Eq. (2):

$$A = T \cup S, T \cap S = \emptyset, \tag{2}$$

where, the set of elements  $A$  which belong to  $T$  and  $S$  and to both  $T$  or  $S$ , the set is not empty set. The set of task  $T$  consists of  $\{t_1, t_2, t_3, \dots, t_n\}$  where  $t_k \in T, k = 1..n$ . A sub-process in BPMN is a specific process which contains a series of small parts of activity. The set of sub-process  $S = \{s_1, s_2, s_3, \dots, s_n\}$  where  $s_l \in S, l = 1..n$ . The set of event  $E$  consists of  $\{e_1, e_2, e_3, \dots, e_n\}$ ,  $e_i \in E, i = 1..n$ . Another part in activity is gateway. A set of gateways denoted with  $\mathbb{G}$ , where  $\mathbb{G} = \{g_1, g_2, g_3, \dots, g_n\}$ ,  $g_i \in \mathbb{G}, i = 1..n$  and

Table 4. Example of SBVR transformation to simple model process.

|               |  |
|---------------|--|
| Task          | t1 : determine subject of library material<br>t2 : determine classification number<br>t3 : check calling number on each copy   |
| Fact type     | f <sub>1</sub> : librarian determine subject of library material<br>f <sub>2</sub> : librarian determine classification number<br>f <sub>2</sub> : librarian check calling number on each copy   |
| SBVR          | SBVR1: <b>It is obligatory that</b> librarian <i>determine</i> classification number <b>after</b> librarian <i>determines</i> subject of library material<br>SBVR2: <b>It is obligatory that</b> librarian <i>check</i> calling number on each copy <b>if</b> librarian <i>determine</i> classification number |
| Process model | <pre> graph LR     Start((start)) --&gt; T1[determine subject of library material]     T1 --&gt; T2[determine classification number]     T2 --&gt; T3[check calling number on each copy]     T3 --&gt; End((end))             </pre>   |

complex, event-based, parallel event-based. The set of gateway attributes  $M_g$  is a proper subset of  $M$  ( $M_g \subset M$ ), where  $M_g$  consists of id, name, documentation, and gateway direction. The gateway direction consists of 2 direction as forking named diverging  $g_d$  and converging  $g_c$ . Gateway notation element  $\mathbb{G}_+$  as parallel gateway indicated in Eq. (3):

$$\mathbb{G}_+ = \{g \in \mathbb{G}: type(g) = Parallel\}; \quad (3)$$

gateway notation element for exclusive gateway  $\mathbb{G}_\times$  indicated in Eq. (4),

$$\mathbb{G}_\times = \{g \in \mathbb{G}: type(g) = Exclusive\}; \quad (4)$$

gateway notation for inclusive gateway  $\mathbb{G}_\circ$  indicate in Eq. (5),

$$\mathbb{G}_\circ = \{g \in \mathbb{G}: type(g) = Inclusive\}; \quad (5)$$

gateway notation for complex gateway  $\mathbb{G}_*$  indicate in Eq. (6),

$$\mathbb{G}_* = \{g \in \mathbb{G}: type(g) = Complex\}. \quad (6)$$

All gateways notation in Eqs. (3-6) will be use as scenarios in result and discussion part. Therefore,  $O = A \cup E \cup \mathbb{G}$  and  $A \cap E = A \cap \mathbb{G} = E \cap \mathbb{G} = \emptyset$ . The set of all possible BPMN 2.0 process model denoted with  $\mathcal{P}$  where  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n \in \mathcal{P}$ .

### 3.2.2 SBVR formal model

In research by Nalepa and Kaczor, Kluza and Nelapa, and Nalepa [25-27]; introduced knowledge representation for rule-based systems called XTT2 (eXtended Tabular Trees). The formal description of the SBVR follows a formal description of the process model. Formally XTT2 will be serialized to the XML-based XMI format [5]. After the formal process model has been identified, the authors continue with the formal SBVR model.

An SBVR of a process model is a tuple of  $\mathbb{S} = (\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_n)$  where  $\mathcal{B}_i \in \mathcal{B}, i = 1..n$ . An SBVR formed by business rules. Business rule ( $\mathcal{B}$ ) is a combination between two or more  $\mathcal{F}$  using the keywords ( $\mathcal{K}$ ). Business rules specified in operational, structural, and derivation rules [24]. Fact type structured with combination of concept types. Concept type consists of noun concept ( $\mathcal{C}_n$ ), individual concept ( $\mathcal{C}_i$ ) and verb concept ( $\mathcal{C}_v$ ). Combination of concept type to arrange fact type formulated in Eq. (7):

$$\mathcal{F} = \mathcal{C}_n \cup \mathcal{C}_v = \mathcal{C}_n \cup \mathcal{C}_v \cup \mathcal{C}_i, \quad (7)$$

where  $\mathcal{F} = T$ , e.g.  $(f_1, f_2, \dots, f_n) = (t_1, t_2, \dots, t_n) =$  where  $f_i \in \mathcal{F}, i = 1..n$  and  $t_i \in T, i = 1..n$ . Concept type  $\mathcal{C} \ni (\mathcal{C}_n, \mathcal{C}_v, \mathcal{C}_i)$

Business rules in BPMN categorize as operative rules, so the keyword sentence use “**it is obligatory that**”.  $\mathcal{K}$  is consists of “**it is obligatory that**” as  $k_1$ , “**after**” as  $k_2$ , “**if**” as  $k_3$ , “**then**” as  $k_4$ ; where  $k_1, k_2, k_3, k_4 \in \mathcal{K}$ . The formal model of SBVR shows in Eq. (8):

$$\mathcal{B}_{k1,2} = (k_1, f_{i+1}, k_2, f_{j-1}), \quad (3)$$

where  $f_{i+1} \in \mathcal{F}, i = 1..n$  and  $f_{j-1} \in \mathcal{F}, j = 2..n$  and keyword use “**it is obligatory that**” ( $k_1$ ) and “**after**” ( $k_2$ ). Another formal model that use keyword  $k_1$  and  $k_3$  shows in Eq. (9):

$$\mathcal{B}_{k1,3} = (k_1, f_{j-1}, k_3, f_{i+1}), \quad (4)$$

where  $f_{i+1} \in \mathcal{F}, i = 1..n$  and  $f_{j-1} \in \mathcal{F}, j = 2..n$ . There are some cases consists of forking which is indicated by gateway ( $\mathbb{G}$ ) the rule will be:

Parallel gateway  $\mathbb{G}_+$  formal model shows in Eq. (10):

$$\mathcal{B}_+ = (k_1, f_{i+1} \wedge f_{i+2} \wedge \dots \wedge f_{i+n}, k_2, f_{j-1}), \quad (5)$$

where, all of  $f_{i+1}, f_{i+2}, \dots, f_{i+n}$  must be done,  $f_{i+1}, f_{i+2}, \dots, f_{i+n} \in \mathcal{F}, i = 1..n$ , and  $f_{j-1} \in \mathcal{F}, j = 2..n$ .

Exclusive gateway  $\mathbb{G}_\times$  formal model shows in Eq. (11):

$$\mathcal{B}_\times = (k_1, f_{i+1} \times f_{i+2} \times \dots \times f_{i+n}, k_2, f_{j-1}), \quad (6)$$

where, all of  $f_{i+1}, f_{i+2}, \dots, f_{i+n}$  must choose exactly one of fact types,  $f_{i+1}, f_{i+2}, \dots, f_{i+n} \in \mathcal{F}, i = 1..n$ , and  $f_{j-1} \in \mathcal{F}, j = 2..n$ .

Inclusive gateway  $\mathbb{G}_\circ$  formal model shows in Eq. (12):

$$\mathcal{B}_\circ = (k_1, f_{i+1} \vee f_{i+2} \vee \dots \vee f_{i+n}, k_2, f_{j-1}), \quad (7)$$

where, all of  $f_{i+1}, f_{i+2}, \dots, f_{i+n}$  can choose at least one of all fact types,  $f_{i+1}, f_{i+2}, \dots, f_{i+n} \in \mathcal{F}, i = 1..n$ , and  $f_{j-1} \in \mathcal{F}, j = 2..n$ .

Complex gateway  $\mathbb{G}_*$  formal model shows in Eq. (13):

$$\mathcal{B}_* = (k_1, f_{i+1} \vee f_{i+2} \vee \dots \vee f_{i+n}, k_2, f_{j-1}), \quad (8)$$

where, all of  $f_{i+1}, f_{i+2}, \dots, f_{i+n}$  can choose at least one of all fact types with extra rules or condition,  $f_{i+1}, f_{i+2}, \dots, f_{i+n} \in \mathcal{F}, i = 1..n$  and  $f_{j-1} \in \mathcal{F}, j = 2..n$ .

$\mathbb{G}_{\otimes}$  as event-based gateway and  $\mathbb{G}_{\oplus}$  as parallel event-based gateway did not discuss, because it involved an alternative path that follow the gateway based on events. Those situation in sequence flow is out of the subject in this research.

#### 4. Result and disscusion

Four scenarios were used to carry out the evaluation. The authors used the formal process model above to represent the business process of four different cases in a university library. Forking situations are represented by using the exclusive gateway, the parallel gateway, the inclusive gateway, and the complex gateway.

##### 4.1 Scenario 1: Library material classification process

In this scenario, the authors consider the BPMN of a university library for library material classification, as shown in Fig. 2. Some conditions require branching, for example when it has to be determined whether the library material has been classified or not. If it has been classified, then the library material will be sent directly to the inventory

subsection. If the library material has not been classified yet, then it is necessary to classify it and then send it to the inventory subsection. In this scenario, the exclusive gateway is appropriate to use. The exclusive gateway is a gateway where only one of the task paths can be taken. Fig. 2 shows an exemplary BPMN model consisting of two events, two gateways, and seven tasks, model of process library material classification as the pool ( $\mathcal{P}_1$ ) with lane ( $O_1$ ) as the actor called 'librarian'. An explanation of each task, event, and gateway is shown Table .

Using the formulation presented in section 3.2.1, the formal BPMN model of library material classification process is presented:

- $O_1 = A_1 \cup E_1 \cup \mathbb{G}_1$  , where  $A_1 = T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$  ,  $E_1 = \{e_1, e_2\}$  ,  $\mathbb{G}_1 = \{g_1, g_2\}$  where
  - $type(g_1) = type(g_2) = Exclusive$ , accordingly,  $\mathbb{G}_x = \{g_1, g_2\}$ ;
  - $GatewayDirection_{g_1} = g_d$ ;
  - $GatewayDirection_{g_2} = g_c$ ;
- $F_{g_1}^{in} = \{(t_2, g_1, M_{t_2, g_1})\}$  and  $F_{g_1}^{out} = \{(g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4})\}$ ;
- $F_{g_2}^{in} = \{(t_3, g_2, M_{t_3, g_2}), (t_6, g_2, M_{t_6, g_2})\}$  and  $F_{g_2}^{out} = \{(g_2, t_7, M_{g_2, t_7})\}$ ;
- $F_1 = \{(e_1, t_1, M_{e_1, t_1}), (t_1, t_2, M_{t_1, t_2}), (t_2, g_1, M_{t_2, g_1}), \{(g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4})\}, (t_4, t_5, M_{t_4, t_5}), (t_5, t_6, M_{t_5, t_6}), \{(t_3, g_2, M_{t_3, g_2}), (t_6, g_2, M_{t_6, g_2})\}, (g_2, t_7, M_{g_2, t_7}), (t_7, e_2, M_{t_7, e_2})\}$ ;

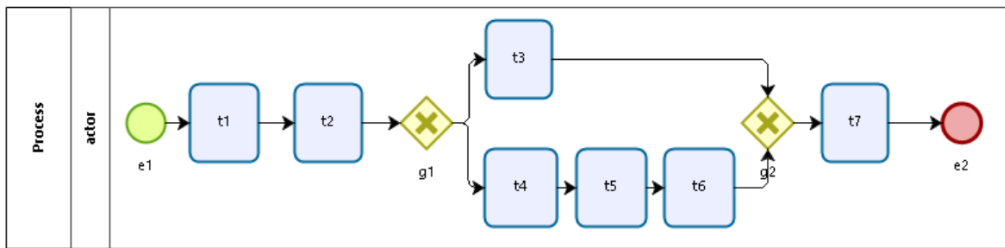


Figure. 2 BPMN of library material classification process

Table 5. Process model explanation of library material classification

| Initial | Flow object                | Label                                | Initial | Flow object                | Label                               |
|---------|----------------------------|--------------------------------------|---------|----------------------------|-------------------------------------|
| $e_1$   | event1                     | Start                                | $t_5$   | task5                      | Determine the classification number |
| $t_1$   | task1                      | Receive library material             | $t_6$   | task6                      | Check call number on each copy      |
| $t_2$   | task2                      | Check existence of library materials | $g_2$   | gateway2 (converging/join) |                                     |
| $g_1$   | gateway1 (diverging/split) | Exist?                               | $t_7$   | task7                      | Send to the inventory subsection    |
| $t_3$   | task3                      | Match subject and call number        | $e_2$   | event2                     | End                                 |
| $t_4$   | task4                      | Determine subject                    |         |                            |                                     |



- $M_1 \supset \{M_{e_1,t_1}, M_{t_1,t_2}, M_{t_2,g_1}, M_{g_1,t_3}, M_{g_1,t_4}, M_{t_4,t_5}, M_{t_5,t_6}, M_{t_3,g_2}, M_{t_6,g_2}, M_{g_2,t_7}, M_{t_7,e_2}\}$ , where  $M_1$  are sets of attribute sequence flow.

The XOR condition in Fig.2 is a diverging or split XOR condition, expressed as  $(F_{g_1}^{in} = \{(t_2, g_1, M_{t_2,g_1})\})$ ,  $(F_{g_1}^{out} = \{(g_1, t_3, M_{g_1,t_3}), (g_1, t_4, M_{g_1,t_4})\})$ .

A converging or join XOR condition is expressed as  $(F_{g_2}^{in} = \{(t_3, g_2, M_{t_3,g_2}), (t_6, g_2, M_{t_6,g_2})\})$ ,  $(F_{g_2}^{out} = \{g_2, t_7, M_{g_2,t_7}\})$ .

Furthermore, using the formulation presented in section 3.2.2 Eqs. (7)- (9), (11) the formal SBVR of the library material classification process presented:

$C_n = (\text{librarian, library material, subject and call number, subject, classification number, call number})$ ;

$C_v = (\text{receive, check existence, match, determine, check, send})$ ;

- $\mathcal{F} = C_n \cup C_v$ , where  $f_1, f_2, f_3, f_4, f_5, f_6, f_7 \in \mathcal{F}$ ,  
 $f_1$ : librarian receive library material;  
 $f_2$ : librarian check existence of library materials;  
 $f_3$ : librarian match subject and call number;  
 $f_4$ : librarian determine subject;  
 $f_5$ : librarian determine the classification number;  
 $f_6$ : librarian check call number on each copy;  
 $f_7$ : librarian send to the inventory subsection;
- $B_1 = (k_1, f_2, k_2, f_1)$ ;

- $B_{2(\times)} = (k_1, (f_3 \times f_4), k_2, f_2)$ , where  $f_3 = \text{true then } f_4 = \text{false} \vee f_4 = \text{true then } f_3 = \text{false}$ ;
- $B_3 = (k_1, f_5, k_2, f_4)$ , where  $f_4 = \text{true then } f_3 = \text{false}$ ;
- $B_4 = (k_1, f_6, k_2, f_5)$ , where  $f_4 = \text{true then } f_3 = \text{false}$ ;
- $B_{5(\times)} = \{(k_1, f_7, k_2, f_3), \text{ if } f_3 = \text{true then } f_4 = \text{false}\} \vee \{(k_1, f_5, k_2, f_4), \text{ if } f_4 = \text{true then } f_3 = \text{false}\}$ ;
- $B_1, B_{2(\times)}, B_3, B_4, B_{5(\times)} \in B$ .

### 4.2 Scenario 2: Library exhibition process

When the university library holds an exhibition, many things must be prepared by executing several processes simultaneously. Therefore, the tasks of selecting the exhibition material, determining the design, preparing support events, and preparing promotion concepts can be carried out simultaneously by the librarians. In this situation, the authors need a gateway to represent the situation that occurs in this process. A parallel gateway is a gateway that can run more than one path simultaneously. Fig. 3 shows that there are twelve tasks with two events, two parallel gateways, pool ( $\mathcal{P}_2$ ) is the process called Library exhibition, and lane ( $O_2$ ) is librarian as actor of the process. An

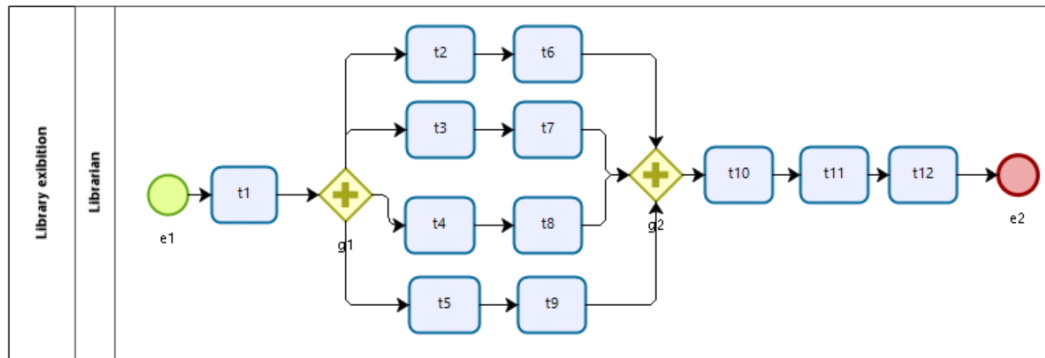


Figure. 3 BPMN of library exhibition process

Table 6. Process model explanation of library exhibition

| Initial | Flow object                | Label                                |
|---------|----------------------------|--------------------------------------|
| $e_1$   | event1                     | Start                                |
| $t_1$   | task1                      | Receive library material             |
| $t_2$   | task2                      | Check existence of library materials |
| $g_1$   | gateway1 (diverging/split) | Exist?                               |
| $t_3$   | task3                      | Match subject and call number        |
| $t_4$   | task4                      | Determine subject                    |

| Initial | Flow object                | Label                               |
|---------|----------------------------|-------------------------------------|
| $t_5$   | task5                      | Determine the classification number |
| $t_6$   | task6                      | Check call number on each copy      |
| $g_2$   | gateway2 (converging/join) |                                     |
| $t_7$   | task7                      | Send to the inventory subsection    |
| $e_2$   | event2                     | End                                 |

explanation of each task, event, and gateway is shown in Table. Using the presented formal process model of the library exhibition process ( $P_2$ ) presented:

$\mathcal{P}_2 = (O_2, F_2, M_2)$  where  $O_2 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, e_1, e_2, g_1, g_2\}$  and  $O_2$  can be divided into subsets below:  
 $O_2 = A_2 \cup E_2 \cup \mathbb{G}_2$ ,  $A_2 = T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}\}$ ,  $E_1 = \{e_1, e_2\}$ ,  $\mathbb{G}_1 = \{g_1, g_2\}$  where  $type(g_1) = type(g_2) = Parallel$ , accordingly  $\mathbb{G}_+ = \{g_1, g_2\}$ ;  
 GatewayDirection $_{g_1} = g_d$ ;  
 GatewayDirection $_{g_2} = g_c$ ;  
 $F_{g_1}^{in} = \{(t_1, g_1, M_{t_1, g_1})\}$  and  $F_{g_1}^{out} = \{(g_1, t_2, M_{g_1, t_2}), (g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4}), (g_1, t_5, M_{g_1, t_5})\}$ ;  
 $F_{g_2}^{in} = \{(t_6, g_2, M_{t_6, g_2}), (t_7, g_2, M_{t_7, g_2}), (t_8, g_2, M_{t_8, g_2}), (t_9, g_2, M_{t_9, g_2})\}$  and  $F_{g_2}^{out} = \{(g_2, t_{10}, M_{g_2, t_{10}})\}$ ;  
 $F_2 = (e_1, t_1, M_{e_1, t_1}), (t_1, g_1, M_{t_1, g_1}), \{(g_1, t_2, M_{g_1, t_2}), (g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4}), (g_1, t_5, M_{g_1, t_5})\}, (t_2, t_6, M_{t_2, t_6}), (t_3, t_7, M_{t_3, t_7}), (t_4, t_8, M_{t_4, t_8}), (t_5, t_9, M_{t_5, t_9}), \{(t_6, g_2, M_{t_6, g_2}), (t_7, g_2, M_{t_7, g_2}), (t_8, g_2, M_{t_8, g_2}), (t_9, g_2, M_{t_9, g_2})\}, (g_2, t_{10}, M_{g_2, t_{10}}), (t_{10}, t_{11}, M_{t_{10}, t_{11}}), (t_{11}, t_{12}, M_{t_{11}, t_{12}}), (t_{12}, e_2, M_{t_{12}, e_2})$ ;  
 $M_2 \supset \{M_{e_1, t_1}, M_{t_1, g_1}, M_{g_1, t_2}, M_{g_1, t_3}, M_{g_1, t_4}, M_{g_1, t_5}, M_{t_2, t_6}, M_{t_3, t_7}, M_{t_4, t_8}, M_{t_5, t_9}, M_{t_6, g_2}, M_{t_7, g_2}, M_{t_8, g_2}, M_{t_9, g_2}, M_{g_2, t_{10}}, M_{t_{10}, t_{11}}, M_{t_{11}, t_{12}}, M_{t_{12}, e_2}\}$  where  $M_2$  are sets of attribute sequence flow.

Using the formulation presented in section 3.2.2 Eqs. (7)- (10), the SBVR formulation of the library exhibition process presented:

- $C_n = (librarian, theme, material, design, support event, promotion concept, catalog, completion, HR, facilities, and infrastructure, promotion strategies)$ ;
- $C_v = (determine, select, prepare, compilation, determination, implementation, promotion, reporting)$ ;
- $\mathcal{F} = C_n \cup C_v$ , where  $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12} \in \mathcal{F}$ ,  $f_1$  : librarian determine theme;

- $f_2$  : librarian select material;
- $f_3$  : librarian determine design;
- $f_4$  : librarian prepare support event;
- $f_5$  : librarian prepare promotion concept;
- $f_6$  : librarian compile catalog;
- $f_7$  : librarian prepare for completion;
- $f_8$  : librarian prepare for HR, facilities, and infrastructure;
- $f_9$  : librarian determine promotion strategies;
- $f_{10}$  : librarian implement event;
- $f_{11}$  : librarian promote event;
- $f_{12}$  : librarian report event;
- $\mathcal{B}_{1(+)} = (k_1, (f_2 \wedge f_3 \wedge f_4 \wedge f_5), k_2, f_1)$ , where  $\mathcal{B}_{1(+)}$  is  $\mathcal{B}_1$  with parallel gateway;
- $\mathcal{B}_2 = (k_1, f_6, k_2, f_2)$ ;
- $\mathcal{B}_3 = (k_1, f_7, k_2, f_3)$ ;
- $\mathcal{B}_4 = (k_1, f_8, k_2, f_4)$ ;
- $\mathcal{B}_5 = (k_1, f_9, k_2, f_5)$ ;
- $\mathcal{B}_{6(+)} = (k_1, f_{10}, k_2, (f_6 \wedge f_7 \wedge f_8 \wedge f_9))$ ;
- $\mathcal{B}_7 = (k_1, f_{10}, k_2, f_9)$ ;
- $\mathcal{B}_8 = (k_1, f_{11}, k_2, f_{10})$ ;
- $\mathcal{B}_9 = (k_1, f_{12}, k_2, f_{11})$ ;
- $\mathcal{B}_{1(+)}, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_{6(+)}, \mathcal{B}_7, \mathcal{B}_8, \mathcal{B}_9 \in \mathcal{B}$

### 4.3 Scenario 3: Procurement process

The inclusive gateway or OR condition is a condition where there are branches in the form of one or more tasks that can be chosen to run. The procurement process at the library is the process of procuring stationery and inventory items. The procurement process is carried out for stationary purchases, procurement of inventory items, or both. This condition causes the procurement business process modeling to use an inclusive gateway (OR condition).

In another particular case the OR condition finish task 7 ( $t_7$ ) has an XOR condition that will select one of the expanded subprocesses. However, when purchasing stationery ( $t_5$  and  $t_6$ ) is done, this condition requires the expanded subprocesses distribute stationery ( $s_2$ ), whereas if the previous process is the expanded subprocesses tender ( $s_1$ ), the expanded subprocesses inventory goods distribution ( $s_3$ ) should be run. The research by S. Huda, R. Sarno,

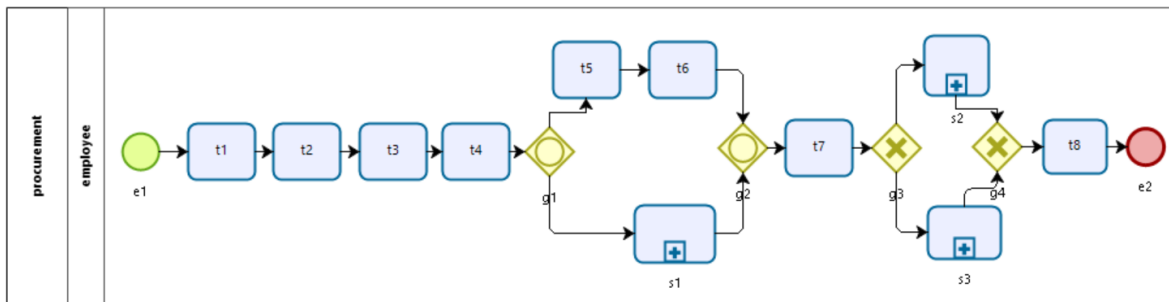


Figure. 4 BPMN of the procurement process.

Table 7. Explanation of the procurement process

| Initial | Flow object                             | Label                     | Initial | Flow object                                | Label                     |
|---------|---|---------------------------|---------|--|---------------------------|
| $e_1$   | event1                                  | Start                     | $g_2$   | gateway2<br>(Inclusive OR converge/join)   |                           |
| $t_1$   | task1                                   | identify                  | $t_7$   | task7                                      | distribute goods          |
| $t_2$   | task2                                   | classify                  | $t_8$   | task8                                      | archive                   |
| $t_3$   | task3                                   | prioritize                | $g_3$   | Gateway3<br>(XOR Exclusive diverge/split)  |                           |
| $t_4$   | task4                                   | verify specification      | $s_2$   | Expanded Subprocess stationery             | distribute stationery     |
| $g_1$   | gateway1<br>(InclusiveOR diverge/split) |                           | $s_3$   | Expanded Subprocess inventory              | distribute inventory good |
| $t_5$   | task5                                   | purchase directly         | $g_4$   | Gateway4<br>(XOR Exclusive converge /join) |                           |
| $t_6$   | task6                                   | buy goods from the seller | $e_2$   | event2                                     | End                       |
| $s_1$   | Expanded Subprocess tender              | Tender                    |         |  |                           |

T. Ahmad, and H. A. Santoso [25] asserted that such conditions are non-free choice constructs. A non-free choice condition arises when a task is executed after a specific job, which gets a parallel control-flow pattern [26]. In Fig. 4 shows an exemplary BPMN model consisting of procurement process as pool ( $\mathcal{P}_3$ ), lane ( $O_3$ ) employee as, two events ( $e_1$  and  $e_2$ ), four gateways ( $g_1, g_2, g_3, g_4$ ), eight tasks ( $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$ ) and three expanded subprocesses ( $s_1, s_2, s_3$ ). In Table shows an explanation of Fig. 4. The OR condition is a task branching condition where the authors can choose one or more tasks, where the default choice is true, and the other task is false.

Based on Eqs. (1)- (2) and (4)- (5) the BPMN of the procurement process can be described using a formal process:

$\mathcal{P}_3 = (O_3, F_3, M_3)$  where

$O_3 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, e_1, e_2, g_1, g_2, g_3, g_4\}$  and

$O_3 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, e_1, e_2, g_1, g_2, g_3, g_4\}$  and

$O_3$  can be divided into subsets:

$O_3 = A_3 \cup E_3 \cup \mathbb{G}_3$ ,  $A_3 = T \cup S$ ,  $T =$

$\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$ ,  $S = \{s_1, s_2, s_3\}$ ,  $E_1 = \{e_1, e_2\}$ ,

$\mathbb{G}_1 = \{g_1, g_2, g_3, g_4\}$  where

-  $type(g_1) = type(g_2) = Inclusive$ , accordingly

$\mathbb{G}_0 = \{g_1, g_2\}$ ;

-  $GatewayDirection_{g_1} = g_d$ ;

-  $GatewayDirection_{g_2} = g_c$ ;

-  $type(g_3) = type(g_4) = Exclusive$ , accordingly

$\mathbb{G}_x = \{g_3, g_4\}$ ;

-  $GatewayDirection_{g_3} = g_d$ ;

-  $GatewayDirection_{g_4} = g_c$ ;

-  $F_{g_1}^{in} = \{(t_4, g_1, M_{t_4, g_1})\}$  and  $F_{g_1}^{out} = \{(g_1, t_5, M_{g_1, t_5}), (g_1, s_t, M_{g_1, s_t})\}$ ;

-  $F_{g_2}^{in} = \{(t_6, g_2, M_{t_6, g_2}), (s_t, g_2, M_{s_t, g_2})\}$  and  $F_{g_2}^{out} = \{(g_2, t_7, M_{g_2, t_7})\}$ ;

-  $F_{g_3}^{in} = \{(t_7, g_3, M_{t_7, g_3})\}$  and  $F_{g_3}^{out} = \{(g_3, s_s, M_{g_3, s_s}), (g_3, s_i, M_{g_3, s_i})\}$

-  $F_{g_4}^{in} = \{(s_s, g_4, M_{s_s, g_4}), (s_i, g_4, M_{s_i, g_4})\}$  and  $F_{g_4}^{out} = \{(g_4, e_2, M_{g_4, e_2})\}$ ;

-  $F_3 = \{(e_1, t_1, M_{e_1, t_1}), (t_1, t_2, M_{t_1, t_2}), (t_2, t_3, M_{t_2, t_3}), (t_3, t_4, M_{t_3, t_4}), (t_4, g_1, M_{t_4, g_1}), \{(g_1, t_5, M_{g_1, t_5}), (g_1, s_t, M_{g_1, s_t})\}, (t_5, t_6, M_{t_5, t_6}), \{(t_6, g_2, M_{t_6, g_2}), (s_t, g_2, M_{s_t, g_2})\}, \{(t_7, g_3, M_{t_7, g_3})\}, \{(g_3, s_s, M_{g_3, s_s}), (g_3, s_i, M_{g_3, s_i})\}, \{(s_s, g_4, M_{s_s, g_4}), (s_i, g_4, M_{s_i, g_4})\}, \{(g_4, e_2, M_{g_4, e_2})\}\}$

-  $M_3 \supset \{M_{e_1, t_1}, M_{t_1, t_2}, M_{t_2, t_3}, M_{t_3, t_4}, M_{g_1, t_4}, M_{t_4, g_1}, M_{g_1, t_5}, M_{g_1, s_t}, M_{t_5, t_6}, M_{t_6, g_2}, M_{s_t, g_2}, M_{g_2, t_7}, M_{t_7, g_3}, M_{g_3, s_s}, M_{g_3, s_i}, M_{s_s, g_4}, M_{s_i, g_4}, M_{g_4, e_2}\}$  where  $M_3$  are sets of attributes sequence flow.

SBVR formulation of procurement process based on section 3.2.2 in Eqs. (7)- (10) and (11), presented:

-  $C_n = (employee, seller, goods, stationary, specification, inventory goods)$ ;

$C_v = (identify, classify, prioritize, verify, purchase, buy, tender, distribute, archive)$ ;

-  $\{s_1, s_2, s_3\} \in S$

$F = C_n \cup C_v$ , where  $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8 \in F$ ,

$f_1$ : employee identify;

$f_2$ : employee classify;

$f_3$ : employee prioritize;

$f_4$ : employee verify specification ;

$f_5$ : employee purchase directly ;

$f_6$ : employee buy goods from the seller;

$f_7$ : employee distribute goods;

$f_8$ : employee archive;

- $\mathcal{B}_1 = (k_1, f_2, k_2, f_1)$ ;
- $\mathcal{B}_2 = (k_1, f_3, k_2, f_2)$ ;
- $\mathcal{B}_3 = (k_1, f_4, k_2, f_3)$ ;
- $\mathcal{B}_{4(\circ)} = (k_1, (f_8 \vee s_1), k_2, f_4)$ ;
- $\mathcal{B}_5 = (k_1, f_6, k_2, f_5)$ ;
- $\mathcal{B}_6 = (k_1, f_7, k_2, (f_5 \vee s_2))$ ;
- $\mathcal{B}_{7(\times)} = (k_1, (s_2 \times s_3), k_2, f_7)$ ;
- $\mathcal{B}_8 = (k_1, f_8, k_2, s_2)$ ,  $s_2 = true$ ,  $s_3 = false$ ;
- $\mathcal{B}_8 = (k_1, f_8, k_2, s_3)$ ,  $s_2 = false$ ,  $s_3 = true$ ;
- $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_{4(\circ)}, \mathcal{B}_5, \mathcal{B}_6, \mathcal{B}_{7(\times)}, \mathcal{B}_8, \in \mathcal{B}$

#### 4.4 Scenario 4: Complex gateway

The complex gateway can be used to model complex synchronization behavior. An expression is used to describe the correct behavior. The outgoing paths taken by the gateway are determined by the conditions of the outgoing sequence flows as in the split behavior of the inclusive gateway. The process model below shows the book conservation process. Nested conditions are gateway conditions that indicate a gateway within a gateway. It shown in Fig.5 where gateways  $g_2, g_3$  inside gateways  $g_1, g_4$ . Complex gateways need synchronization in multiple conditions, i.e. accept book with minor damage ( $t_3$ ), accept book with major damage (irreparable) ( $t_4$ ), and accept book with major damage (reparable) ( $t_5$ ), which is a complex condition. The expressions that apply to this condition are  $t_3$  and  $t_5$  and can be run with the OR condition, but  $t_4$  cannot run simultaneously with  $t_4$  and  $t_5$ , because  $t_4$  concerns an irreparable book, so it must store in a warehouse. Fig. 5 shows an exemplary BPMN model consisting of Book reparable conservation process as pool ( $\mathcal{P}_4$ ), actor ‘librarian’ as lane ( $O_4$ ) two events ( $e_1$  and  $e_2$ ), four gateways ( $g_1, g_2, g_3, g_4$ ), thirteen tasks ( $t_1, t_2, \dots, t_{13}$ ) and one expanded subprocess ( $s_1$ ). Table shows an explanation of Fig. 5. In the complex gateway, the combination of OR and XOR conditions exist. We use Eqs. (1)- (2), (4), and (6), the BPMN formal method of the book conservation process presented:

$\mathcal{P}_4 = (O_4, F_4, M_4)$  where

$O_4 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, e_1, e_2, g_1, g_2, g_3, g_4\}$  and  $O_4$  can be divided into subsets:

- $O_4 = A_3 \cup E_3 \cup \mathbb{G}_3$ ,  $A_3 = T \cup S$ ,
- $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}\}$ ,
- $S = \{s_1\}$ ,  $E_1 = \{e_1, e_2\}$ ,  $\mathbb{G}_1 = \{g_1, g_2, g_3, g_4\}$  where
- $type(g_1) = type(g_2) = Complex$ , accordingly  $\mathbb{G}_* = \{g_1, g_2\}$ ;
- $GatewayDirection_{g_1} = g_d$ ;
- $GatewayDirection_{g_4} = g_c$ ;
- $type(g_2) = type(g_3) = Exclusive$ , accordingly  $\mathbb{G}_\times = \{g_2, g_3\}$ ;
- $GatewayDirection_{g_2} = g_d$ ;
- $GatewayDirection_{g_3} = g_c$ ;

- $F_{g_1}^{in} = \{(t_2, g_1, M_{t_2, g_1})\}$  and  $F_{g_1}^{out} = \{(g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4}), (g_1, t_5, M_{g_1, t_5})\}$ ;
- $F_{g_2}^{in} = \{(t_6, g_2, M_{t_6, g_2})\}$  and  $F_{g_2}^{out} = \{(g_2, t_{10}, M_{g_2, t_{10}}), (g_2, t_{11}, M_{g_2, t_{11}})\}$ ;
- $F_{g_3}^{in} = \{(t_{10}, g_3, M_{t_{10}, g_3}), (t_{11}, g_3, M_{t_{11}, g_3})\}$  and  $F_{g_3}^{out} = \{(g_3, t_{12}, M_{g_3, t_{12}})\}$ ;
- $F_{g_4}^{in} = \left\{ (t_{12}, g_4, M_{t_{12}, g_4}), (t_9, g_4, M_{t_9, g_4}), (s_1, g_4, M_{s_1, g_4}) \right\}$  and  $F_{g_4}^{out} = \{(g_4, t_{13}, M_{g_4, t_{13}})\}$ ;
- $F_4 = (e_1, t_1, M_{e_1, t_1}), (t_1, t_2, M_{t_1, t_2}), \{(t_2, g_1, M_{t_2, g_1}), (g_1, t_3, M_{g_1, t_3}), (g_1, t_4, M_{g_1, t_4}), (g_1, t_5, M_{g_1, t_5})\}, \{(t_6, g_2, M_{t_6, g_2})\}, \{(g_2, t_{10}, M_{g_2, t_{10}}), (g_2, t_{11}, M_{g_2, t_{11}})\}, \{(t_{10}, g_3, M_{t_{10}, g_3}), (t_{11}, g_3, M_{t_{11}, g_3})\}, \{(g_3, t_{12}, M_{g_3, t_{12}})\}, (t_4, t_7, M_{t_4, t_7}), (t_7, t_8, M_{t_7, t_8}), (t_8, t_9, M_{t_8, t_9}), (t_5, s_1, M_{t_5, s_1}), \{(t_{12}, g_4, M_{t_{12}, g_4}), (t_9, g_4, M_{t_9, g_4}), (s_1, g_4, M_{s_1, g_4})\}, \{(g_4, t_{13}, M_{g_4, t_{13}})\}, (t_{13}, e_2, M_{t_{13}, e_2})$
- $M_4 \supset \{M_{e_1, t_1}, M_{t_1, t_2}, M_{t_2, g_1}, M_{g_1, t_3}, M_{g_1, t_4}, M_{g_1, t_5}, M_{t_6, g_2}, M_{g_2, t_{10}}, M_{g_2, t_{11}}, M_{t_{10}, g_3}, M_{t_{11}, g_3}, M_{g_3, t_{12}}, M_{t_4, t_7}, M_{t_7, t_8}, M_{t_8, t_9}, M_{t_5, s_1}, M_{t_{12}, g_4}, M_{t_9, g_4}, M_{s_1, g_4}, M_{g_4, t_{13}}, M_{t_{13}, e_2}\}$  where  $M_4$  are sets of attributes sequence flow.

SBVR formulation of book conservation process based on section 3.2.2 in Eqs. (7)- (9) and (13) shown:

- $\mathcal{C}_n = (librarian, book, missing\ pages, irreparable\ book, official\ report, replacement\ pages)$ ;
- $\mathcal{C}_v = (Receive, Sort, Accept, Check, count, repair, create, put, look\ for, invisible\ task, fix)$ ;
- $s_1 \in S$  as a sub process
- $\mathcal{F} = \mathcal{C}_n \cup \mathcal{C}_v$ , where
- $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13} \in \mathcal{F}$ ,
- $f_1$  : librarian receive book;
- $f_2$  : librarian sort book;
- $f_3$  : librarian accept book with minor damage
- $f_4$  : librarian accept book with major damage (irreparable);
- $f_5$  : librarian accept book with major damage (reparable);
- $f_6$  : librarian check for missing pages;
- $f_7$  : librarian count the irreparable book ;
- $f_8$  : librarian create an official report of the book damage;
- $f_9$  : librarian put the book into a warehouse;
- $f_{10}$  : librarian look for replacement pages;
- $f_{11}$  : invisible task;
- $f_{12}$  : librarian fix the book;
- $f_{13}$  : invisible task;
- $\mathcal{B}_1 = (k_1, f_2, k_2, f_1)$ ;
- $\mathcal{B}_{2(*)} = (k_1, (f_3 \vee f_5 \vee (f_3 \wedge f_5) \vee f_4), k_2, f_2)$ ;
- $\mathcal{B}_3 = \{(k_1, f_6, k_2, f_3), f_3 \vee (f_3 \wedge f_5) = true, f_4 = false\}$ ;
- $\mathcal{B}_4 = \{(k_1, s_1, k_2, f_5), f_5 \vee (f_3 \wedge f_5) = true, f_4 = false\}$ ;

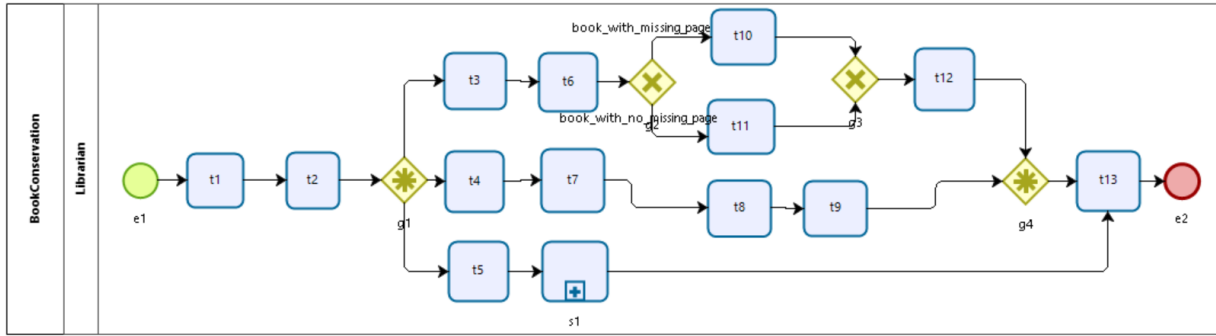


Figure. 5 BPMN of the book conservation process model

Table 8. Explanation of book conservation process model

| Initial | Flow object          | Label                                       | Initial  | Flow object                      | Label  |
|---------|----------------------|---|----------|----------------------------------|--|
| $e_1$   | event1               | Start                                       | $t_8$    | task8                            | Create an official report of the book damage |
| $t_1$   | task1                | Receive book                                | $t_9$    | task9                            | Put the book into a warehouse                |
| $t_2$   | task2                | Sort book                                   | $g_2$    | gateway2                         | (xor diverge /split)                         |
| $g_1$   | gateway1             | (complex diverge/split)                     | $t_{10}$ | task10                           | Look for replacement pages                   |
| $t_3$   | task3                | Accept book with minor damage               | $t_{11}$ | task11                           | Invisible task                               |
| $t_4$   | task4                | Accept book with major damage (irreparable) | $g_3$    | gateway3 (xor converge /join)    |  |
| $t_5$   | task5                | Accept book with major damage (reparable)   | $t_{12}$ | task12                           | Fix the book                                 |
| $t_6$   | task6                | Check for missing pages                     | $g_4$    | gateway4 (complex converge/join) |  |
| $t_7$   | task7                | Count the irreparable book                  | $t_{13}$ | task13                           | Invisible task                               |
| $s_1$   | Expanded Subprocess1 | Repair the book                             | $e_2$    | event2                           | End  |

- $\mathcal{B}_5 = \{(k_1, f_7, k_2, f_4), f_4 = true, f_3 \vee f_5 \vee (f_3 \wedge f_5) = false\}$ ;
- $\mathcal{B}_{6(\times)} = \{(k_1, (f_{10} \times f_{11}), k_2, \vee s_2), f_3 \vee (f_3 \wedge f_5) = true, f_4 = false\}$ ;
- $\mathcal{B}_7 = \{(k_1, f_{12}, k_2, (f_{10} \times f_{11})), f_3 \vee (f_3 \wedge f_5) = true, f_4 = false\}$ ;
- $\mathcal{B}_8 = \{(k_1, f_8, k_2, f_7), f_4 = true, f_3 \vee f_5 \vee (f_3 \wedge f_5) = false\}$ ;
- $\mathcal{B}_9 = (k_1, f_{13}, k_2, (f_3 \vee f_5 \vee (f_3 \wedge f_5) \vee f_4)), (f_3 \vee f_5 \vee (f_3 \wedge f_5) = true, f_4 = false) \vee (f_4 = true \text{ then } f_3 \vee f_5 \vee (f_3 \wedge f_5) = false)$ ;
- $\mathcal{B}_1, \mathcal{B}_{2(*)}, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_{6(\times)}, \mathcal{B}_7, \mathcal{B}_8, \mathcal{B}_9 \in \mathcal{B}$

useful in determining XMI. Transforming SBVR into XMI is necessary for BPMN modeling. In the above cases, there are various parallelism complexity levels, i.e. nested conditions and non-free choice conditions. In the case of the complex gateway (see section 4.4) above, the formal SBVR model has a nested gateway. Another special condition that was found is the condition of non-free choice. This condition indicates the case of an inclusive gateway (see section 4.3). The four gateways, the authors have examined in each of the examples above represent the specific cases the authors have encountered in the university library.

The study uses the formalization based on the formal model integrating BPMN and XTT2, which is

## 5. Conclusion

The main contributions of this research are to introduce new formal model of inclusive gateway (OR gateway) and complex gateway in business rule using SBVR. The authors have successfully implemented the SBVR formal model for four gateways in several scenarios i.e. library material classification process, library exhibition process, procurement process, and book conservation process model. All scenarios represent the specific cases the authors have encountered in the university library. This formal model is represented by a process based on BPMN notation and business rules based on SBVR using the XTT2 method. The various parallelism complexity levels, i.e. nested conditions and non-free choice conditions successfully solved. The new formal model i.e. inclusive gateway (OR gateway) and complex gateway represent 66.7% BPMN gateways in business rule.

Our future work would be introducing formal rules of two other gateways, i.e. event-based gateway and the parallel event-based gateway. Thus, this would enable designer to seamlessly transform BPMN into SBVR.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, Irene RHT Tangkawarow and Riyanarto Sarno; methodology, Irene RHT Tangkawarow and Daniel Siahaan; software, Irene RHT Tangkawarow; validation, Irene RHT Tangkawarow, Riyanarto Sarno, and Daniel Siahaan; formal analysis, Irene RHT Tangkawarow, Riyanarto Sarno, and Daniel Siahaan; writing—original draft preparation, Irene RHT Tangkawarow; writing—review and editing, Riyanarto Sarno and Daniel Siahaan; supervision, Riyanarto Sarno and Daniel Siahaan.

## Acknowledgments

This work has been supported by Indonesia Endowment Fund for Education (LPDP) scholarship.

## References

[1] T. Skersys, L. Tutkute, and R. Butleris, “The enrichment of BPMN business process model with SBVR business vocabulary and rules”, *J. Comput. Inf. Technol.*, Vol. 20, No. 3, pp. 143–150, 2012.

- [2] T. Halpin, “ORM/NIAM Object-Role Modeling”, *Handbook on Architectures of Information Systems*, Springer, Berlin, Heidelberg, 1998, pp. 81–101.
- [3] T. Skersys, P. Danenas, and R. Butleris, “Extracting SBVR business vocabularies and business rules from UML use case diagrams”, *J. Syst. Softw.*, Vol. 141, pp. 111–130, 2018.
- [4] Omg, “Semantics of Business Vocabulary and Business Rules”, *VI.2*, Vol. 1.2, No. May, p. 422, 2013.
- [5] OMG, *Semantics of Business Vocabulary and Business Rules Version 1.4*, 2017.
- [6] S. Arshad, I. S. Bajwa, and R. Kazmi, “Generating SBVR-XML Representation of a Controlled Natural Language”, *Intelligent Technologies and Applications. INTAP 2018.*, 2019, Vol. 932, pp. 379–390.
- [7] B. Akhtar, A. Mehmood, A. Mehmood, and W. Noor, “Generating RDFS Based Knowledge Graph from SBVR”, *Communications in Computer and Information Science*, Vol. 932, pp. 618–629, 2019.
- [8] G. Aiello, R. Di Bernardo, M. Maggio, D. Di Bona, and G. Lo Re, “Inferring business rules from natural language expressions”, In: *Proc. of - IEEE 7th International Conf. on Service-Oriented Computing and Applications, SOCA 2014*, pp. 131–136, 2014.
- [9] M. Mohanan and P. Samuel, “Natural Language Processing Approach for UML Class Model Generation from Software Requirement Specifications via SBVR.”, *Int. J. Artif. Intell. Tools. World Sci. Publ. Co. Pte Ltd.*, 2018.
- [10] O. Tantan and J. Akoka, “Automated transformation of business rules into business processes From SBVR to BPMN”, In: *Proc. of the International Conf. on Software Engineering and Knowledge Engineering, SEKE*, Vol. 25, No. 4, pp. 684–687, 2014.
- [11] R. D. A. Rodrigues, L. G. Azevedo, and K. C. Revoredo, “BPM2Text: A language independent framework for Business Process Models to Natural Language Text”, *ISys - Brazilian J. Inf. Syst.*, Vol. 10, No. 1, pp. 38–56, 2016.
- [12] F. Lévy and A. Nazarenko, “Formalization of Natural Language Regulations through SBVR Structured English”, no. July. LIPN Paris13 University, Sorbonne Paris Cité & CNRS, 2013.
- [13] K. Kluza and K. Honkisz, “From SBVR to BPMN and DMN Models . Proposal of Translation from Rules to Process and Decision Models From SBVR to BPMN and DMN Models. Proposal of Translation from Rules to

- Process and Decision Models”, In: *Proc. of Artificial Intelligence and Soft Computing: 15th International Conf., ICAISC*, No. June, 2016.
- [14] I. R. H. T. Tangkawarow and J. Waworuntu, “A Comparative of business process modelling techniques”, In: *Proc. of IOP Conf. Series: Materials Science and Engineering*, 2016, Vol. 128, pp. 1–16.
- [15] A. Awadid, S. Nurcan, and S. Ayachi Ghannouchi, “On leveraging the fruits of research efforts in the arena of business process modeling formalisms: a map-driven approach for decision making”, *Softw. Syst. Model.*, Vol. 18, No. 3, pp. 1905–1930, 2019.
- [16] I. R. H. T. Tangkawarow, R. Sarno, and A. C. Fauzan, “Evaluation the Performance of Tax Determination Using Discrete Event Simulation”, In: *Proc. of 2nd Int. Conf. Informatics Comput. Sci. ICICoS 2018*, pp. 23–28, 2019.
- [17] S. Huda, R. Sarno, and T. Ahmad, “Fuzzy MADM Approach for Rating of Process-Based Fraud”, *J. ICT Res. Appl.*, Vol. 9, No. 2, pp. 111–128, 2015.
- [18] A. Raj, A. Agrawal and T. Prabhakar, “Transformation of Business Processes into UML Models: An SBVR Approach”, *Int. J. Sci. Eng. Res.*, Vol. 4, No. 7, pp. 647–661, 2013.
- [19] T. Skersys, K. Kapocius, R. Butleris, and T. Danikauskas, “Extracting business vocabularies from business process models: SBVR and BPMN standards-based approach”, *Comput. Sci. Inf. Syst.*, Vol. 11, No. 4, pp. 1515–1536, 2014.
- [20] E. Mickeviciute, R. Butleris, S. Gudas, and E. Karciauskas, “Transforming BPMN 2.0 business process model into SBVR business vocabulary and rules”, *Inf. Technol. Control*, Vol. 46, No. 3, pp. 360–371, 2017.
- [21] K. Kluza and G. J. Nalepa, “Formal Model of Business Processes Integrated with Business Rules”, *Inf. Syst. Front.*, Vol. 21, No. 5, pp. 1167–1185, 2019.
- [22] G. J. Nalepa, A. Ligęza, and K. Kaczor, “Overview of knowledge formalization with XTT2 rules”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Vol. 6826 LNCS, pp. 329–336, 2011.
- [23] G. J. Nalepa, *Modeling with Rules Using Semantic Knowledge Engineering*. Kraków, Poland: Springer Nature, 2018.
- [24] A. Šukys, L. Ablonskis, L. Nemuraitė, and B. Paradauskas, “A grammar for ADVANCED SBVR editor”, *Inf. Technol. Control*, Vol. 45, No. 1, pp. 27–41, 2016.
- [25] S. Huda, R. Sarno, T. Ahmad, and H. A. Santoso, “Identification of Process-based Fraud Patterns in Credit Application”, In: *Proc. of 2nd International Conf. on Information and Communication Technology (ICoICT) Perform*, pp. 84–89, 2014.
- [26] R. Sarno and K. R. Sungkono, “Hidden markov model for process mining of parallel business processes”, *Int. Rev. Comput. Softw.*, Vol. 11, No. 4, pp. 290–300, 2016.
- [27] O. C. Tantan and J. Akoka, “Automated transformation of Business Rules into Business Processes From SBVR to BPMN”, In: *Proc. of the 26th International Conf. on Software Engineering and Knowledge Engineering (SEKE)*, pp. 684–687, 2014.
- [28] K. Anand, S. Mitra, and P. K. Chittimalli, “Semantic search and query over SBVR-based business rules using SMT based approach and information retrieval method”, *ENASE 2019 - In: Proc. of 14th Int. Conf. Eval. Nov. Approaches to Softw. Eng.*, pp. 47–58, 2019.