

# A Ontology-based Approach for Business Process Compliance Checking

Tuan Pham, Nhan Le Thanh

► **To cite this version:**

Tuan Pham, Nhan Le Thanh. A Ontology-based Approach for Business Process Compliance Checking. IMCOM '16 - the 10th International Conference on Ubiquitous Information Management and Communication, Jan 2016, Da Nang, Vietnam. ACM SIGAPP, pp.1 - 6, 2016, <<http://www.icuimc.org/2016/index.html>>. <10.1145/2857546.2857603>. <hal-01401769>

**HAL Id: hal-01401769**

**<https://hal.inria.fr/hal-01401769>**

Submitted on 23 Nov 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Ontology-based Approach for Business Process Compliance Checking

Tuan Anh Pham  
WIMMICS – INRIA Sophia Antipolis  
2004 Route des Lucioles, 06902  
Valbonne, France  
tuan-anh.pham@inria.fr

Nhan Le Thanh  
WIMMICS – INRIA Sophia Antipolis  
2004 Route des Lucioles, 06902  
Valbonne, France  
nhan.le-thanh@inria.fr

## ABSTRACT

The early detection of flaws and errors has become a significant feature of a business process modeling tool. This paper proposes an ontology-based approach for business process compliance checking. The business processes and the business rules are represented in a machine understandable form, a reasoner is used to reason on this knowledge base for detecting the potential semantic error by using a set of predefined rules.

## Keywords

Ontology Reasoning, Business Process Compliance, Business Rule

## 1. INTRODUCTION

Business Process Management (BPM) has become more and more important with the methods, techniques and software developed to design, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information [1]. Organizations have understood the cost effectiveness of BPM that emerges from flaw detection and automation of business processes [2] [3].

To be able to validate a business process, one of the challenges is to check the compliance of business process with a set of business rules and to create the correspondence between them automatically. The other challenge is that the business process and the business rules may be modified during the runtime because of changing in enterprise's policies. Therefore, the challenges for system designers are to build a flexible intelligent system which accepts and verifies the change on the business process and the business rules automatically. It can also work as a rule-based system which can reason and deduce a new knowledge or new decision based on a set of rules and facts.

In this paper, an ontology-based approach for business process modeling and checking is proposed, it gives full play to the excellence of ontology's semantic representation ability and automation reasoning. The business process model has been designed by using Coloured Petri Net (CPN) [12] and translated into a set of axioms in Business Process Ontology (BPO). The business rules have been also translated into a set of axioms in Business Rules Ontology (BRO). After finishing the translation, the consistency of the business process model and the business rule has been validated by a reasoner. The system can detect potential flaws automatically either at design time or during runtime. If the user designs or modifies a business process and this action causes a conflict with the business rules, the system will notify automatically.

Our contributions in this paper are:

- An ontology-based method for modeling a domain ontology and business rules.
- Proposing a method for detecting the potential error of business processes automatically during design time and runtime.

The rest of the paper is organized as follows. Section 2 analyses the differences with previous works. Section 3 presents some basic definition to help the reader to understand this paper. Section 4 presents the methodology for modeling a business process. Section 5, we will introduce about the content of business rule ontology (ontology domain). A methodology of potential semantic error detection in a business process is introduced in section 6. Section 7, we provide an example. We will do some evaluation in section 8. In section 9, some discussions about the addressed topics, some conclusions and future research directions are presented.

## 2. RELATED WORKS

Detecting the potential semantic error of business processes is a challenging task: the number and complexity of business rules is increasing and the rules are subject to constant change. This calls for a holistic approach to manage compliance. Becker et al. define business process compliance management (BPCM) as “the steady modeling, refinement, and analysis of business processes regarding the fulfillment of regulatory compliance” [27]. Ramezani et al. outline the interdependencies between BPM and Compliance Management (CM) and describes CM as a “methodology to elicit, specify and formalize, implement, check and analyze, and optimize compliance requirements in organizations” [28]. Works on Business process compliance have focused on examining whether a given process model is compliant with a certain reference model/pattern. On the technical aspect, the business process pattern initiative has identified various patterns for the specification of control-flow [17], data-flow [18], and resources [18] in business process management systems. The work in [19] deals also with the planning layer by formalizing process patterns using UML concepts. These compliance works have focused on the structural level of process models, while another line of works focuses on the combination of data and structure [20, 21, 22, 23, 24]. The frame-works in [25, 24, 26], for example, provide general compliance criteria for assessing the compliance of processes with semantic constraints. In addition, some compliance works aimed at supporting specific purposes, for example: correcting process models at design time [25], verifying changes in existing models [25], identifying compliance in the context of process mining [26], and identifying violations of execution order compliance rules [20].

In our approach, we take advantage of CPN's color set for checking the compliance of processed data in data-flow with the constraints in BRO. A color set can be defined in many types: int, string or

object. When firing a transition, the value of each color can be changed, the new value must respect to the constraint in BRO. This work will be explained in more detail in section 3.

On the other hand, for representing the business rules, some works use an ontology approach [8] [9] [10] [11]. They translate the Semantic of Business Vocabulary and Rule (SVBR) [16] vocabulary to OWL [29] and Semantic Web Rule Language (SWRL) [15], they provide the mapping or the rule in order to translate each property of SVBR (the definition of OMG) to a set of axioms in an ontology. In our approach, we classify the business rule into five main type of business rules, and we use Attempto Controlled English (ACE) [13] for defining the business rule, each ACE phrase will be translated into an axiom in BRO. We consider not only the business rule representation aspect but also the compliance of business process with a set of business rules. The advantage of our approach is to ensure that the business process is well-defined at design time and executed correctly during runtime. We must consider this aspect because a business process must always respect the predefined rules in BRO. Therefore, the difference between our work and the previous ones, is that we allow the system to check not only the consistency of business process and business rule but also the consistency of the integration between them.

### 3. BACKGROUND

#### 3.1 OWL ontology and Semantic Web Rule Language

##### 3.1.1 OWL

Ontology is about the exact description of things and their relationships. The Web Ontology Language (OWL) [29] is a family of knowledge representation languages for authoring ontologies. It was designed to provide a common way to process the content of web information (instead of displaying it). OWL was also designed to be read by computer applications (instead of humans).

##### 3.1.2 SWRL

The Semantic Web Rule Language (SWRL) [30] is a proposed language for the Semantic Web that can be used to express rules as well as logic, combining OWL DL or OWL Lite with a subset of the Rule Markup Language.

The rule is represented in this form:

$$\text{antecedent} \Rightarrow \text{consequent}$$

where both antecedent and consequent are conjunctions of atoms written  $a1 \wedge \dots \wedge an$ . Variables are indicated using the standard convention of prefixing them with a question mark (e.g., ?x). Using this syntax, a rule asserting that the composition of parent and brother properties implies the uncle property would be written:

$$\text{parent}(?x,?y) \wedge \text{brother}(?y,?z) \Rightarrow \text{uncle}(?x,?z)$$

In this syntax, built-in relations that are functional can be written in functional notation, i.e.,  $op:\text{numeric-add}(?x,3,?z)$  can be written instead as  $?x = op:\text{numeric-add}(3,?z)$

#### 3.2 Coloured Petri Nets

**Definition 1.** A net is a tuple  $N = (P, T, A, \Sigma, C, N, E, G, I)$  where:

- $P$  is a set of places.
- $T$  is a set of transitions.
- $A$  is a set of arcs

In CPNs sets of places, transitions and arcs are pairwise disjoint

$$P \cap T = P \cap A = T \cap A = \emptyset$$

$\Sigma$  is a set of color sets defined within CPN model. This set contains all possible colors, operations and functions used within CPN.

$C$  is a color function. It maps places in  $P$  into colors in  $\Sigma$ .

$N$  is a node function. It maps  $A$  into  $(P \times T) \cup (T \times P)$ .

$E$  is an arc expression function. It maps each arc  $a \in A$  into the expression  $e$ . The input and output types of the arc expressions must correspond to type of nodes the arc connected to.

Use of node function and arc expression function allows multiple arcs connect the same pair of nodes with different arc expressions.

$G$  is a guard function. It maps each transition  $t \in T$  into guard expression  $g$ . The output of the guard expression should evaluate to Boolean value true or false.

$I$  is an initialization function. It maps each place  $p$  into an initialization expression  $i$ . The initialization expression must evaluate to multiset of tokens with a color corresponding to the color of the place  $C(p)$ .

### 3.3 Business Process Compliance

Business process compliance checking is the set of activities an enterprise does to ensure that its core business does not violate relevant regulations, in the jurisdictions in which the business is situated, governing the (industry) sectors where the enterprise operates. The activities an organisation does to achieve its business objectives can be understood as business processes, and consequently they can be represented by business process models. On the other hand a normative document (e.g., a code, a bill, an act) can be understood as a set of clauses, and these clauses can be represented in an appropriate formal language. Business process compliance is a relationship between the formal representation of a process model and the formal representation of a relevant regulation.

### 4. BUSINESS PROCESS ONTOLOGY

As mentioned above, we use Color Petri Nets for modeling business processes; In this section, we introduce the method of building BPO, this method helps the user to represent a business process by an ontologies. The advantage of this method is to allow the user to check the consistency business process automatically by the reasoning. The TBox of BPO is defined as follow:

1.  $CPN \sqsubseteq \exists I \text{ hasPlace.Placement} \sqsupseteq I \text{ hasTransition.Transition} \sqsupseteq I \text{ hasInputArc.InputArc} \sqsupseteq I \text{ hasOutputArc.OutputArc}$
2.  $Place \sqsubseteq \exists 0 \text{ hasToken.Token} \sqsupseteq I \text{ hasArc.(InputArc} \sqcap \text{OutputArc)}$
3.  $Transition \sqsubseteq \exists I \text{ hasInputArc.InputArc} \sqsupseteq I \text{ hasOutputArc.OutputArc} \sqsupseteq I \text{ hasGuardFunction.Expression}$
4.  $InputArc \sqsubseteq I \text{ hasSourcePlace.Placement} \sqsupseteq I \text{ hasTargetTransition.Transition} \sqsupseteq I \text{ hasExpression.Expression}$
5.  $OutputArc \sqsubseteq = I \text{ hasTargetPlace.Placement} \sqsupseteq I \text{ hasSourceTransition.Transition} \sqsupseteq I \text{ hasExpression.Expression}$

CPN is the concept for representing all CPN graphs. A CPN graph is well defined if and only if it has at least one place, one transition, one input arc and output arc. We define in BPO following classes: *CPN*, *Place*, *Transition*, *OutputArc*, *InputArc* and some properties which define the relations between them, *hasPlace*, *hasTransition*, *hasInputArc*, *hasOutputArc*. *Place* represents the properties of place, we define a concept *Place*. A place may have a token or not, it has also at least one *InputArc* or one *OutputArc*. The concept

*Transition* is defined for all transitions. A transition must have at least one *InputArc* and one *OutputArc*. It's one of the minimum conditions for having a well-defined CPN graph. A transition may have only one guard function or not. The concept *InputArc* defined for all input arcs. An input arc has only one source place and one target transition. It may be marked by only one expression or not. An *OutputArc* has only one source transition and only one target place. It also may have only one expression or not.

## 5. BUSINESS RULE ONTOLOGY

### 5.1 Classification of Business Rule

The different structural categories of business rules are (Wagner 2005):

1. **Integrity (or constraints)**; For example: Each company must have one and only one director.
2. **Derivation** (conditions resulting in conclusions); For example: Platinum customers receive a 5% discount. John Doe is a platinum customer. As a conclusion, John Doe receives a 5% discount.
3. **Reaction** (Event, Condition, Action, Alternative action, Post-condition); For example: An invoice is received. If the invoice amount is more than \$2,000 then a supervisor must approve it.
4. **Production** (condition, action); For example: If there are no defects in the last batch of cars then the batch is approved.
5. **Transformation** (change of state); For example: A man's age can change from 28 to 29, but not from 29 to 28.

### 5.2 Business Rules Ontology

In this section, we introduce the method of building a business rule ontology. As mentioned above, there are five type of rules. For each type of rule, we create a set of axioms in the BRO. We also introduce some addition rules for allowing the reasoner to reason on BRO and BPO to detect the potential semantic error automatically.

#### 5.2.1 Integrity Rule

The integrity rule have the same meaning with a constraint in the relational database. In table 1, we define the cardinality rules. It will be translated into a set of cardinality axiom inside BRO.

**Table 1. Integrity rules**

Rule Example	OWL and SWRL
something that owns at least 2 cars	ObjectMinCardinality(n R C)
something that owns at most 2 cars	ObjectMaxCardinality(n R C)
something that owns exactly 2 cars	ObjectExactCardinality(n R C)

#### 5.2.2 Derivation Rule

This kind of rule allow the system to deduce a new knowledge. If a set of fact satisfy the derivation rule, the reasoner will deduce a new fact from the existing facts. We use SWRL rule to represent this kind of rule. This is the advantage of ontology, the information is represent in a machine understandable form, so the system can reason on our information to make some proposition to the user.

**Table 2. Derivation rule**

Rule Example	OWL and SWRL
platinum customers receive a 5% discount. John Doe is a platinum customer. As a conclusion, John Doe receives a 5% discount	PaltiumCustomer(w)-> hasDiscount(x,5)

#### 5.2.3 Reaction Rule

One of the important rule is the reaction rule which allows the user to define the relationship between a set of actions in a specific domain. We propose six kind of relationships: dependency, parallel execution, choice execution, sequential exclusion, parallel exclusion and choice exclusion.

**Table 3. Reaction rule**

Rule Example	OWL and SWRL
Task A is depended on task B	hasDependencyTask(B,A) TransitiveObjectProperty
Task A exclude task B in sequential	hasExSequentialTask(B,A) SymmetricObjectProperty
Task A exclude task B in parallel	hasExParallelTask(B,A) SymmetricObjectProperty
Task A exclude task B in choice	hasExChoiceTask(B,A) SymmetricObjectProperty
Task A must be executed in parallel with task B	hasParallelTask(B,A) SymmetricObjectProperty
Task A must be executed in choice with task b	hasChoiceTask(B,A) SymmetricObjectProperty

#### 5.2.4 Production Rule and Transformation Rule

This kind of business rules is represented in the form "IF something DO something". For representing this form with OWL language, we use SWRL [30].

**Table 4. Production rule and Transformation rule**

Rule Example	OWL and SWRL
If something do something	PaltiumCustomer(w)-> hasDiscount(x,5)

#### 5.2.5 Addition Rule

In order to detect the potential semantic error inside a business process, we add into the BRO a set of addition rules. It defines the rule which allows the reasoner to reason and find the potetial semantic error. It can dedect not only the neighbour step in a business process, but also the step which can find by using a chain reasoning.

**Table 5. Reaction rule**

Rule Example	OWL and SWRL
Task A is depended on task B	hasDependencyTask(A,B) $\wedge$
Task A exclude task B in sequential	hasExSequentialTask(B,A) ->Error(A)

Task A must be executed in parallel with task B Task A exclude task B in parallel	hasParallelTask(A,B) $\wedge$ hasExParallelTask(B,A) ->Error(A)
Task A must be executed in choice with task B Task A exclude task B in choice	hasChoiceTask(A,B) $\wedge$ hasExChoiceTask(B,A) ->Error(A)
Task A must be executed in parallel with task B Task B must be executed in parallel with task C $\Rightarrow$ Task A must be executed in parallel with task C	ObjectPropertyChain(hasExParallelTask parallelTask hasExParallelTask) hasExParallelTask
Task A exclude task B in parallel Task A exclude task B in sequential Task A exclude task B in choice	hasExParallelTask(B,A) hasExChoiceTask(B,A) $\wedge$ hasExSequentialTask(B,A) $\wedge$ ->Error(A)
Task A is depended on task B Task A must be executed in choice with task B	hasDependencyTask(A,B) $\wedge$ hasChoiceTask(B,A) - >Error(A)
Task A must be executed in parallel with task B Task A is depended on task B	hasDependencyTask(A,B) $\wedge$ hasParallelTask(B,A) - >Error(A)
Task A must be executed in choice with task B Task A must be executed in parallel with task B	hasChoiceTask(A,B) $\wedge$ hasParallelTask(B,A) - >Error(A)
Define class Tast is a disjoint class of the class Error	DisjointClasses(ekiiip:Action ekiiip:Error)

## 6. DETECTING THE POTENTIAL SEMANTIC ERROR

In Fig 1, we introduce the sketch of our solution. Business processes (CPN graph) are represented by an individual of the correspondence concept in BPO. Business rules are created and modified by an editor. Each rule is represented by a set of axioms and SWRL rule inside BRO. In order to check the compliance of business process with business rules, we merge BRO and BPO into one ontology; two concepts Transition in BPO and Task in BRO are defined as two equivalence concepts. The business term individuals can be used as a color and a token in CPN graph (business process). During the execution of business process, the value of individual can be changed but it must respect the constraints inside BRO (TBox and Properties).

At design time, when a user defines a business process, the business term will be used to name an item. Each transition individual in BPO is equivalent to an action individual in BRO. Depending on the user's given order, BPO editor will generate a set axioms inside BPO.

For example: there are two tasks inside BRO, which is defined that b depends on a as follow:

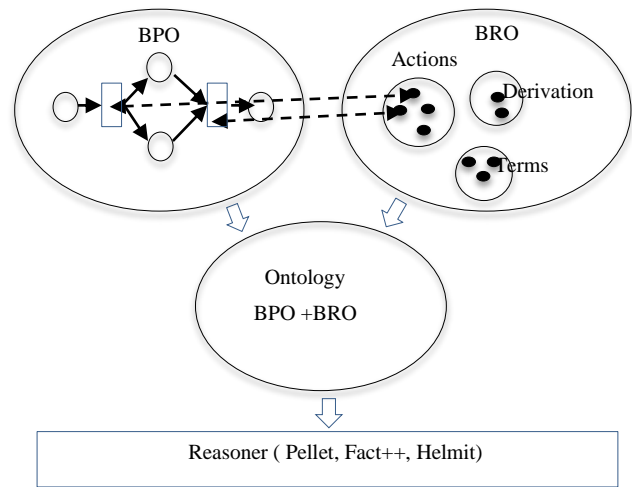
*ObjectPropertyAssertion(:hasDependencyTask a:b)*

It means that a must be executed before b, but at design time a user define that a executes after b, and the rule is generated as follow:

*ObjectPropertyAssertion(:hasDependencyTask a:b)*

Two rules above are opposite, so the merged ontology of BRO and BPO will be inconsistent. It can be checked by a reasoner (Pellet, Helmit). Because the property *hasDependencyTask* is defined as a *TransitiveObjecProperty* in the reaction rule 1 in table 3.

At runtime, we use the same approach to verify the consistency of merged ontology. If a user modifies a business process, the modification will be generated and insert into BPO; for each modification, the reasoner will check the consistency of merged ontology and notify the result to the user automatically.



**Figure 1. Checking the compliance of business process with a set of business rules.**

In table 6, the list of potential semantic error which can be detected by our approach is presented. It allows the designer to avoid the semantic error at the design-time. It help to reduce the time of building a system and to ensure the correctness of this system in the early step of development.

**Table 6. Type of detected errors**

Test case	BPO	BRO
1	$T_0$ depend $T_i$ with $1 \leq i \leq n$	$T_0$ seqExc $T_1$ and $T_i$ depends $T_1$ with $2 \leq i \leq n$
2	$T_0$ depend $T_i$ with $1 \leq i \leq n$	$T_0$ seqExc $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$
3	$T_0$ parallel $T_i$ with $1 \leq i \leq n$	$T_0$ paraExc $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$
4	$T_0$ parallel $T_i$ with $1 \leq i \leq n$	$T_0$ choiceExc $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$
5	$T_1$ paralExc $T_i$ with $1 \leq i \leq n$	$T_1$ parallel $T_i$ with $2 \leq i \leq n$
6	$T_1$ sequenExc $T_i$ with $1 \leq i \leq n$	$T_1$ depends $T_i$ with $2 \leq i \leq n$

7	$T_1$ choiceExc $T_i$ with $1 \leq i \leq n$	$T_1$ choice $T_i$ with $2 \leq i \leq n$
8	$T_1$ sequenExc $T_2$	$T_1$ choiceExc $T_2$ , $T_1$ parallelExc $T_2$
9	$T_0$ depend $T_i$ or $T_0$ choice $T_i$ or with $1 \leq i \leq n$	$T_0$ parallel $T_1$ and $T_i$ depends $T_1$ with $2 \leq i \leq n$ Or $T_0$ parallel $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$
10	$T_0$ depend $T_i$ or $T_0$ parallel $T_i$ with $1 \leq i \leq n$	$T_0$ choice $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$
11	$T_0$ choice $T_i$ with $1 \leq i \leq n$ $T_0$ depend $T_i$ with $1 \leq i \leq n$	$T_0$ depends $T_1$ and $T_i$ parallel $T_1$ with $2 \leq i \leq n$

## 7. EVALUATION

We implemented this approach and did some unit test on our business process repository. In table 7, we provide three main test which were done by a computer Core I7, RAM 16 GB.

**Table 7. Evaluation**

BPO + BRO Ontology	Inconsistency	Consistency
20 concepts 15 properties 30 reaction rules (7 SWRL) 30 tasks	1s	2s
20 concepts 15 properties 80 reaction rules (7 SWRL) 80 tasks	1s	1,2 minute
20 concepts 15 properties 400 reaction rules (7 SWRL) 400 tasks	2s	48 minutes

In table 7, in order to verify the consistency of business process model and there are no potential error, it may take long times and depend on the size of BPO and BRO.

## 8. CONCLUSION

In this paper, an ontology-based approach for detecting the potential semantic error of business process and business rules is proposed. It takes important features of the ontology which are the reasoning capabilities, the possibility to express complex actions, and its declarative semantics to validate not only the consistency of business rules and business process but also the compliance of business process with a set of business rules. The advantage of this approach is to allow the system to detect the flaws of business process automatically at design time and run time by using the

ontology's reasoning capabilities. Nevertheless, by using this approach, if BRO has many concepts and properties, the reasoning may take long time for checking the consistency of BPO and BRO ontology. According to that, future theoretic works involve three main issues. The first one is to focus on the distributed reasoning. The second one will be achieved by selecting the related rule of an action for the validation. And the last goal is to consider the business process execution and work with a data source.

## 9. REFERENCES

- [1] Ryan K.L. Ko, Stephen S.G. Lee, and Eng Wah Lee, "Business process management (BPM) standards: a survey," Business process Management Journal, vol. 15, pp. 744--791, 2009.
- [2] Marc Fasbinder, "Why model business processes", 2007.
- [3] L. J. Hommes, "The Evaluation of Business process Modeling Techniques," Delft University of Technology, Ph.D. thesis 90-9017698-5, 2004.
- [4] Craig Schlenoff, Michael Gruninger, Florence Tissot, John Valois, Taddle Creek Road, Steptools Inc, Josh Lubell, Jintae Lee, "The Process Specification Language (PSL) Overview and Version 1.0 Specification, 1999.
- [5] Jos de Bruijn, Holger Lausen, Axel Polleres, Dieter Fensel: The Web Service Modeling Language WSML: An Overview. ESWC 2006: 590-604, 2006.
- [6] Armin Haller, Mateusz Marmolowski, Eyal Oren, Walid Gaaloul, "oXPDL: a Process Model Exchange Ontology", 2007.
- [7] Heinrich Herre, "General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling", Theory and Applications of Ontology: Computer Applications 2010, pp 297-345.
- [8] Emiliano Reynares, Ma Laura Caliusco, Ma Rosa Galli, "An automatable approach for SBVR to OWL 2 mappings", 2013.
- [9] Gintare Bernotaityte, Lina Nemuraite, Rita Butkiene, and Bronius Paradauskas, "Developing SBVR Vocabularies and Business Rules from OWL2 Ontologies", ICIST, 2013.
- [10] Emiliano Reynares, Ma Laura Caliusco, Ma Rosa Galli, "SBVR to OWL 2 Mappings: An Automatable and Structural-Rooted Approach", In CLEI ELECTRONIC JOURNAL, 2014.
- [11] Gintare Bernotaityte, Lina Nemuraite, Rita Butkiene, Bronius Paradauskas, "Developing SBVR Vocabularies and Business Rules from OWL2 Ontologies", Information and Software Technologies Communications in Computer and Information Science Volume 403, 2013, pp 134-145.
- [12] Liu Feng, Zhang Wei. Colored Petri net extended with price information and its applicaion[J]. Journal of Computer Applications; 2007, 201:2501-2503.
- [13] S'oren Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki — A Tool for Social, Semantic Collaboration. In Proceedings of the 5th International Semantic Web Conference, number 4273 in Lecture Notes in Computer Science, pages 736–749. Springer, 2006.
- [14] Tuan Anh Pham, Thi-Hoa-Hue Nguyen, Nhan Le Thanh: Ontology-based business process validation. RIVF 2015: 41-46.

- [15] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member Submission (May 21 2004), <http://www.w3.org/Submission/SWRL/>.
- [16] <http://www.omg.org/spec/SBVR/>
- [17] WMP Van Der Aalst, AHM Ter Hofstede, B. Kiepuszewski, and AP Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5-51, 2003.
- [18] N. Russell, A. ter Hofstede, D. Edmond, and W. van der Aalst. Workflow data patterns: Identification, representation and tool support. *Conceptual Modeling-ER 2005*, pages 353-368, 2005.
- [19] H.N. Tran, B. Coulette, and B.T. Dong. Broadening the use of process patterns for modeling processes. *Proc. SEKE, Knowledge Systems Institute Graduate School*, pages 57-62, 2007.
- [20] A. Awad, S. Smirnov, and M. Weske. Towards Resolving Compliance Violations in Business Process Models. *GRCIS. CEUR-WS.org*, 2009.
- [21] I. Weber, J. Homann, and J. Mendling. Semantic business process validation. In *Proc. of International workshop on Semantic Business Process Management*, 2008.
- [22] G. Governatori, Z. Milosevic, and S. Sadiq. Compliance checking between business processes and business contracts. In *Enterprise Distributed Object Computing Conference, 2006. EDOC'06. 10th IEEE International*, pages 221-232. IEEE, 2006.
- [23] L. Thom, M. Reichert, C. Chiao, C. Iochpe, and G. Hess. Inventing Less, Reusing More, and Adding Intelligence to Business Process Modeling. In *Database and Expert Systems Applications*, pages 837-850. Springer, 2008.
- [24] M. El Kharbili, S. Stein, I. Markovic, and Pulverm E. Towards a framework for semantic business process compliance management. In *Proceedings of the workshop on Governance, Risk and Compliance for Information Systems*, pages 1-15. Citeseer, 2008.
- [25] F. Arbab, N. Kokash, and S. Meng. Towards using reo for compliance-aware business process modeling. *Leveraging Applications of Formal Methods, Verification and Validation*, pages 108-123, 2009.
- [26] T. Gschwind, J. Koehler, and J. Wong. Applying patterns during business process modeling. In *BPM*, volume 5240, pages 4-19. Springer, 2008.
- [27] Becker, J., Delfmann, P., Eggert, M., Schwittay, S.: Generalizability and Applicability of Model-Based Business Process Compliance-Checking Approaches – A State-of-the-Art Analysis and Research Roadmap. *BuR - Business Research* 5, 221–247 (2012).
- [28] Ramezani, E., Fahland, D., van der Werf, J.M., Mattheis, P.: Separating Compliance Management and Business Process Management. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM Workshops 2011, Part II. LNBIP*, vol. 100, pp. 459–464. Springer, Heidelberg(2012)
- [29] <http://www.w3.org/TR/owl-features/>
- [30] <http://www.w3.org/Submission/SWRL/>
- [31] Tuan Anh Pham, Nhan Le Thanh.: A Rule-Based Language for Integrating Business Processes and Business Rules. *The 9th International Web Rule Symposium (RuleML)*, Berlin, Germany, 2015