View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Checking the Compliance of Business Process in Business Process Life Cycle

Tuan Pham, Nhan Le Thanh

To cite this version:

Tuan Pham, Nhan Le Thanh. Checking the Compliance of Business Process in Business Process Life Cycle. 10th International Web Rule Symposium (RuleML 2016), RuleML, Jul 2016, New York, United States. hal-01401728

HAL Id: hal-01401728

https://hal.inria.fr/hal-01401728

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.

Checking the Compliance of Business Process in Business Process Life Cycle

Tuan Anh Pham and Nhan Le Thanh

WIMMICS-INRIA Sohphia Antipolis
2004 Route des Lucioles, 06902, Valbonne, France

{tuan-anh.pham, nhan.le-thanh}@inria.fr

Abstract. Business process compliance has become more and more important function for business process management (BPM). One of challenges in this area is to check the business process compliance in the business process life-cycle (design-time, run-time). In this paper, we propose a description logic-based approach for business process compliance checking during two phases of the business process life-cycle. In our approach, business process and the set of regulations are represented in a machine readable form. And we use that knowledge bases to check the compliance between them.

Keywords: Business Process Compliance, OWL 2, SWRL, Reasoner

1 Introduction

Business process compliance (BPC) checking in companies is a crucial feature for a BPM system. In essence, many approaches are developed to formally and automatically prove that business processes comply with relevant constraints like rules, laws. The requirement for business process compliance checking based on a set of constraints might emerge in different phases of the process life-cycle. During design time, the compliance of a process model with a set of constraints is checked for detecting the structure error (infinite loop, i.e.) and the compliance violation in the early step. At runtime, the progress of a potentially large number of process instances is monitored to detect or even predict compliance violations. After studying most important publications on this topic, we focus on some following research questions:

- What phase of the business process life-cycle our research approach can apply?
- What kind of rule our approach can cover?
- How to adapt our solution to existing business process definition language and system?

After providing our analysis on some related works, each research question will be answered in section 4.

Our contributions in this paper are:

- Proposing a method for representing and integrating business process and regulatory, business rules or laws.
- Building a method for business process compliance checking.

This paper is organized as follows, section 2 reviews the related work for compliance checking of BPM. Section 3 provides our research methodology. Section 4, we present our solution. Finally, Section 6 concludes this paper and indicates next steps of our work.

2 Related Work

Checking the compliance of business processes is a challenging task: the number and complexity of business rules is increasing and the rules are subject to constant change. Becker et al. [27] define business process compliance management (BPCM) as "the steady modeling, refinement, and analysis of business processes regarding the fulfillment of regulatory compliance". 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 frameworks 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 Color Petri Net (CPN)'s [32] color set for checking the compliance of processed data in data-flow with the constraints in the set of business rules. 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 the set of predefined rules. This work will be explained in more detail in section 4.

On the other hand, for representing the business rules, some works use an ontology-based 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 also Attempto Controlled English (ACE) [13] for defining the business rule,

each ACE phrase will be represented by an axiom in business rule ontology. 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 a set of predefined rules during the business process life-cycle. Therefore, the difference between our work and the previous related works is that our solution allows the system to check not only the consistency of business process and business rule but also the consistency of the integration between them at design-time and run-time. And the other difference is to use the reasoning for preventing the potential sematic error of a business process instance which occurs on the deduced knowledge or the generated data during both phrases of business process life-cycle (design-time, run-time), this kind of error cannot be detected by some other business process compliance checking approach (query-based approach, i.e.).

3 Research Methodology

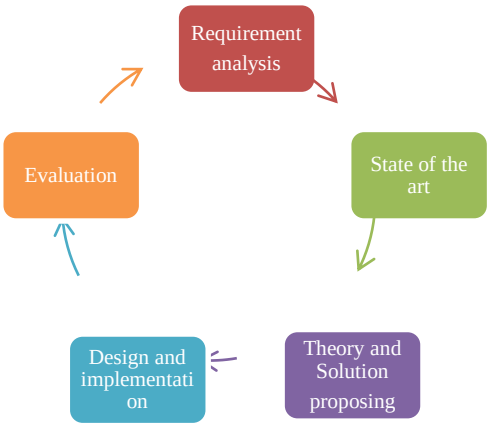

Fig. 1. Research Methodology

Initially, we started with requirements analysis based on case studies. We also formulated a state-of-the-art of business process compliance. Based on the state of the art, we choose a theoretical basis and propose a solution that is also implemented in prototypes. The solutions do not necessarily address all aspects of the requirements analysis at once but may also focus on certain aspects. Using the developed prototypes, we are able to analyze and evaluate developed solutions using data from practical applications. This may lead to further development and implementation iterations (e.g., in case the developed concepts do not yet cover all relevant aspects or do not yet yield adequate solutions). The evaluation of developed solutions may also result in a completely new iteration leading to modifications or refinements of the solution when studies reveal additional requirements.

4 Modeling Business Process Compliance

In this section, we will answer the mentioned research questions above. Fig 2 illustrates the sketch of our approach.

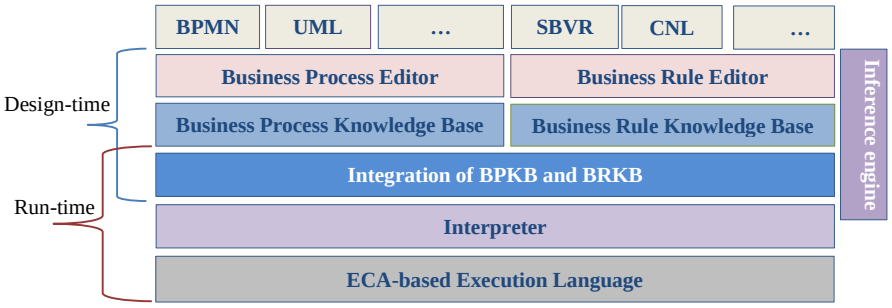

Fig. 2. Sketch of our solution

In Fig 2, a business process can be defined by some graphic design language (BPMN, UML, i.e.). Depending on the selected design language, the user's defined model will be translated into a Color Petri Net [32] (CPN) graph for checking the correctness of the model at the design time. In order to do that automatically, the CPN graphs are represented in a machine readable form. Description logic and first order logic are chosen to describe the model and the constraint for ensuring the structure correctness of the business process model. An inference engine (reasoner) is used to reason and verify the consistency of the model's knowledge base. If the knowledge base is consistent, the business process is validated. On the other hand, a set of business rules is represented in a business rule knowledge base. The inference engine is used to check the consistency of this knowledge base for ensuring that there are no conflicts inside the set of business rules. After having two consistent knowledge bases, we combine them into one knowledge base. The combined knowledge base is used for checking the compliance of a business process instance with a set of business rules inside business rule knowledge base.

4.1 Representation of Business Process Knowledge Base

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

- $CPN \sqsubseteq \geqq 1 hasPlace.Place \sqcap \geqq 1 hasTransition.Transition \sqcap \geqq 1 hasInputArc.InputArc \sqcap \geqq 1 hasOutputArc.OutputArc$
- $Place \sqsubseteq \geqq 0 hasToken.Token \sqcap \geqq 1 hasArc.(InputArc \sqcap OutputArc)$
- $Transition \sqsubseteq \geqq 1 hasInputArc.InputArc \sqcap \geqq 1 hasOutptArc.OutputArc \sqcap \leqq 1 hasGuard-Function.Expression$

- $InputArc \sqsubseteq 1hasSourcePlace.Place \sqcap =1hasTargetTransition.Transition \sqcap \leqq 1hasExpression.Expression$
- $OutputArc \sqsubseteq =1hasTargetPlace.Place \sqcap =1hasSourceTransition.Transition \sqcap \leqq 1hasExpression.Expression$

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. A business process instance is a set of individuals inside the business process ontology (BPO).

4.2 Representation of Regulatory

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.

Business Rules Ontology.

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

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.

Example 1:

The rule "Someone that owns at least 2 cars" is represented by an OWL 2 syntax as follow:

```
ObjectMinCardinality(2 hasCar SomeOne)
```

Derivation Rule.

This kind of rule allows the system to deduce a new knowledge. If a set of facts satisfies the derivation rule, the reasoner will deduce a new fact from the existing facts. We use SWRL rule to represent this kind of rule.

Example 2:

A rule "Platinum customers receive a 5% discount" is represented by a SWRL rule as follow:

```
PlatinumCustomer(w)->hasDiscount(x,5)
```

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 between the tasks: dependency, parallel execution, choice execution, sequential exlusion, parallel exclusion and choice exclusion. These relationships are represented by an OWL object property.

Example 3:

A rule "Task A is depended on task B" is represented by an object property as follow:

```
hasDependencyTask(B,A)
```

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].

Example 4: a rule "if a customer is a VIP member, they do not have to provide more information" is represented by a SWRL rule as follow:

```
isVIPMember(cus)-> not provideInfor(cus)
```

Transitive Rule.

We define a set of transitive rules inside the business rule ontology. This kind of error occurs after a sequence of activities. At design-time, when a user designs a business process instance, the user's defined business process will be combined with the set of transitive rules for detecting the potential semantic error when they use these chosen activities in a business process. At run-time, if there are some changes on the business process instance, transitive rule will be used for detecting the error which can occur with this change.

Rule Example	OWL and SWRL
Task A is depended on task B Task A exclude task B in sequential	hasDependencyTask(A,B) Λ hasExSequentialTask(B,A) -> Class(owl:Nothing)
Task A execute in parallel with task B Task A exclude task B in parallel	hasParallelTask(A,B) Λ hasExParallelTask(B,A) -> Class(owl:Nothing)
Task A execute in choice with task B Task A exclude task B in choice	hasChoiceTask(A,B) Λ hasExChoiceTask(B,A) -> Class(owl:Nothing)
Task A execute in parallel with task B Task B execute in parallel with task C => Task A execute in parallel with task C	ObjectPropertyChain(hasExParallelTask 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) Λ hasExSequentialTask(B,A) Λ -> Class(owl:Nothing)
Task A is depended on task B Task A execute in choice with task B	hasDependencyTask(A,B) Λ hasChoiceTask(B,A) -> Class(owl:Nothing)
Task execute in parallel with task B Task A is depended on task B	hasDependencyTask(A,B) Λ hasParallelTask(B,A) -> Class(owl:Nothing)
Task A execute in choice with task B Task A execute in parallel with task B	hasChoiceTask(A,B) Λ hasParallelTask(B,A) -> Class(owl:Nothing)

Table 1. Transitive Rule

4.3 Checking the Business Process Compliance Using Reasoning

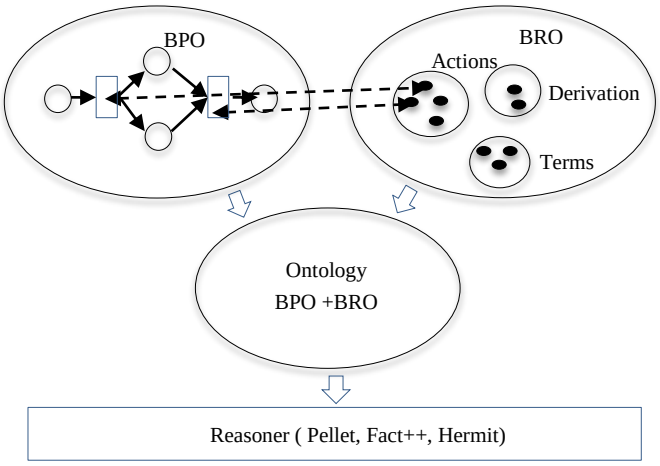

Fig. 3. Integration of two ontologies

In Fig 3, Business processes (CPN graph) are represented by a set of individuals 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 the change 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 :b:a)
```

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, Hermit). 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.

5 Conclusion

In this paper, an ontology-based approach for business process compliance checking 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 semantic flaws of business process automatically at design time and run-time. 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.

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 Brujin, 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 837850. 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 Man gement 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
32. Jensen, Kurt. Coloured Petri Nets (2 ed.). Berlin: Heidelberg. p. 234. ISBN 3-540-60943-1, 1996