A Semiautomatic Process Model Verification Plug-in based on Process Modeling Guidelines

Valter Helmuth Goldberg Júnior[1], Lucineia Heloisa Thom[1], and Diego Toralles Avila[1]

Department of Informatics, Federal University of Rio Grande do Sul, UFRGS, Porto Alegre, Brazil
{EMAILDOWALTER,lucineia,dtavila}@inf.ufrgs.br

Abstract. ¡Text of the summary of your article¿

1 Introduction

Business Process Management (BPM) is a discipline that provides a systematic approach to manage an organization's work by modeling, analyzing, improving and controlling its business processes (hereafter called processes, for simplification). It contributes to the increase of productivity and reduction of costs through more effective, more efficient and more adaptable processes [**?**].

Through the use of BPM, organizations continually seek to improve the quality of their processes. However, studies analyzing industry process model collections reveal that many models contain issues that harm its quality, such as control flow errors, badly designed structures and layouts or incorrect labeling [**?**] [**?**]. With process modeling being a key part of BPM, it is important that we try to prevent these issues if we are to have processes of better quality.

It is widely accepted that modeling a process is difficult [**?**]. This is usually due to the complexity of the modeling notation, it's many different elements and their respective semantics [**?**]. Choosing the appropriate representation depends upon the expertise or the guidance of an experienced modeler, which can greatly influence the quality of the resultant model. While the use of process modeling tools can help in this regard, they cannot guarantee a model's correctness nor it's comprehensibility.

One way to solve this challenge and help beginner modelers is to consolidate the knowledge of experienced modelers into guidelines, whose purpose is to help the user reduce the complexity and the number of errors in a process model by restricting undesirable constructs from being introduced. Many guidelines have been proposed by both practitioners [10] [13] [**?**] and researchers [1] [6] [12] [3]. Once it is verified that a process model is following a set of guidelines, we can assume that it has good comprehensibility.

However, doing that doesn't make sense if the process model is not syntactically correct. Any knowledge extracted from an incorrect process model has its validity compromised because, while you may be able to understand it, you may doubt whether it is what the modeler intended to portray [**?**]. Therefore, one must check if a process model is correct before considering it's comprehensibility.

It is possible to verify the correctness of process models through different ways. One of these is by the means of ontologies, which has seen wide-spread use in research on information science. Ontology is the study of being, which seeks to represent the world in entities, categories and relations [4]. In a more practical setting, an ontology provides an approach to define types, properties and relations. Given this, we can an ontology for process models as a meta-model to verify any process model correctness.

Along these lines, the purpose of this paper is to show how the use ontologies may assist in the identification of problems that reduce a process model comprehensibility. To do that, we need to input a process model into a process model ontology and, after that, verify it using a set of guidelines, pointing out any problems in the model that can be improved upon.

This paper is organized as follows: Section 2 outlines previous works related to the verification of process models. Section 3 shortly introduces the basic concepts used in this paper. Section 4 displays PLACEHOLDER. Section 5 presents our case study and our results. Section 6 closes the paper with our conclusions.

2   Related Works

The verification of business process models is nothing new, in fact, there have been numerous papers and books published that address this. The difference is that most of these publications are concerned with issues of correctness of a process model. In [4], for example, the author proposes two different approaches to verifying soundness of a process model draw using Event-Process-Chains.

Evaluating and reducing the complexity of a process model, though, is harder to achieve. It is not possible to measure a process complexity directly and, because of this, many metrics have been proposed that try this indirectly. [?]. The validity of these metrics is evidenced through statistical experiments, where models are judged both by the metrics and by people with varying levels of modeling experience.

As it was mentioned before, there are numerous guidelines proposed for modeling business processes, so many that a lot of them are repeats that may only vary in small details. In [7], 27 unified guidelines have been derived from a systematic review about business process modeling quality from over a 100 proposed in the reviewed literature.

Some of the existing BPMN tools try to provide some support for creating good process models. Based on the guidelines found in the previous article, a study [11] was performed to test how extensive was the support of the popular BPMN tools im creating good models. From this, we can learn that the Signavio modeler tool provides the best amount of support for modeling processes using guidelines.

Signavio is actually very flexible in how it's guidelines are used. The entire set can be disabled or enabled to different degrees of enforcement. Yet, the tool cannot guarantee that those guidelines are useful, as there isn't any empirical evidence suggesting that their use contributes to models of better quality. It also

3 Basic Notions

The modeling task of BPM is often done using the Business Process Model and Notation (BPMN). BPMN was developed by the Object Management Group (OMG), with the purpose of consolidating the many existing notations for process models in a single standard. This standard should provide a easy to comprehend notation to all stakeholders [8]. However, BPMN does not teach modelers how to use it's elements in the creation of simple and expressive process models. The consequence of this is that it's hard to achieve a a good level of quality in BPMN process models.

This difficulty motivated the creation of many frameworks that try to define what a process model quality is and classify the different quality types that compose it. Examples of this are the SEQUAL Framework, the SIQ Framework and the Guidelines of Modeling (GoM).

One common quality that can be extracted from all of these frameworks is the comprehensibility of the process model, as one of it's primary goals is to be understood [**?**].

This Top-Down perspective, however, doesn't really help beginner modelers to achieve the desired quality in their models. For this reason, a Bottom-Up approach is more appropriate, involving guidelines. In [5], seven process modeling guidelines (7PMG) have been proposed that are "thought to be helpful in guiding users towards improving the quality of their models, in the sense that these are likely (1) to become comprehensible to various stakeholders and (2) to contain few syntactical errors". These guidelines have been built upon empirical insights and, as such, provide a short but meaningful set of rules in which the work presented in this paper has been built upon. They are as follows:

G1 Use as few elements in the model as possible.
G2 Minimize the routing paths per element.
G3 Use one start and one end event.
G4 Model as structured as possible.
G5 Avoid OR routing elements.
G6 Use verb-object activity labels.
G7 Decompose a model with more than 50 elements.

To verify these guidelines, we need to transform the model from the BPMN standard into a more formal one, which in this case is an ontology, because, in computer science, the most common definition of an ontology is an explicit and formal specification of a shared conceptualization [2, **?**, **?**]. The transformation between BPMN and an ontology is done through the use of the already existing *BPMN Ontology* [9]. This allows us to verify the process model, or more precisely its structure, through the use of an ontological reasoner. The Table 3 shows how the mapping is done.

4 Verifying Process Models based on Process Modeling Guidelines

4.1 Methodology

To fulfill the objective of the work represented in this paper, there are 5 steps that must be performed:

Table 1. BPMN ⇒ Ontology Mapping

BPMN	Ontology	Example
Element Type	OWL Class	Activity, Gateway
Element Instance	Individual Named	Task 1: Submit Report
Attribute	Object Property	Label="Name"
Attribute Value	Data Property	Name:String="Task 1: Submit Report"

1. Extract each individual element from a BPMN model.

2. Instantiate each extracted element into the BPMN Ontology.

3. Verify the integrity of the ontology of the instantiated model

4. Verify if the model obeys the defined set of modeling guidelines.

5. Show which guidelines the model does not follow.

To extract the elements from a model, we must first determine how the model is defined. There are a number of file extensions for BPMN models that are used by distinct BPMN modeling tools, but those are usually only readable by the tools they come from. Instead, we use the interchangeable format defined by OMG, which is simply a XML file with a specific schema and a .bpmn extension. From this file we use a Java program to extract each individual element of a model (its tasks, gateways, sequence flows, messages and others).

With the model's elements at hand, we use the OWL-API for Java to create individuals for each element, according to each type described by the BPMN Ontology. After this is completed, we can open the instantiated model in a ontology editor to verify its integrity using the editor's reasoner. We chose to use the Protégé, because it is a popular open-source option for editing ontologies.

To verify the model according to the 7PMG, a plug-in for Protégé was developed, in which each guideline proposed had to be represented. For most guidelines (G1, G2, G3, G5, G7), we can do this using a simple test performed on top of a model's metrics. For G4, we simplify the guideline by measuring the number of splits and joins for each type of gateway. If that number is different then G4 has been disobeyed. Finally, G6 involves natural language processing for analyzing the syntax of each label, which is something outside of the scope of the work presented in this paper, therefore it has been left out. Table 4.1 shows each rule and associated metric being tested. Finally, the results of the verification are shown using a another plug-in in Protégé. The result of the test of each guideline is show with a "True" or "False" value.

Table 2. Metrics tested for each guideline from 7PMG

7PMG	Metric
G1	Number of Elements > 30
G2	Highest Element Degree > 7
G3	Number of Start/End Events > 1
G4	Number of Splits $\neq$ Number of Joins
G5	Number of OR Gateways > 0
G6	Syntax Analysis (Not Implemented)
G7	Number of Elements > 30

5   Case Study and Results

6   Conclusion

6.1   Limitations

7   References

References

1. Becker, J., Rosemann, M., von Uthmann, C., Uthmann, C.V.: Guidelines of Business Process Modeling. Business Process Management 1806, 241–262 (2000), http://www.springerlink.com/content/drwfdxguupx7evxb/\nhttp://dx.doi.org/10.1007/3-540-45594-9_3

2. Borst, P., Akkermans, H.: An ontology approach to product disassembly. In: Knowledge Acquisition, Modeling and Management, pp. 33–48. Springer (1997)

3. Correia, A., Abreu, F.B.e.: Adding Preciseness to BPMN Models. Procedia Technology 5, 407–417 (2012), http://www.sciencedirect.com/science/article/pii/S2212017312004768

4. Mendling, J.: Metrics for Process Models. Metrics for Process Models 6(0), 103–133 (2008), http://link.springer.com/chapter/10.1007/978-3-540-89224-3_4

5. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7PMG). Information and Software Technology 52(2), 127–136 (2010)

6. Mendling, J., Neumann, G., Aalst, W.M.P.V.D.: On the Correlation between Process Model Metrics and Errors. 26th international conference on Conceptual modeling pp. 173–178 (2007), http://dl.acm.org/citation.cfm?id=1386985

7. Moreno-Montes de Oca, I., Snoeck, M.: Pragmatic guidelines for Business Process Modeling p. 70 (2014)

8. OMG (Object Management Group): BPMN Specification - Business Process Model and Notation (2015), http://www.bpmn.org/

9. Rospocher, M., Ghidini, C., Serafini, L.: An ontology for the Business Process Modelling Notation. In: Garbacz, P., Kutz, O. (eds.) Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS2014, September, 22-25, 2014, Rio de Janeiro, Brazil. vol. 267, pp. 133–146. IOS Press (2014), https://books.google.com.br/books?hl=en&lr=&id=oG7YBAAAQBAJ&oi=fnd&pg=PA133&dq=An+ontology+for+the+bu Pqy0xi5w http://dx.doi.org/10.3233/978-1-61499-438-1-133

10. Silver, B.: BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0 (2009), http://www.amazon.com/BPMN-Method-Style-levels-based-methodology/dp/0982368100

11. Snoeck, M., Moreno-Montes de Oca, I., Haegemans, T., Scheldeman, B., Hoste, T.: Testing a selection of BPMN tools for their support of modelling guidelines. In: Lecture Notes in Business Information Processing. vol. 235, pp. 111–125 (2015), http://www.scopus.com/inward/record.url?eid=2-s2.0-84858309669&partnerID=tZOtx3y1

12. Vanderfeesten, I., Reijers, H.A., Mendling, J., Van Der Aalst, W.M.P., Cardoso, J.: On a quest for good process models: The cross-connectivity metric. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5074 LNCS, 480–494 (2008)

13. White, S.a., Miers, D.: BPMN Modeling and Reference Guide (2008)