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]

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

Abstract: The abstract should summarize the contents of the paper and should contain at least 70 and at most 200 words. The text must be set to 9-point font size.

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 process modeling 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 (Silver, 2009) (White and Miers, 2008) (**?**) and researchers (Becker et al., 2000) (Mendling et al., 2007) (Vanderfeesten et al., 2008) (Correia and Abreu, 2012). 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 (Mendling, 2008). 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 metamodel 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 out-

lines 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 (Mendling, 2008), 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 (Moreno-Montes de Oca and Snoeck, 2014), 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 (Snoeck et al., 2015) 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 (OMG (Object Management Group), 2015). 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 these are the SEQUAL Framework, the Guidelines of Modeling (GoM) and, more recently, the SIQ framework, in which we base this work upon. The SIQ framework defines process model quality as factor of three basic quality types:

- **Syntactic Quality** identifies if a process model conforms to the rules defined by the notation used to create it. In other words, if a process model follows the syntax and the vocabulary of its modeling language, we can verify that process model and declare it correct. To do so, the verification must check the static proprieties of a process model - how different types of elements are used and combined - and its behavioral proprieties - the process modeled should not reach a deadlock and must be completed properly, i.e the process model is *sound*.

- **Semantic Quality** bears the connection between a process model and the real world process it's supposed to represent. Checking a process model's semantic quality is, basically, making sure it is valid - all elements of the process model correctly represent the real world - and complete - there are no real world process parts that are missing in the process model. This check is simply called validation and, if it passes, the process model is determined true.

- **Pragmatic Quality** characterizes the comprehensibility of a process model. It is the certification that a user's interpretation of a process model is equal to the actual, real world process. If done so, the process model is said to be understood.

Syntactic quality is the basis for the other two qualities. As mentioned before, it is not sensible to consider the comprehensibility of a process model if it is not syntactically correct. The same can be said

of its semantic quality. As such, the verification of a process model must be done before its validation or certification.

As previously explained, it's possible to do this verification using an ontology. More specifically, we can use an ontology design to serve as a meta-model for a process modeling notation. In the case of BPMN, there exists what is called the *BPMN Ontology* (Rospocher et al., 2014), which supports the mapping of a BPMN process model into elements of the ontology, while preserving the relations and strutures between the process model elements. Then, we can use an inference engine to verify the mapped model, which will check if the static propreties of BPMN model, , i.e its structure, is correct according to the BPMN syntax.

Finally, assuming the process model is indeed correct, we can try yo ensure its pragmatic quality, which, in this case, is done by checking via the use of process modeling guidelines. In (Mendling et al., 2010), 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. 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.

4 METHODOLOGY

To fulfill the purpose of this paper, we must create a series of steps that, with the assistance of an ontology, allows us to verify whether a process model follows the seven process modeling guidelines. However, before we can into this, there are a few questions we must answer.

To start with, we must decide how are the process models going to be represented. A few different notations for process models exist and each notation has different ways of how the model is coded within a file. As you can imply from our previous chapter, the notation used in this work will be BPMN 2.0. Models

Table 1: Parameters tested for each guideline from 7PMG

7PMG	Parameters
G1	Number of Elements > 30
G2	Highest Element Degree > 7
G3	Number of StartEvents > 1
	Number of End Events > 1
G4	Number of Splits $\neq$ Number of Joins
G5	Number of OR Gateways > 0
G6	Wordnet
G7	Number of Elements > 50

using this notation can usually be exported into the interchangeable format defined by OMG, which is simply a XML file with a specific schema and a .bpmn extension (OMG (Object Management Group), 2015). From this file, we can easily map elements from the model into the an ontology, via the *BPMN Ontology*.

Secondly, we need to establish how we are going to check whether or not a guideline is being followed or not by a model. We must express them is such a way that they turn into a yes or no question. To do that, for each guideline must identify parameters that can measure and compare that model against what each guideline says is best. Based upon previous works (Mendling, 2008); (**?**); (**?**), the table **??** presents the parameters utilized to check if the model violates each guideline.

With these parameters, two guidelines show problems: G1 and G6.

G1, the guideline for encouraging the use of less elements when modeling, becomes redundant with G7, which determines when a model should be decomposed. This happens because G1 is much more suited to be used when a model is being developed, when one can refrain from introducing a few elements instead of when the modeling is finished and the choice becomes of how the model can be decomposed. As a result, the parameters for both guidelines are very similar, requiring us to choose the more appropriate one for testing. In this case, since we are working with a model and not with modeling, the guideline G7 is more appropriate.

G6, which tells us to label activities in the verb-object style, is the second problematic guideline. In this case, the problem is the complexity of checking the language of each label. It exceeds the scope of this work, since it requires the use of Natural Language Processing to identify the words of each label and compare them and their use against a thesaurus to define context (**?**). Therefore, we are ignoring this guideline.

Finally, we must determine how are we going to load and edit an ontology. We chose to use the ontol-

Table 2: 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"

ogy editor *Protégé* [1]. *Protégé* not only can verify the integrity of ontologies using an inference engine, it is also easily extensible through the use of plugins.

4.1 Verifying Process Models Based on Process Modeling Guidelines

Given the decisions made in this previous section, we can define a method to verify the process models.

1. Load the BPMN Ontology into *Protégé*

2. Extract each individual element from a BPMN model.

3. Instantiate each extracted element into *Protégé* via OWL-API and using the BPMN Ontology.

4. Use *Protégé*'s inference engine to verify the integrity of the new ontology.

5. Check if the model's parameters obey the limits defined by the modeling guidelines.

6. For each guideline not followed, recommend model alterations to solve the issue.

The first step is simply preparatory. The BPMN ontology will support the mapping of elements from BPMN to the ontology by serving as the meta-model containing the structuring rules of BPMN.

The second step is done through a Java plugin for *Protégé*, which reads the .bpmn file of a process model, extracting each of its elements (its tasks, gateways, sequence flows, messages and others). After that, this same plugin does the third step, using the OWL-API to create individuals for each element, mapping and instantiating them according to each type described by the BPMN Ontology. The table **??** shows how the mapping is done.

After this is completed, the fourth step consists of opening the instantiated model in Protégé to verify its integrity. If this is successful, it can be assumed that the structure of the BPMN model mapped into Protégé is syntactically correct, since any syntactical error in the process model's structure would violate the ontology's integrity according to the BPMN Ontology.

_____
[1] http://protege.stanford.edu/

Table 3: Recommended Actions for each tested guideline

7PMG	Recommended Action
G2	Reduce the number of sequence flows connected to a single element
G3	Restructure the model to reduce the number of Start and End events
G4	Restructure the model to have the same number of Split and Joins
G5	Restructure the process model to remove the OR Gateways
G7	Decompose the process model

All that remains now is checking the process model according the guidelines. The fifth step uses another Java Plugin for *Protégé* that checks the process model's parameters. It then show recommendations according to the table 3 for the sixth step.

5 CASE STUDY AND RESULTS

6 CONCLUSIONS

6.1 Limitations

REFERENCES

Becker, J., Rosemann, M., von Uthmann, C., and Uthmann, C. V. (2000). Guidelines of Business Process Modeling. *Business Process Management*, 1806:241–262.

Correia, A. and Abreu, F. B. e. (2012). Adding Preciseness to BPMN Models. *Procedia Technology*, 5:407–417.

Mendling, J. (2008). Metrics for Process Models. *Metrics for Process Models*, 6(0):103–133.

Mendling, J., Neumann, G., and Aalst, W. M. P. V. D. (2007). On the Correlation between Process Model Metrics and Errors. *26th international conference on Conceptual modeling*, pages 173–178.

Mendling, J., Reijers, H. A., and van der Aalst, W. M. P. (2010). Seven process modeling guidelines

(7PMG). *Information and Software Technology*, 52(2):127–136.

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

OMG (Object Management Group) (2015). BPMN Specification - Business Process Model and Notation.

Rospocher, M., Ghidini, C., and Serafini, L. (2014). An ontology for the Business Process Modelling Notation. In Garbacz, P. and Kutz, O., editors, *Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS2014, September, 22-25, 2014, Rio de Janeiro, Brazil*, volume 267, pages 133–146. IOS Press.

Silver, B. (2009). *BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0.*

Snoeck, M., Moreno-Montes de Oca, I., Haegemans, T., Scheldeman, B., and Hoste, T. (2015). Testing a selection of BPMN tools for their support of modelling guidelines. In *Lecture Notes in Business Information Processing*, volume 235, pages 111–125.

Vanderfeesten, I., Reijers, H. A., Mendling, J., Van Der Aalst, W. M. P., and Cardoso, J. (2008). 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.

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