

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

Valter Helmuth Goldberg Júnior<sup>1</sup>, Lucineia Heloisa Thom<sup>1</sup> and Diego Toralles Avila<sup>1</sup>

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

**Keywords:** The paper must have at least one keyword. The text must be set to 9-point font size and without the use of bold or italic font style. For more than one keyword, please use a comma as a separator. Keywords must be titlecased.

**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, its 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 its 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, using guidelines to verify a process model doesn't make sense if it isn't 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 meta-model to verify any process model correctness.

Along these lines, the purpose of this paper is to show how the use of ontologies may assist in the identification of problems that reduce a process model comprehensibility. To do that, it is necessary to input a process model into a process model ontology and, after that, verify it using a set of process modeling 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 (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, many authors have proposed guidelines for modeling business processes, so many that it is likely they repeat guidelines already proposed, with only few small variations in the details. In (Moreno-Montes de Oca and Snoeck, 2014), these repeats were gathered from a systematic review about business process modeling quality from over a 100 proposed in the literature and turned into 27 unified guidelines.

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, at the time, the support of the popular BPMN tools in creating good models. From this, we can learn that the Signavio<sup>1</sup> modeler tool provides the best amount of support for modeling processes using guidelines.

## 3 BASIC NOTIONS

The modeling task of BPM is often done using the Business Process Model and Notation (BPMN).

<sup>1</sup>[www.signavio.com](http://www.signavio.com)

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 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 structures between the process model elements. Then, we can use an inference engine to verify the mapped model, which will check if the static properties 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 to 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 define a series of steps that, with the assistance of an ontology, allows us check a process model's pragmatic quality by verifying whether it follows Mendling's seven process modeling guidelines. However, before we can do this, there are a few things we must define.

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 can be implied from the previous section, the notation used in this work is BPMN 2.0. Models using this notation can 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,

Table 1: Parameters tested for each guideline from 7PMG

7PMG	Metrics and Comparisons
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 > 30

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 in such a way that they turn into a yes or no question. To do that, for each guideline must be related to a metric that can measure and compare a model against supposed optimal values. Based upon previous works (Mendling, 2008) (?), the table 1 presents the metrics and optimal values 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 the metric used for G1 is the same as the one used for G7. Because of this, we need to choose which guideline is more appropriate for our method. Since G1 is more suited to be used when a model is being developed and the modeler can refrain from introducing a few elements, instead of when the modeling is finished, 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 in the complexity of checking the language of each label. This 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 the label's 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 ontology editor *Protégé*<sup>2</sup>. *Protégé* not only can verify the integrity of ontologies using an inference engine, it is also easily extensible through the use of plugins.

<sup>2</sup><http://protege.stanford.edu/>

Table 2: BPMN  $\Rightarrow$  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"

#### 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. Our objective is to map and load BPMN models into an ontology using *Protégé* and then checking those models using Mendling's guidelines.

The first step is simply to prepare *Protégé* for instantiating BPMN models. To do this, we load BPMN ontology into the editor, so that it will support the mapping of elements from BPMN to the ontology by serving as the meta-model containing the structuring rules of BPMN.

Once that is done, we can begin extracting the individual elements from the BPMN models and instantiating each of them into the ontology. We developed a Java plugin for *Protégé* which reads the .bpmn file of the process models and extracts its tasks, gateways, sequence flows and messages. After that, this same plugin uses the OWL-API to create individuals for each element, mapping and instantiating them according to each type described by the BPMN Ontology. The table 2 shows how the mapping is done.

After this is completed, we can finally have the ontology assist us in checking the process model syntax. Using *Protégé* inference engine, we can verify the ontology's 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.

The final steps of this method are checking the process model according Mendling's guidelines and recommending alterations based upon the results. We developed another Java Plugin that checks the process model's metrics and, for each violated guideline, the plugin recommends actions according to the table 3.

The entire series of steps is as follow:

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.

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

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. Recommend model alterations for each guideline not followed.

## 5 CASE STUDY AND RESULTS

To validate our method, we applied it to a collection of 31 BPMN models. These are models of the the processes of a university, created by students of BPM, verified and corrected by their professor and semantically validated by each model's stakeholders.

For each guideline used in our method, we extracted the related metrics for analysis via statistics. The tables 4 and 5 show this. We can then use these statistics to predict whether most of the models of this collection follow or not the process modeling guidelines.

For example, in the statistics for Highest Connectivity degree, for guideline G2, we see... G5 is similar, since the average number of OR gateways is low, but not zero, and the standard deviation is almost 1, implying that a few models do use OR gateways, up to the maximum of 4. Because of this, we expect that a few process models do violate this guideline, and our method will show this.

For G3, on the other hand, the number of start and end events has the high average of more than 2 , compared to the recommended use of 1 event of each.

Table 4: Statistics for metrics related to the guidelines G2, G3, G5, G7

	Highest Connectivity Degree	Nº Start Events	Nº End Events	Nº OR Gateways	Nº Elements
Average		2.20513	2.25641	0.28125	24.79487
Std. Deviation		2.00236	1.96974	0.85135	17.71288
Minimum		1	1	0	6
Maximun		13	13	4	98
Median		2	2	0	21

Table 5: Statistics for metrics related to the guideline G4

	Splits - Joins Difference		
	AND	XOR	OR
Average	0.125	0.53125	-0.09375
Std. Deviation	0.33601	1.39085	0.39016
Minimum	0	-2	-2
Maximun	1	4	0
Median	0	0.5	0

This would imply that most process models from the collection do not follow G3.

Analysis for guideline G4 is more complicated, since we measure the difference between the two metrics related to it. Not only that, we also have to measure this difference for each different type of gateway. We can, however, clearly see the imbalance of the number of XOR splits versus the number of XOR joins by looking at the average of that metric. That information, together with the high standard deviation show for each gateway, indicates that most process models in the collection are not structured.

The statistics for G7 are slightly vague. The high, but not unreasonable, average for the measure of the number of elements suggest that the guideline is followed. Yet, the high standard deviation hints that at least some models do have more than 30 elements.

With all this information in mind, our expectations are that most process models violate guidelines G3 and G4, while following guidelines G2 and G5. Lastly, guideline G7 will have a few violations, but not enough to stand out.

Table 6: Number of violations Per guideline

	Total Violations	Percent of Total
G2	2	6.45%
G3	1	3.23%
G4	10	32.26%
G5	4	12.90%
G7	8	25.81%

Let's compare these conclusions with the results of our applied method, as show in table 6. According to it, our expectations for guidelines G2, G4, G5 and G7 do match the results. The results for guideline G3,

though, is completely unexpected.

Upon further analysis, we found a reason for this. Many process models of the collection have multiple pools or subprocesses, which, according to the notation, require new start and end events, causing a distortion in the number of events that a process model has. Therefore, we must take said distortion into account for our statistical analysis. Nevertheless, our plugin that verifies process models is doing this correctly.

Table 7: Number of models per quantity of violations

	Number of Models
No violations	12
One violation	14
Two violations	4
Three violations	1

## 6 CONCLUSIONS

Process models that follow process modeling guidelines are more likely to be understood by the process stakeholders. With the help of process modeling tools that support those guidelines,

### 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.
- Rospoche, 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*.