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

Valter Helmuth Goldberg Júnior[1], Lucineia Heloisa Thom[1] José Palazzo Moreira de Oliveira and Diego Toralles Avila[1]

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

Keywords: Business Process, BPMN, Ontology, Process Model Quality, Modeling Guidelines

Abstract: Designing process models easy to understand is a complex task. Process analysts must rely on the experience of good modelers to achieve process models with high comprehensibility, which is also know as pragmatic quality. In the literature, this experience is portrayed as process modeling guidelines, that help modelers avoid common issues that hinder the comprehension of the process model. In this paper, we show a method for semi automatically verifying business process models according to process modeling guidelines. This method uses the BPMN Ontology and the ontology editor *Protégé* to assist with verifying a process model's syntax before following with verifying its pragmatic quality. The plugin developed based on this method was applied to a collection of 31 process models, with the results show in our case study.

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 (van der Aalst, 2013).

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 process models contain issues that harm its quality, such as control flow errors, badly designed structures and layouts or incorrect labeling (Mendling et al., 2008) (Leopold et al., 2016). 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 (Mendling et al., 2010). This is usually due to the complexity of the modeling notation, its many different elements and their respective semantics (Leopold et al., 2016). Choosing the appropriate representation depends upon the expertise or the guidance of an experienced modeler, which can greatly influence the quality of the resultant process model. While the use of process modeling tools can help in this regard, they cannot guarantee a process 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) (Allweyer and Allweyer, 2010) 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 (Reijers et al., 2015). 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, 2008a). In a more practical setting, an ontology provides an approach to define types, properties and relations. Given this, we can use an ontology for process models as a meta-model to verify a process model's 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's 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 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 the context we choose to work with and the method we developed. 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, 2008a), 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 (Vanderfeesten et al., 2007) (Mendling, 2008b) (Gruhn and Laue, 2006). The validity of these metrics is evidenced through statistical experiments, where process models are judged both by the metrics and by people with varying levels of modeling experience (Cardoso, 2006) (Sánchez-González et al., 2008).

As it was mentioned before, many authors have proposed guidelines for modeling processes, so many that it is likely they repeat guidelines that had already been 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 process models. From this, we can learn that the Signavio [1] 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). 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 (Krogstie, 2012), the Guidelines of Modeling (GoM) (Schuette and Rotthowe, 1998) and, more recently, the SIQ framework (Reijers et al., 2015)), in which we base this work upon. The SIQ framework defines process model quality as made 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, then it is possible to verify that process model and declare it to be 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

[1] www.signavio.com

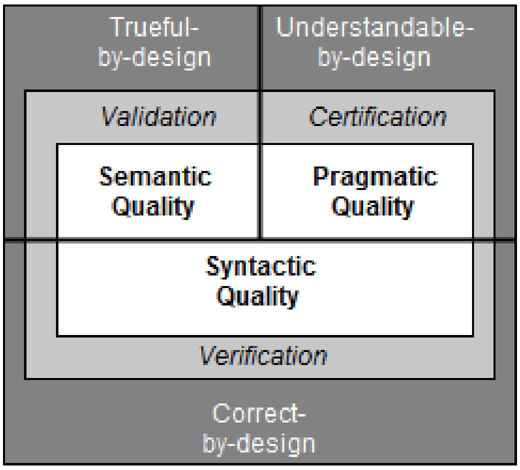

Figure 1: The SIQ Framework (adapted from (Reijers et al., 2015))

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 to be 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 metamodel 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. That way, we can use an inference engine to verify the mapped process 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 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:

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

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 A METHOD FOR VERIFYING PROCESS MODELS BASED ON PROCESS MODELING GUIDELINES

To fulfill the purpose of this paper, we must specify 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 process 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, we can easily map elements from the

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"

process 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 by a process model. We must express them in such a way that they turn into a binary, yes or no question. To do that, each guideline must be associated to a process model metric that can be measured and compared against supposed optimal values. Based upon previous works (Mendling, 2008a) (Recker, 2011) (Mendling et al., 2012), the table 1 presents the metrics and optimal values we considered to check if the process model violates each guideline. Based on these, 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 process 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 process 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, presents a problem 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 (Gassen et al., 2014). 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* [2]. *Protégé* not only can verify the integrity of ontologies using an inference engine, it is also easily extensible through the use of plugins. The majority of our method was built using this extensibility.

Having made these decisions, we specified a method to verify process models based on the seven process modeling guidelines. To do so, this method verify a process model syntax using an ontology and its inference engine and then check the process model using the guidelines.

[2]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 process model to reduce the number of Start and End events
G4	Restructure the process 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

The first step of the method is simply to prepare *Protégé* for instantiating BPMN models. To do this, the BPMN ontology is loaded 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.

Following this, the individual elements from the BPMN models can be extracted and instantiated into the ontology. We have 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.

Once the entire process model has been instantiated into the ontology, it can assist us in verifying if the process model is syntactically correct. Using *Protégé* inference engine, we can verify the ontology's integrity and, 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

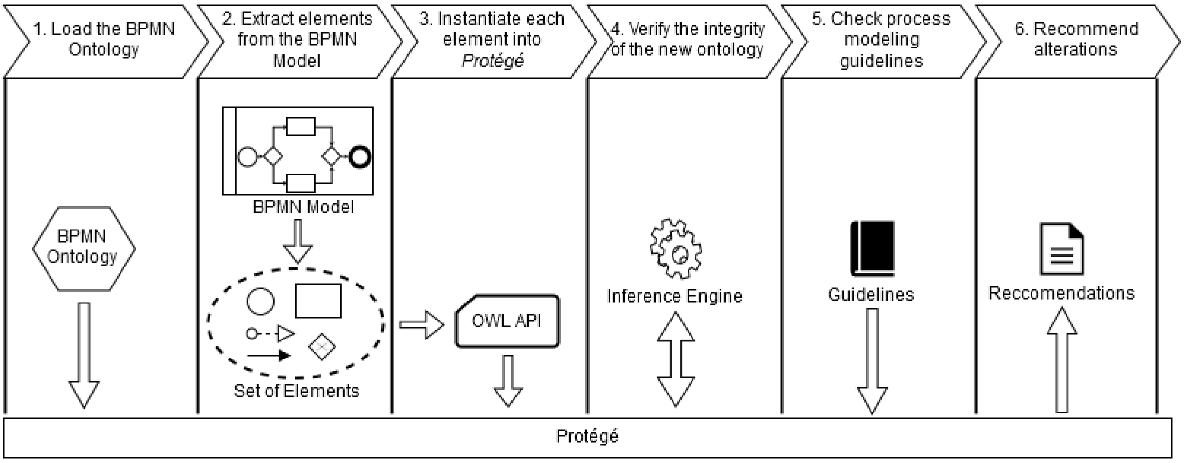

Figure 2: Steps to verify process models based on process modeling guidelines

plugin recommends actions according to the table 3.

The entire series of steps is as follow (also show in figure 2):

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 process model's metrics obey the limits defined by the modeling guidelines.
6. Recommend alterations to the process model 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 processes of a university, created by students of BPM, verified and corrected by their professor and semantically validated by each process stakeholders.

For each guideline used in our method, we extracted the associated metrics for a statistical analysis (as seen in tables 4 and 5). Based on these statistics, we have tried to predict whether most of the process models of this collection follow or not the process modeling guidelines.

For example, in the statistics for highest connectivity degree, associated with guideline G2, we see an average of NUM. If we sum this average with the standard deviation, we can see that it's unlikely, assuming a normal distribution, that a random process model picked from our collection will have more than the recommended threshold for this guideline, which is five. Consequently, we can anticipate that the high majority of process models from the collection will follow the guideline. 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 process models do use OR gateways, up to the maximum of 4. Because of this, we suppose that a few process models do violate this guideline, and our method will show this.

For G3, on the other hand, we see the opposite, as the number of both start and end events has the high average (more than 2), compared to the recommended use of 1 event of each. This would imply that most process models from the collection have multiple start and end event, ergo, they do not follow G3.

Analysis for guideline G4 is more complicated, since we define whether or not the process model is structured as the measure of the difference between the two metrics associated with it - number of splits and number of joins. Not only that, we also have to measure this difference for each different type of gateway. This balance means that the closer to zero is the average difference and the lower the standard deviation is then the more likely that the process models within the collection are structured. We can clearly see the opposite, however, since there is an imbalance of the number of XOR splits versus the number of XOR joins according to the average of that metric. Also, the high standard deviation show for each gateway indicates that most process models in the collection are not structured. Therefore, we predict that most process models of this collection infringe the guideline G4.

Finally, the statistics for G7 are slightly vague. The high, but not unreasonable, average for the mea-

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

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%

sure of the number of elements suggest that the guideline is followed. Yet, the high standard deviation hints that at least some process 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 a have a few violations, but not enough to stand out.

Comparing these conclusions with the results of our applied method (as show in table 6) shows that our predictions 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 process 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, modelers can more easily represent and communicate the workings of a process through the process models, achieving higher levels of modeling quality that better the work of an organization.

Through our case study, we have seen the difficulty beginner modelers have in creating process models that follow modeling guidelines. The majority of process models in our collection have at least one aspect that is inefficient for comprehension, which can be seen in table 7. It is obvious that beginners need help in finding these inefficiencies and that can only be done if more support is given from the industry.

It is because of this that we have created and presented our method. We have shown that it is possible to semi automatically verify process modeling guidelines with the assistance of the BPMN ontology. Doing so can quickly analyze process models and recommend solutions to increase the pragmatic quality of entire process model collections.

One limitation we must acknowledge is related with our use of the BPMN ontology to check the syntax of the process model. While we could verify the static proprieties of a process model, the BPMN ontology is not suited to specify a process model's dynamic behavior (Rospocher et al., 2014). Therefore, we could not verify, in our method, a process model's entire syntactic quality, because we could not guarantee that its *soundness*

While our method is by no means complete, as there are more guidelines in the literature beyond those used here, we hope that it brings attention to the necessity of proper tool support for creating process models with high comprehensibility. We also believe that it can provide a basis for future works in this area.

.

REFERENCES

Allweyer, T. and Allweyer, D. (2010). *BPMN 2.0 : introduction to the standard for business process modeling*. Books on Demand GmbH.

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

Cardoso, J. (2006). Process control-flow complexity metric: An empirical validation. *Services Computing, 2006. SCC'06. IEEE International ...*.

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

Gassen, J. B., Mendling, J., Thom, L. H., and de Oliveira, J. P. M. (2014). Business process modeling. In *Proceedings of the 32nd ACM International Conference on The Design of Communication CD-ROM - SIGDOC '14*, pages 1–10, New York, New York, USA. ACM Press.

Gruhn, V. and Laue, R. (2006). Complexity metrics for business process models. *9th international conference on business information ...*.

Krogstie, J. (2012). *Model-Based Development and Evolution of Information Systems*. Springer London, London.

Leopold, H., Mendling, J., and Gunther, O. (2016). Learning from Quality Issues of BPMN Models from Industry. *IEEE Software*, 33(4):26–33.

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

Mendling, J. (2008b). *Metrics for Process Models*, volume 6 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg.

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.

Mendling, J., Sánchez-González, L., García, F., and La Rosa, M. (2012). Thresholds for error probability measures of business process models. *Journal of Systems and Software*, 85(5):1188–1197.

Mendling, J., Verbeek, H., van Dongen, B., van der Aalst, W., and Neumann, G. (2008). Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering*, 64(1):312–329.

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.

Recker, J. (2011). *Evaluations of process modeling grammars : ontological, qualitative and quantitative analyses using the example of BPMN*. Springer.

Reijers, H. A., Mendling, J., and Recker, J. (2015). Business Process Quality Management. In *Handbook on Business Process Management 1*, pages 167–185. Springer Berlin Heidelberg, Berlin, Heidelberg.

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.

Sánchez-González, L., García, F., Mendling, J., Ruiz, F., and Piattini, M. (2008). *Prediction of Business Process Model Quality Based on Structural Metrics*.

Schuette, R. and Rotthowe, T. (1998). The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models. pages 240–254. Springer Berlin Heidelberg.

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.

van der Aalst, W. M. P. (2013). Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, 2013:1–37.

Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H. A., and van der Aalst, W. (2007). Quality metrics for business process models. *BPM and Workflow ...*, pages 1–12.

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*.