Automatic Standard Compliance Assessment of BPMN 2.0 Process Models

Matthias Geiger[1], Philipp Neugebauer[2], and Andreas Vorndran[1]

[1] Distributed Systems Group, University of Bamberg, Germany
`matthias.geiger@uni-bamberg.de`
[2] innoQ Deutschland GmbH, Kreuzstraße 16, 80331 München, Germany

Abstract. The *Business Process Model and Notation 2.0* is nowadays the de-facto standard for process and workflow modeling. It is supported by many modeling tools and engines which are able to consume and execute processes modeled in the native BPMN 2.0 syntax. Despite its popularity, there are still issues and drawbacks regarding standard compliance of BPMN 2.0 process models. This paper elaborates on reasons for such compliance problems and describes how those compliance issues can be revealed by performing automated checks. Finally, it is shown that standard compliance is still an issue by analyzing a set of process models.

Keywords: BPMN 2.0, standard compliance, static analysis

1 Introduction

Since its publication by the Object Management Group (OMG) in 2011, the *Business Process Model and Notation* (BPMN) 2.0 [15] is the de-facto standard for process and workflow modeling, both accepted in academia and practice. This is also reflected by its adoption as an ISO/IEC standard in 2013 [10]. BPMN 2.0 is used in a wide range of scenarios ranging from simple visualization purposes on a classic whiteboard, over documentation and simulation to actual execution of modeled processes using dedicated BPMN engines. Depending on the scenario, there is an increasing need for *valid*, standard compliant process models and model serializations, as these are a prerequisite for sensible results, especially for process simulation and execution [6, 13].

The specification [10] defines which visual shapes can be used for modeling, which constraints have to be respected and finally, since 2.0, how a model should be serialized in a standardized XML-based format. However, due to the extent and the complexity of the specification it is hard to create actually standard compliant process models without tool support. The same problem arises if the standard compliance of given process models should be evaluated by asserting the absence of rule violations. The compliance checks are hampered by the vast amount of constraints to respect as well as by the fact that these constraints are not clearly presented in a dedicated list, but are scattered all over the standard document. The specification also contains several editorial flaws and semantic inconsistencies [2, 7]. Unfortunately, the OMG does neither provide any reference

implementations, nor a tool to check the conformance of actual process models. They at least provide normative XSDs, but, as shown in [7], a schema validation is not sufficient to check all constraints, as most constraints are not expressed by the XSD files. Thus, an important prerequisite for actual standard compliance checks is an extensive and clear documentation of all relevant constraints defined in the specification. Such a documentation has been published in [4]. Based on this previous work, we present an approach to automatically assess the standard compliance of BPMN process models using the tool *BPMNspector* in this paper.

The remainder of the paper is structured as follows: First, we briefly discuss related approaches dealing with the analysis of standards compliance or conformance in Section 2. Section 3 demonstrates how standard compliance of process models can be assessed by automated checks using the tool *BPMNspector*. The tool is used to evaluate a set of publicly available process models in Section 4, which gives empirical insights into common problems showing that standard compliance still cannot be assumed in practice. The paper is concluded with a summary and a brief outlook on further research directions in Section 5.

2   Related Work

Various related approaches deal with the standard compliance or conformance of process languages and their implementations in modeling tools and engines.

The OMG itself founded the BPMN *Model Interchange Working Group* (MIWG)[3] to foster the interoperability of modeling tools [1, 12]. However, the approach and focus of the BPMN MIWG is different to ours: They analyze tools based on a set of reference models and check whether the tools are capable of loading, storing and recreating the models in a BPMN compliant serialization [12]. Thus, their work improves process models only indirectly as they target the improvement of modeling tools. Moreover, an important aspect is left out by the BPMN MIWG: They only check whether tools are able to create and deal with *valid* models. It is not investigated whether the tools are able to detect *invalid* models and whether the tools prevent the creation of invalid processes. However, the reference models created by the BPMN MIWG are used in Sect. 4 for the evaluation of our approach.

Apart from the static standard compliance properties of process models, several works concentrate on behavioral correctness of BPMN processes (e.g. [3, 8, 16]), and BPMN process engines [5, 6]. For example, the absence of deadlocks as well as lack of synchronization in BPMN process models is investigated in [16] using an approach based on Petri nets.

A broader view on the quality of BPMN models is investigated in a recent empirical study in which more than 500 models from industry are analyzed: In [13] not only syntactical rules and advanced structural aspects such as deadlock freedom, but also layout and labeling guidelines have been analyzed, which refers to the *pragmatic aspects* of model quality. In total, the authors claim to have

---

[3] The project website is `http://www.omgwiki.org/bpmn-miwg/doku.php`

checked "35 well-known BPMN guidelines and correctness rules" [13] extracted from BPMN textbooks. Their results regarding structural aspects indicate that 99% of the models are syntactically correct. However, in 22% of all models a deadlock has been found and 42% contain a lack of synchronization (i.e., an unintended multiple execution of process parts). On the downside, only a small subset of all correctness requirements are actually checked, as there exist far more than the checked 35 guidelines and rules [7]. Moreover, the syntactical correctness of BPMN process models can not be taken for granted, as all analyzed models have been created with the same modeling tool, thus not giving a representative insight in the whole state of the art in the BPMN modeling tool market.

Closely related to this work, but dedicated to WS-BPEL [14], is an approach to automatically check WS-BPEL's static analysis rules [9]. We are reusing their proposed language independent API as the basis for the implementation of the tool presented in the next section.

3   Automatic Assessment of Standards Compliance

As shown in [7], the compliance assessment of BPMN process models must be automated to be feasible for the more than 600 necessary constraint checks. An XML schema validation of the process model in the normative BPMN 2.0 serialization format [10] is a good start for such an automatic assessment. However, previous studies have proven that an XSD validation alone is not sufficient as it only detects structural violations, but referencing issues and the more sophisticated constraints are not covered [7]. Thus, we have developed the Java tool *BPMNspector*[4] to check the referential integrity and also most of the more sophisticated constraints listed in [4].
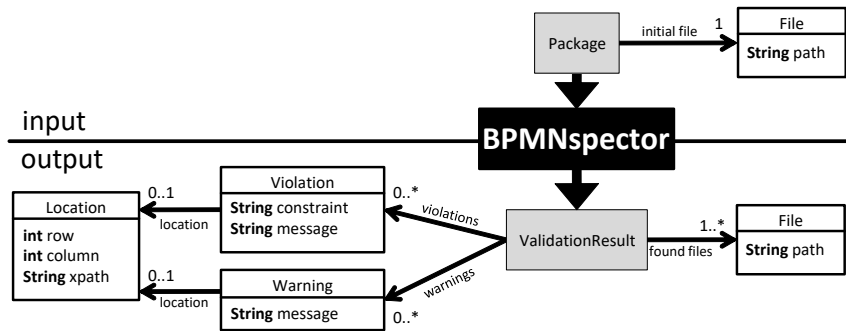

Fig. 1. BPMNspector API Class Diagram (adapted from [9])

BPMNspector enables checks of single BPMN 2.0 process model files as well as whole process model collections - either as a standalone tool or integrated into other tools. The API provided by *BPMNspector* (see Fig. 1) is process language

---

[4] More information can be found on the project web site `http://BPMNspector.org`, the source code is available at GitHub: `https://github.com/uniba-dsg/BPMNspector`

independent and has already been used by a tool checking the static analysis rules for WS-BPEL [14] process models [9].

The basic validation workflow is visualized in Fig. 2: After parsing the command line options, the BPMN file, and potentially needed further artifacts (as part of the input package, see Fig. 1) are transformed into an internal `BPMNProcess` representation. During this step the input files are also checked for schema validity. Next, the referential integrity is checked, i.e., do all referenced elements exist and do they have an allowed type. In the final step, the so called *EXT* constraints [4] are evaluated. The *EXT* constraints describe more complex constraints which are not expressed in the normative XSDs. They are defined in a declaritve way using Schematron [11], which is an ISO standard dedicated to define and check constraints for XML-based documents. If a violation of a constraint is found, it is added to the *ValidationResult* which is the output of the tool (see Fig. 1) and basis for the reports created after the execution of all validation steps.
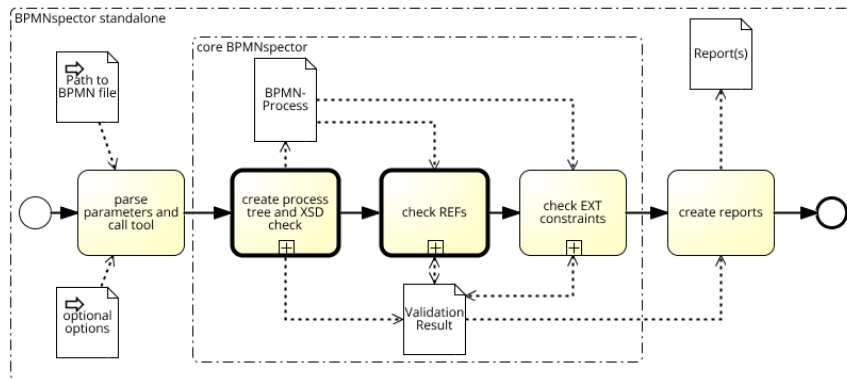

Fig. 2. Basic Workflow of BPMNspector (single file validation)

The reports, either in XML or HTML format, clearly state whether the checked file is valid. For invalid processes each warning, or found constraint violation, is described by a textual explanation and a pointer to the error location.

4   Empirical Insights and Discussion

The tool *BPMNspector* has been used to evaluate publicly available models from different process collections. The first source is the BPMN-by-example document provided by the OMG[5]. These models have been created by BPMN experts as part of standard development to demonstrate the capabilities of the BPMN specification. Although they are officially marked as "non-normative", these models are widely used and referred to both in academia and practice. Thus, they should be valid to avoid confusion of users and tool developers. Another set

---

[5] see `http://www.omg.org/spec/BPMN/20100602/2010-06-03/`

Table 1. Result Overview: Evaluated Process Collections

description	models	valid	invalid
BPMN-by-example	22	16	6
BPMN MIWG reference models	12	8	4
Collection of QuDiMo Project	32	0	32
totals	**66**	**24**	**42**

of process has been created by BPMN MIWG[6] to assess the standard compliance of modeling tools. As these models are the basis for the standard compliance evaluation, they itself must be standard compliant for sensible results. More complex models are included by a process collection of the research project QuDiMo[7].

In total, 66 process models are taken into account, ranging from a very small model containing only 9 process elements, to a complex model containing 364 process elements in an XML file with 2361 LoC. Table 1 provides an overview of the used collections and the number of valid and invalid processes.

The results regarding standard compliance are rather astonishing and a contradiction to the findings of Leopold et al. in [13]: None of the process models provided by QuDiMo is actually completely standard compliant. 27 of the checked 32 models are not even schema valid, mostly because required attributes or sub elements are missing. Furthermore, also the models created by the experts of the BPMN MIWG and the standardization task force contain errors: Whereas only one analyzed model is schema invalid, about 30% of the BPMN-by-example and BPMN MIWG models contain at least one violation of a standard constraint. For example, various BPMN MIWG models do not have all mandatory attributes which "MUST" be present according to the specification, if a `Process` is declared as `executable`.

Most detected constraint violations are related to the wrong usage of `Sequence-Flows`. Some SequenceFlows are completely missing which results in unconnected process elements. Another SequenceFlow related issue, especially occuring in the QuDiMo models, is the connection of two elements from different participants with a SequenceFlow. Instead, these elements must be connected by a Message-Flow [10, Chap. 9.3, p. 111]. Other frequent issues are non-allowed/missing event definitions or missing declarations for executable processes, such as mandatory `ItemDefinitions`. All in all, the 66 processes contain nearly 1000 found XSD schema or other constraint violations. And only 24 of 66, i.e., roughly 37%, are completely standard compliant according to the results of our compliance check.

There are various potential reasons for these findings: First of all it would be possible that the tool *BPMNspector* is incorrectly implemented and thus reports issues that are actually non-existent. To minimize this risk we adopted several measures: The implementation is based on [4] which lists the constraints

---

[6] available at `https://github.com/bpmn-miwg/bpmn-miwg-test-suite`

[7] The project website `http://pi.informatik.uni-siegen.de/qudimo` provides more information and allows for the download of the process collection.

to be respected by valid BPMN process models. All those constraints are directly extracted from the specification and for each constraint a reference to the relevant passage in the specification is given. The source code of *BPMNspector* is thoroughly tested with more than 450 process models we created for testing purposes. Moreover, the list of constraints and the whole source code is open to public scrutiny and we have reviewed both artifacts perpetually within our group to ensure the correctness of our results.

Another potential reason for the compliance issues in the analyzed models might be bad tool support. This is especially relevant for the models published by the QuDiMo project as these models are automatically transformed from another format to the normative BPMN serialization. It is also interesting that the BPMN MIWG references models are accepted as a correct reference by all participants in the working group. In fact, using some tools of participating vendors reveals that they do not complain about any issues. The same applies to the BPMN-by-example models: For example, state-of-the-art modeling tools open the "Noble Prize Process" without reporting any errors and also a manual triggering of the syntax checks does not reveal any problems. In the same file *BPMNspector* reports six violations of two different constraints regarding the invalid usage of non-visible, data related aspects. The inability of detecting constraint violations reveals the already mentioned problem of BPMN MIWG as they solely concentrate on the ability to create valid models, without analyzing their error detection capabilities.

Last but not least, the findings in this paper might indicate that the standard itself should be improved and adjusted. Whereas there are some problems regarding the editorial and also semantical quality of the standard text [2, 7], there also exist clearly and strictly defined constraints which are just ignored by most standard implementers. Especially, the aspects regarding data modeling and execution through Web Services calls do not really reflect the current state of implementation in modeling tools and engines [6].

5   Conclusion and Future Work

In this paper, we have discussed reasons for standard compliance problems and have shown how they can be detected by automated checks using the tool *BPMNspector*. *BPMNspector* does not only perform a schema validation, but is capable of checking most other relevant constraints. Using the tool to test a set of 66 processes revealed that only 24 processes are completely BPMN 2.0 compliant. Thus, we have proven that even basic "syntactic correctness" of BPMN 2.0 process models is not guaranteed, as 40% of all models failed to be schema valid, which clearly contradicts previous results [13].

In future work we try to incorporate existing work for behavioral correctness into *BPMNspector*, as such aspects are even harder to detect in process models [13]. Especially the work of Prinz et al. [16] looks promising to reveal deadlocks and lack of synchronization. Furthermore, we plan to integrate the *BPMNspector* tool into existing BPMN modeling software and BPMN execution engines to increase their capability of detecting invalid process models, which are currently rather

weak int this respect [6,7]. This will help the users and developers to create valid and standard compliant BPMN process models, thereby mitigating one of the omissions of BPMN MIWG. Moreover, we will analyze more processes from open repositories and process collections to gather further insights in the actual usage of BPMN 2.0 and its frequent problems. This should not only indicate common problems but also provide interesting insights for clarifications, improvements and modifications in further revisions of the BPMN specification.

References

1. F. Bonnet, G. Decker, L. Dugan, M. Kurz, Z. Misiak, and S. Ringuette. Making BPMN a True lingua franca. online on BPMtrends.com, 2014.
2. E. Börger. Approaches to Modeling Business Processes. A Critical Analysis of BPMN, Workflow Patterns and YAWL. *Software & Systems Modeling*, 11(3):305–318, 2012.
3. M. Chinosi and A. Trombetta. Modeling and Validating BPMN Diagrams. In *IEEE Conf. on Commerce and Enterprise Computing (CEC), Vienna, Austria*, pages 353–360. IEEE, 2009.
4. M. Geiger. BPMN 2.0 Process Model Serialization Constraints. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik, no. 92, Otto-Friedrich Universität Bamberg, 2013.
5. M. Geiger, S. Harrer, J. Lenhard, M. Casar, A. Vorndran, and G. Wirtz. BPMN Conformance in Open Source Engines. In *9th IEEE SOSE*, pages 21–30, San Francisco Bay, CA, USA, 2015.
6. M. Geiger, S. Harrer, J. Lenhard, and G. Wirtz. BPMN 2.0: The state of support and implementation. *Future Generation Computer Systems*, 2017. (accepted for publication; available online: http://dx.doi.org/10.1016/j.future.2017.01.006).
7. M. Geiger and G. Wirtz. BPMN 2.0 Serialization - Standard Compliance Issues and Evaluation of Modeling Tools. In *5th Intl. WS on Enterprise Modelling and Information Systems Architectures*, pages 177–190, St. Gallen, Switzerland, 2013.
8. P. V. Gorp and R. Dijkman. A Visual Token-based Formalization of BPMN 2.0 Based on In-place Transformations. *Information and Software Technology*, 55(2):365–394, 2013.
9. S. Harrer, M. Geiger, C. R. Preißinger, D. Bimamisa, S. J. Schuberth, and G. Wirtz. Improving the Static Analysis Conformance of BPEL Engines with BPELlint. In *9th IEEE SOSE*, San Francisco Bay, CA, USA, 2015.
10. ISO/IEC. *ISO/IEC 19510:2013 – Information technology - Object Management Group Business Process Model and Notation*, 2013. v2.0.2.
11. ISO/IEC. *ISO/IEC 19757-3:2016 – Information technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation – Schematron*, 2016.
12. M. Kurz. BPMN model interchange: The quest for interoperability. In *8th Intl. Conf. on Subject-oriented Business Process Management*, pages 1–10. ACM, 2016.
13. H. Leopold, J. Mendling, and O. Günther. Learning from Quality Issues of BPMN Models from Industry. *IEEE Software*, 33(4), 2016.
14. OASIS. *Web Services Business Process Execution Language*, 2007. v2.0.
15. OMG (Object Management Group). *Business Process Model and Notation (BPMN) Version 2.0*, 2011. v2.0.
16. T. M. Prinz, N. Spieß, and W. Amme. A First Step towards a Compiler for Business Processes. In *Compiler Construction - 23rd Intl. Conf., Grenoble, France, April 5-13, 2014. Proceedings*, volume 8409 of *LNCS*, pages 238–243. Springer, 2014.