# Detecting Interoperability and Correctness Issues in BPMN 2.0 Process Models

Matthias Geiger and Guido Wirtz

Distributed Systems Group, University of Bamberg, Germany
{matthias.geiger,guido.wirtz}@uni-bamberg.de

**Abstract.** Although BPMN 2.0 is an international standard widely used in practice, interoperability of process models is still an issue. Even between tools and engines claiming to be BPMN compliant the model exchange is often complicated or impossible as the tools produce incorrect model representations or do not support the standardized BPMN serialization format. In this position paper we present reasons for interoperability issues and show why defining a set of constraints derived from the standard is crucial to fix an important subset of those issues. We are currently developing a tool which can check this set of rules automatically.

**Keywords:** BPMN 2.0, Interoperability, Correctness, XML Serialization, Standard compliance

## 1 Motivation

Since its official release in January 2011 the Business Process Model and Notation (BPMN) [8] is used more and more in academia and by practitioners alike. The variety of BPMN models spreads from simple workflow descriptions consisting of only a few sequential tasks for illustration purposes to complex models including data modeling and calls to existing software systems in order to be executed on BPMN compliant process engines. For the former process models, which are often drawn by hand or using tools like Microsoft PowerPoint, interoperability and correctness are not of major interest. But correct and interoperable models are essential when a process definition should be deployed on a BPMN engine like Activiti[1] or when models are to be exchanged between different tools. Model exchange and refinement is often performed in interdisciplinary teams in which usage of the same used modeling tools cannot be assumed. Therefore the need for a standardized BPMN serialization format to enable model interchange which also ensures "correct" process models is widely accepted[2].

In fact, BPMN version 2.0 [8] introduces such a standardized serialization format based on a XML Schema Definition (XSD). Unfortunately, this serialization format is not used or correctly/fully implemented by most tool vendors and therefore real interoperability is still far from given.

[1] http://www.activiti.org

[2] see for example the "BPMN-I" initiative of Bruce Silver (http://www.brsilver.com/2011/04/05/a-profile-for-bpmn-interoperability/)

## 2 Reasons for Missing Interoperability

The reasons for incorrect models and therefore interoperability issues are manifold but can be divided into two main groups:

- **Vendor Policy:** The usage of a proprietary serialization format, missing import and (especially) export functionality is often intentionally used by vendors. Either the ability to switch to competitive products is limited (vendor lock-in) or a simpler format is used to comply to the internal meta model. This is especially the case when a tool is not initially designed for BPMN models but an existing process and workflow modeling tool is enhanced to deal also with BPMN models.
  For instance, the BizAgi Process Modeler[3] is not able to handle BPMN models saved in the format proposed by BPMN [8]. The Signavio Process Editor[4] supports importing and exporting BPMN compliant XML-files but heavily uses extension elements which hinders the usage of Signavio models in other tools.
- **Implementation Problems:** Although vendors might be willing to provide interoperable model definitions, actual interoperability is not in place, because the constraints raised by the standard are overlooked, misinterpreted or faultily implemented so that incorrect BPMN models might be exported. Since the official release of BPMN 2.0, various ambiguities, inconsistencies and faults have been revealed[5] which further inhibit the successful usage of BPMN [1].

## 3 Creating a Standard-based Rule Set for XML Serialization and Checking the Extracted Rules

As vendor policies may not be affected easily, we focus on giving support for implementation issues. In order to observe all rules it is needless to say that an overview of all constraints stated in [8] is essential. Unfortunately, BPMN falls short in providing such an overview.

Sources for constraints in the standard document are the running text, tables, class diagrams and XSD excerpts. The extraction of rules from the latter sources is rather straightforward (e.g., mandatory attributes, cardinalities, value constraints). In contrast to this, constraints in the running text are harder to identify and frequently some interpretation of the text is needed. We worked through the standard document and in a first iteration we derived more than 300 rules.

Moreover, in about fifty cases inconsistencies between the text and/or the class diagrams and the XSD have been revealed. Frequently some attributes of elements are defined as mandatory in the text and the class diagrams, but the

[3] `http://www.bizagi.com/modeler/`
[4] `http://www.signavio.com`
[5] e.g., see the issue tracking list `http://www.omg.org/issues/bpmn2-rtf.open.html`

schema definition marks the same attribute as optional. To give an example: Even the definition for the BPMN root element `definitions` is affected by this problem: Table 8.1 in [8, p. 53] states that the attribute `name` is mandatory. In contrast to that, this attribute is completely missing in the XSD excerpt in Table 8.3 [8, p. 54] and is defined as optional in the normative XSD schema[6].

In order to improve and check the completeness and correct interpretation of our rule set, we are currently cross-checking it with other less extensive collections of BPMN constraints[7] and with the consistence checks integrated in various tools[8].

Based on a consolidated list of rules for BPMN models, it is possible to check whether a serialized process model is consistent to this rule set (and therefore to the standard itself).

Parts of the extracted rules can already be checked by performing a XML schema validation. An example for such rules are value limitations, as for the attribute `gatewayDirection` of a BPMN *Gateway*. Here only the values 'Unspecified', 'Converging', 'Diverging' or 'Mixed' are allowed. This restriction is realized as a XSD *SimpleType* restricting *Strings* (see [8], p.91).

However the overwhelming majority of constraints cannot be covered by schema validation. Specific examples can also be found in the context of BPMN *Gateways*: Gateways have incoming *SequenceFlows* which are realized as `incoming` sub elements that refer to a *SequenceFlow* definition using a `xs:QName` reference. With XML schema validation, a reference to an arbitrary or even non-existent BPMN element would be regarded as correct. Moreover, depending on the value of the attribute `gatewayDirection` the number of incoming and outgoing sequence flows has to be limited in a different manner (see [8], p.91) which is also not checkable by schema validation.

To tackle these issues, we are currently developing a tool to check all extracted rules which are not covered by schema validation yet.

## 4 Related Work

Academic research mainly concentrates on semantic validation, verification and correctness checks for BPMN models [3, 5, 9], assuming that the BPMN models used already comply to the standard. In contrast to this, we concentrate on issues regarding the serialized form of process models. Hence, in this paper the term correctness refers to compliance to the constraints postulated by the standard [8] and not to semantic correctness.

Closer related to our work is [2] which proposes a meta model and a serialization format for the prior BPMN Version 1.1 [7] including some checks regarding reference existence and leveraging XPath for more sophisticated validations. Due to the major revisions in BPMN 2.0, most parts of this approach are outdated

[6] see `http://www.omg.org/spec/BPMN/20100501/BPMN20.xsd`

[7] e.g., see Bruce Silver: *BPMN Method & Style, 2nd edition*, 2011, p. 135-139

[8] e.g., the itp-commerce modeling tool checks a series of constraints (see: `http://help.itp-commerce.com/index.php?id=81&L=0`)

by now, as BPMN 2.0 provides a standardized XML interchange format. Nevertheless, the proposed usage of XML Technologies like schema validation which is now used in the current version of the standard is able to check some basic rules as stated above.

A good example for the importance and practical benefits of a list of relevant constraints for a process language standard is the Web Services Business Process Execution Language (BPEL) [6]. BPEL provides a list of 95 static analysis rules which cannot be checked using a simple XML schema validation, but should be checked by BPEL engines during the deployment process. Using such a list tailored to BPMN, it is much easier to generate test cases and verify standard conformance such as presented for BPEL in the tool betsy [4].

## 5 Conclusion and Outlook

The main contribution of our work will be an extensive set of constraints stated in the standard and a tool checking these constraints. These contributions provide a basis for the successful practical usage of BPMN. The rule set and the tool may be used by modeling tool vendors and engine developers to check if their software generates, respectively is able to import and deploy standard compliant documents. It might be used during import stages in order to reject non-compliant models or to benchmark different tools/engines regarding their standard compliance.

As a side effect of our work we are able to report several issues to the BPMN 2.1 Revision Task Force to improve the upcoming version of BPMN.

## References


1. E. Börger. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software & Systems Modeling*, 11:305–318, 2012.
2. M. Chinosi and A. Trombetta. Modeling and Validating BPMN Diagrams. In B. Hofreiter and H. Werthner, editors, *CEC*, pages 353–360. IEEE Computer Society, 2009.
3. R. M. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, 50(12):1281 – 1294, 2008.
4. S. Harrer, J. Lenhard, and G. Wirtz. BPEL Conformance in Open Source Engines. In *Proceedings of the 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA'12), Taipei, Taiwan*. IEEE, December 17-19 2012.
5. N. Lohmann, E. Verbeek, and R. M. Dijkman. Petri net transformations for business processes - a survey. *T. Petri Nets and Other Models of Concurrency*, 2:46–63, 2009.
6. OASIS. *Web Services Business Process Execution Language*, April 2007. v2.0.
7. OMG. *Business Process Modeling Notation, v1.1*, January 2008.
8. OMG. *Business Process Model and Notation (BPMN) Version 2.0*, January 2011.
9. P. Wong and J. Gibbons. A Process Semantics for BPMN. In *Formal Methods and Software Engineering*, LNCS, pages 355–374. Springer Berlin Heidelberg, 2008.