What we can learn from Quality Issues of BPMN Models from Industry

Henrik Leopold[1], Jan Mendling[2], and Oliver Günther[3]

[1] Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081HV Amsterdam, The Netherlands
[2] Wirtschaftsuniversität Wien, Welthandelsplatz 1, 1020 Vienna, Austria
[3] Universität Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany

Abstract

Many organizations use business process models for documenting business operations and for formalizing business requirements in software engineering projects. In recent years, the Business Process Model and Notation (BPMN), a specification by the Object Management Group (OMG), evolved into the leading standard for process modeling. A challenge of BPMN is its complexity: the specification offers a huge variety of different elements and often several representational choices for the same semantics. This raises the question of how well modelers can deal with these choices. Empirical insights into BPMN usage from the perspective of practitioners are still missing. We close this gap by analyzing a large set of BPMN 2.0 process models from practice. We found that particularly representational choices for splits and joins, the correct use of message flow, the proper decomposition of models, and the consistent labeling appear to be connected with quality issues and give five specific recommendations how these issues can be avoided in the future.

1 Introduction

Business process models play an important role for documenting business operations and for formalizing business requirements in software engineering. The Business Process Model and Notation (BPMN) is a specification by the Object Management Group (OMG) [1] and the de-facto standard for process modeling. A major challenge of BPMN in practice concerns its complexity resulting from the considerable number of elements it offers, its sophisticated semantics, and its representational choices.

While the uptake of BPMN in practice has triggered the research community to study its usage, empirical studies on how BPMN is actually used are scarce [2, 3, 4]. The scope of most contributions is restricted to language properties, e.g. [5, 6], instead of actual usage. On the other hand, it is widely acknowledged that process modeling is difficult and needs expertise and guidance [7].

With this paper, we aim to shed light on the actual usage of BPMN, and conducted a study with six companies from industry. This way, we wanted to understand if quality issues arise and how they can be prevented. The participating companies provided us access to a total of 585 BPMN process models. We implemented an automatic guideline checker that covers rules described in BPMN textbooks [8, 9, 10]. The results helped us to learn about the frequency of different classes of modeling problems and to suggest a set of measures to overcome them.

2 Correctness and Quality of BPMN Process Models

Figure 1 shows the BPMN model of a recruiting process. We use this example to illustrate typical problems we found in our industry sample. At first glance, the model seems reasonably well defined. A start event is used to depict that the receipt of an application triggers the process. All activities are connected from start to end. However, a closer look reveals problems with the structure of the process as well as with its layout and labels.

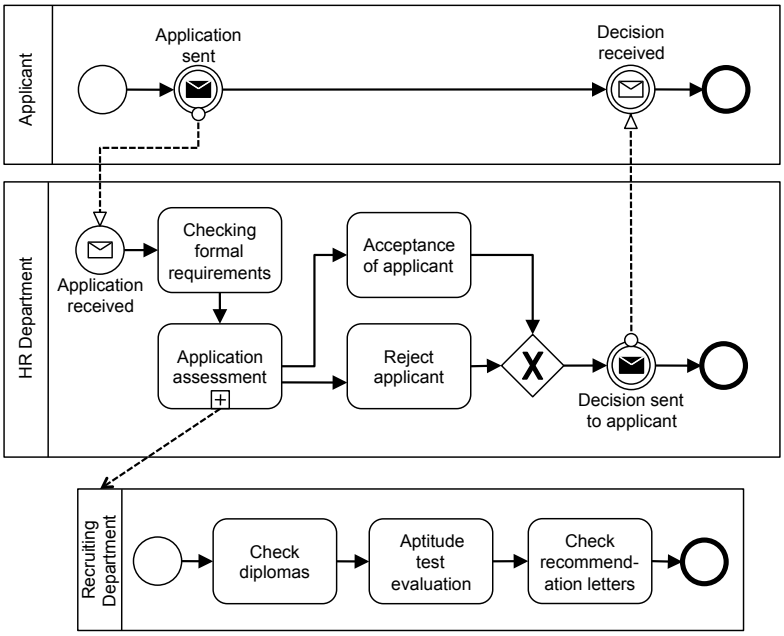

Figure 1: BPMN process model with typical errors made in practice

Structural problems relate to the way how elements of the model are connected. The application process displays some inconsistent split and join behavior. The two outgoing arcs of the activity "Application assessment" actually define a parallel split according to the BPMN specification. This parallel split is inappropriate here since an applicant cannot be rejected and accepted at the same time. Indeed, BPMN offers two options to represent splits and joins: (a) with gateways, and (b) with multiple incoming and outgoing arcs. The modeler must be aware that two outgoing arcs represent parallel split behavior, while two incoming arcs capture an unsynchronized merge behavior, which appears to be difficult for many modelers.

Another problem relates to the association between main process and sub process. In the example, this association is inconsistent, as the role descriptions of the main and the sub process do not match. While the main process is performed by the "HR Department", the steps of the sub process are executed by the "Recruiting Department". Since the sub process is semantically part of the main process, such a mismatch represents an inconsistency.

Furthermore, the model could be improved in terms of its layout. In fact, there is no reason why the control flow direction varies for the activities "Checking formal requirements" and "Application assessment". While some users apply this practice in order to safe space, it is likely to confuse readers due to an increased cognitive effort for recognizing the order of tasks [11, 12]. In our sample, we found several cases of what we call "banana models" - they run from left to right up to the end of the modeling canvas and then make a backward turn to run from right to

left in the space underneath.

Another class of problems relates to the inconsistent usage of natural language. The activity "Reject applicant" instructs to "reject" an applicant by using an imperative verb in the beginning. The activity "Application assessment", by contrast, contains the noun "assessment" at the end in order to instruct the reader to "assess" it. Such inconsistencies in the grammar of the labels have been found to be confusing for model readers [13]. To avoid inconsistencies and ambiguities, guidelines recommend the use of verb-object structures for activities (e.g. "Reject applicant"), object-participle structures for events (e.g. "Applicant rejected"), and object-participle questions for gateways (e.g. "Applicant rejected?") [8, 9, 7].

3 Prior Studies on BPMN

Several researchers have investigated BPMN and its role in industry. Recker et al. examined BPMN using a so-called representational analysis and, among others, identified issues related to the lane and pool constructs [14, 5]. Recker also investigated actual BPMN usage by conducting a world-wide survey with 590 BPMN users. His findings provide insights into the modeling tools practitioners employ, the tool functionalities they make us of, and the problems and desires practitioners have with respect to the notation [4]. Detailed findings about the usefulness of BPMN in industry were presented by zur Muehlen and Ho [3]. They report on their experiences of applying BPMN in the context of redesigning the service management process in a truck dealership.

While all these studies provide valuable insights, they do not discuss the quality issues of BPMN models in practice. To the best of our knowledge, the only study based on BPMN models from industry was conducted by zu Muehlen and Recker in 2008 [2]. They analyzed 120 BPMN process models and reported on the frequency of use of the different BPMN constructs. That study, however, did not discuss quality issues of models. To close this gap, we conduct an analysis based on a large set of BPMN 2.0 process models from industry.

4 A Study on BPMN Use in Industry

For our study, we collected a total of 585 BPMN 2.0 process model from six companies. The companies came from different industries and varied in size as well as in their degree of modeling experience. What all companies had in common is that they had a focus on process model quality as they voluntarily participated in the study and that they received basic training on how to create proper models. What is more, all of them used the Signavio process modeling editor[1] for creating and maintaining their models, i.e., all models were created under the same technical circumstances. Using a variety of automated checking techniques, we developed a tool for checking a set of 35 well-known BPMN guidelines and correctness rules. This set covers in particular the guidelines proposed by Silver [9] and Allweyer [8] as well as the recommendations by White and Miers [10].

Figure 2 gives an overview of the 15 most frequent quality issues[2]. As indicated by the different bar colors, the quality issues can be subdivided into the three categories *structure*, *layout*, and *labeling*. In the following, we have a closer look at each of these categories.

[1] www.signavio.com

[2] Note that there are no overlaps among the quality issues.

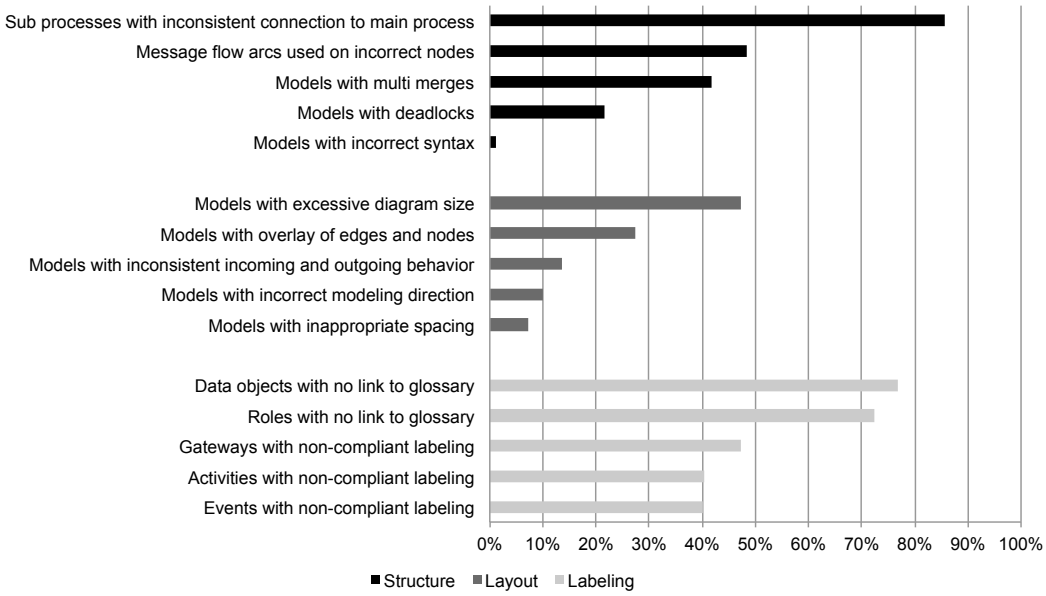

Figure 2: Violation of modeling guidelines in practice

4.1 Structure

The structure category refers to the consistent and correct use of modeling elements such as activities, gateways, events, pools, and flow connectors. Table 1 provides an overview of the encountered structure inconsistencies.

Table 1: Structure inconsistencies

Inconsistency	**Explanation**
Inconsistent connection between sub and main process	Pools of the main process need to have the same role as the pool from the sub process.
Usage of message flows on incorrect nodes	Message flows should only be attached to the designated and semantically correct elements.
Existence of multi merges	Multi merges lead to multiple executions of the subsequent flow.
Existence of deadlocks	Deadlocks block the continuation of the process and arise if gateways are combined in an erroneous fashion.
Incorrect syntax	Elements must be appropriately linked according to the BPMN syntax rules.

Our study yielded mixed results. On the positive side, we found that about 99% of the investigated models are free from any syntactical errors. Apparently, the general notion of syntactical correctness is well understood and does not represent a problem in our sample. On the negative side, we observe that several more specific concepts are not well respected. 22% of the models contain deadlocks and 42% contain multi merges. Both errors typically result from the use of implicit splits and joins and dramatically affect the understanding as well as the automated processing of the models. More issues stem from the use of message flows and the

inconsistent association of main and sub processes. As for message flow, modelers often connected it to wrong elements. In particular, the throwing message event appears to cause confusion. Modelers often attached an incoming message flow to a throwing (i.e., sending) message event. It seems that the active notion of a throwing message event is not well understood. Altogether, we encountered issues for 48% of all message flow arcs. The biggest problem, however, is caused by the inconsistent association of main and sub processes. In 86% of all models containing sub processes, the roles of the sub process do not match the corresponding role of the main process. This shows that many modelers from the investigated organizations are either not aware or do not sufficiently take care of the links between process models. They still focus on single models instead of considering their model as part of a company-wide process architecture.

4.2 Layout

The layout category is concerned with the proper positioning of the process model elements in terms of cognitive effectiveness. Hence, it is the goal of the rules and guidelines of this category to guarantee that a model can be easily read and understood. Table 2 provides an overview of the investigated inconsistencies.

Table 2: Layout inconsistencies

Inconsistency	**Explanation**
Excessive diagram size	Diagram must fit on an A3 page (297mm x 420mm).
Overlay of edges and nodes	Edges and nodes are not allowed to overlap.
Inconsistent incoming and outgoing behavior	Control flow as well as message flow arcs should be consistently used. For example, control flow arcs should not be attached to the bottom of activities.
Incorrect modeling direction	The modeling direction "left to right" should be consistently applied.
Inappropriate spacing	The distance between connected elements should be at least 50% of the element size.

Our study shows that only a few models suffer from issues such as inappropriate spacing, arcs flowing into the wrong direction, or inconsistent incoming and outgoing behavior. Apparently, many modelers from the investigated organizations have understood the importance of these basic layout requirements and apply them appropriately. Nevertheless, not all aspects are respected and implemented to the same degree. 27% of all investigated models contain edge or node overlays. Such overlays often originate from the intention to save space. Still, since overlays affect the understanding of a model, they should be avoided. The biggest layout issue, however, concerns models of extensive size. About 47% of all models exceed the maximum diagram size, i.e., they do not fit on a DIN A3 page. This shows that the modelers do not sufficiently consider using sub processes or decomposing a model into several parts in order to reduce the complexity of a model. Instead, they try to capture all the details in a single model.

4.3 Labeling

As pointed out earlier, labeling refers to the proper use of natural language in the process model. Table 3 provides an overview of the investigated labeling inconsistencies.

Looking at the results, we see two major issues: structure and reuse. While the non-compliant labeling items refer to the structure, i.e., the usage of potentially ambiguous syntactic patterns, the items concerned with the use of glossaries refer to proper reuse. Our empirical results

Table 3: Labeling inconsistencies

Inconsistency	**Explanation**
Data objects with no link to glossary	Data object description is not linked to glossary.
Roles with no link to glossary	Role description is not linked to glossary.
Gateways with non-compliant labeling	Violation against the object-participle-question convention ("Contract signed?").
Activities with non-compliant labeling	Violation against the verb-object convention ("Sign contract").
Events with non-compliant labeling	Violation against the object-participle convention ("Contract signed").

show that between 40% and 47% of the labels follow syntactic patterns that are potentially ambiguous and hence may negatively affect the understanding of the model [13]. Apparently, linguistic aspects and their impact are not clear to all modelers. As a result, they use varying and sometimes even ambiguous structures for labeling process model elements [15]. The problem is further complicated by the fact that a guideline asking for labeling an activity with an *imperative verb* and an *object* is simply not sufficiently intuitive to every modeler. Moreover, glossaries are used by a fraction of modelers only. About 72% of all roles and 77% of all data objects are not linked to a glossary. Although this may not be intuitive to every modeler, the importance of glossaries can hardly be overestimated. As modeling initiatives typically include several modelers, not using glossaries often leads to the inconsistent usage of roles and data objects. As a result, the integrity of the entire process architecture is at risk.

5 Five Measures for Improved BPMN Modeling

The results show that many quality and correctness criteria are well respected in the investigated organizations. In particular, our data illustrates that syntactical correctness and compliance with basic layout rules, such as appropriate spacing, do not represent a problem. On the other hand, we observed that many advanced structural concepts, such as consistency among process models, process model size as well as the labeling of process model elements, appear to be connected with quality issues. Apparently, the available modeling recommendations and guidelines are not sufficiently clear. Although all companies participating in the study received a respective training and were aware of the modeling guidelines proposed in well-known textbooks [8, 9, 10], they still incorporated a considerable number of mistakes and violations. In the following, we discuss the five major problem areas we identified and give specific recommendations on how to avoid them. Figure 3 illustrates our recommendations graphically.

1. *Avoid implicit splits and joins*: Implicit splits and joins via multiple outgoing and incoming arcs are the major cause for deadlocks and multi merges. This problem is caused by BPMN offering several options to represent such semantics. While users can easily remember the meaning of the different gateway symbols, they often do not realize that multiple outgoing arcs represent an AND-split and multiple incoming arcs represent an XOR-join. As a result, they erroneously model an XOR-join although they intended to model an AND-join. From a theoretical perspective, the possibility to represent the same semantics in multiple graphical ways is referred to as *concept excess* and has been found to negatively affect understandability [5]. We therefore recommend to prohibit the use of multiple arcs. The semantics of splits and joins can be clearly and unambiguously defined using gateways.

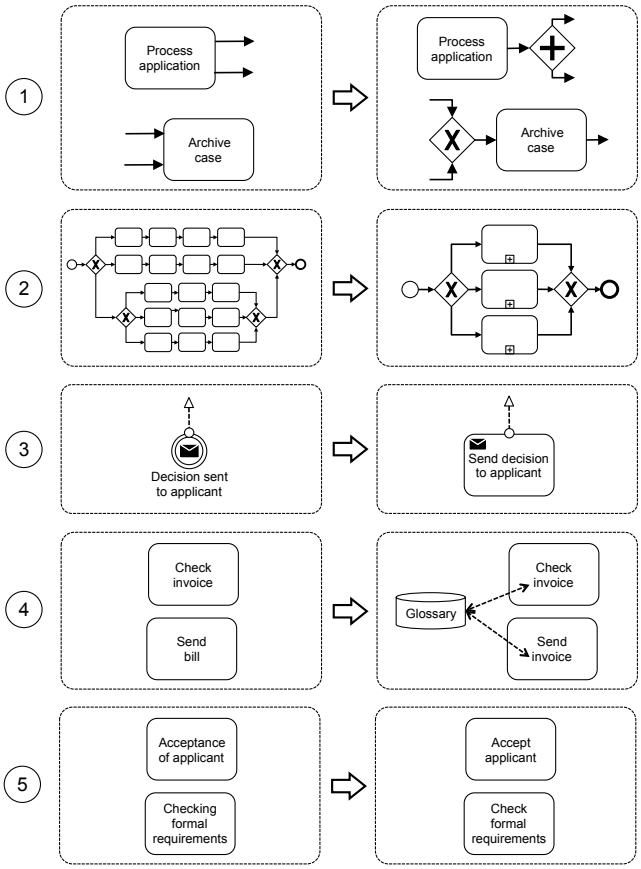

Figure 3: Graphical illustration of five improvement measures

2. *Provide tool support for proper model decomposition*: Our empirical results show that modelers may struggle with the proper decomposition of their models. Either the models are too big, or they are not fully consistent. Since both problems can be effectively enforced by a modeling tool, we recommend to implement respective mechanisms. Users could be notified if their models become too big or how to consistently split up their models. This way, inconsistencies such as the usage of different roles in sub processes can be effectively avoided.

3. *Omit the throwing message event*: Our study suggests that message flow arcs may cause several problems. Modelers in our study struggled with correctly applying it and often connected it to incorrect elements. Particularly the throwing message event appeared to cause confusion. As events are generally perceived as passive components of a process, it seems that the active notion of a throwing message event is often misunderstood. We hence recommend to remove the throwing message event from the symbol set. It is easy to use activities for throwing events instead. In such a way, the notion of an event is more consistent and many cases of wrong message flow use can be avoided.

4. *Establish centrally maintained glossary*: The consistent reuse of central concepts such as roles and data objects is an important requirement for a sound process architecture. However, the consistent use and maintenance of a glossary is unlikely to be autonomously

implemented by all modelers that are involved in a modeling endeavor. The fact that the tool used for creating the investigated models offered a glossary functionality and the observation that this functionality was hardly used illustrates this point vividly. A possible solution could include a centrally defined glossary that either automatically monitors and imports new terms or is regularly updated by a dedicated glossary manager. The resulting glossary containing all roles and data objects would greatly contribute to the consistency of the process architecture.

5. *Provide tool support for applying linguistic checks during the modeling process*: Achieving consistency with respect to the structural use of natural language seems to be difficult. Guidelines such as the compliance with the verb-object structure that requires modelers to use an "imperative verb" followed by an "object" appear to not be intuitive to many modelers. As a result, they fail to implement them. The most effective measure seems to communicate such inconsistencies already during the modeling process. Modeling tools could use techniques such as refactoring to automatically suggest a correct version of a non-compliant label [15]. This way, inconsistent natural language formulations could be prevented right from the start.

A closer look at our recommendation list reveals that particularly recommendations 1 and 3 can be traced back to the representational choices of BPMN. The modelers from the investigated organizations struggled with correctly dealing with these choices and incorporated errors that should be avoided. Our recommendations have the advantage that they do not restrict the expressive power of BPMN. Instead, they help the modeler to select a preferable representation when a specific pattern of behavior needs to be expressed. Recommendations 2, 4, and 5 refer to quality issues that may also occur in other process notations such as Event-driven Process Chains or UML activity diagrams. Still, our study demonstrates that also BPMN models may suffer from these problems. Hence, also these recommendations contribute to a consistent process architecture.

6 Conclusion

In this paper, we empirically investigated the quality issues of 585 BPMN 2.0 process models from industry. Our study shows that many fundamental quality aspects of BPMN seem to be well understood. However, it also demonstrates that particularly the representational choices of BPMN may provoke errors. While our study does not allow to argue about the general quality of BPMN process models in industry, it provides valuable insights into which aspects may require particular attention.

To avoid the observed quality issues, we recommended five specific measures: (1) the avoidance of implicit splits and joins, (2) tool support for proper decomposition, (3) the omission of the throwing message event, (4) the maintenance of a centrally defined glossary, and (5) tool support for checking labels during the modeling process. All these recommendations do not restrict the expressive power of BPMN, but help the modeler to select a preferable representation.

Finally, it should be noted that our recommendation list is not exhaustive. However, according to the models we analyzed, the compliance with these recommendations has the potential to avoid over 90% of the problems we observed. We are convinced that this represents an important step forward towards a consistent, unambiguous, and understandable process architecture.

References

[1] OMG. Business Process Model and Notation (BPMN), Version 2.0, 2011. URL `http://www.omg.org/spec/BPMN/2.0`.

[2] Michael zur Muehlen and Jan Recker. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In *Proceedings of CAiSE*, pages 465–479. Springer-Verlag, 2008.

[3] Michael zur Muehlen and Danny T Ho. Service process innovation: a case study of bpmn in practice. In *Hawaii international conference on system sciences, proceedings of the 41st annual*, pages 372–372. IEEE, 2008.

[4] Jan Recker. Opportunities and constraints: the current struggle with bpmn. *Business Proc. Manag. Journal*, 16(1):181–201, 2010.

[5] Jan Recker, Michael Rosemann, Peter F. Green, and Marta Indulska. Do ontological deficiencies in modeling grammars matter? *MIS Quarterly*, 35(1):57–79, 2011.

[6] Kathrin Figl, Jan Mendling, and Mark Strembeck. The influence of notational deficiencies on process model comprehension. *J. AIS*, 14(6), 2013.

[7] Jan Mendling, Hajo A. Reijers, and Wil M. P. van der Aalst. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology*, 52(2):127–136, 2010.

[8] Thomas Allweyer. *BPMN 2.0 - Business Process Model and Notation*. Books on De-mand GMBH, Norderstedt, 2nd edition, 2009.

[9] Bruce Silver. *BPMN Method and Style, with BPMN Implementer's Guide*. Cody-Cassidy Press, 2nd edition, January 2011. ISBN 978-0982368114.

[10] Stephen A. White and Derek Miers. *BPMN Modeling and Reference Guide: Understanding and Using BPMN*. Future Strategies Incorporated, 2008.

[11] Kathrin Figl and Mark Strembeck. On the importance of flow direction in business process models. In *9th International Conference on Software Engineering and Applications*, 2014.

[12] Philip Effinger, Nicole Jogsch, and Sandra Seiz. On a study of layout aesthetics for business process models using bpmn. In *Business Process Modeling Notation*, pages 31–45. Springer, 2011.

[13] Jan Mendling, Hajo A. Reijers, and Jan Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems*, 35(4):467–482, 2010.

[14] Jan Recker, Marta Indulska, Peter Green, et al. How good is bpmn really? insights from theory and practice. 2006.

[15] Henrik Leopold, Sergey Smirnov, and Jan Mendling. On the refactoring of activity labels in business process models. *Information Systems*, 37(5):443–459, 2012.