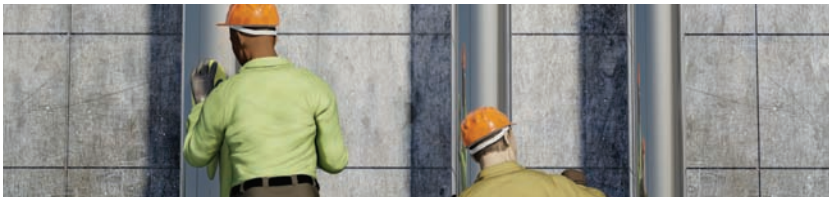
Learning from Quality Issues of BPMN Models from Industry

Henrik Leopold, Vrije Universiteit Amsterdam
Jan Mendling, Wirtschaftsuniversität Wien
Oliver Günther, Universität Potsdam

// Researchers analyzed 585 BPMN 2.0 process models from six companies. They found that split and join representations, message flow, the lack of proper model decompositivon, and labeling relate to quality issues. They formulated five recommendations to deal with these issues. //



BUSINESS PROCESS MODELS play an important role in documenting business operations and formalizing business requirements in software engineering. The de facto standard for process modeling is the Business Process Model and Notation (BPMN; www.omg.org/spec/BPMN/2.0), a specification by the Object Management Group. A major challenge of BPMN in practice is its complexity resulting from its many elements, sophisticated semantics, and representational choices.

Although BPMN's widespread adoption has motivated researchers to study its use, empirical studies of how people actually use it are scarce.[1–3] Most contributions are restricted to language properties.[4,5] On the other hand, it's widely acknowledged that process modeling is difficult and requires expertise and guidance.[6] (For more on prior BPMN research, see the sidebar.)

Here, we aim to shed light on the actual use of BPMN, through a study with six companies from industry. We wanted to understand how quality issues arise and how to prevent them. The companies gave us access to 585 BPMN process models. We implemented an automatic guideline checker that covered rules described in BPMN textbooks. The results helped us learn about the frequency of different classes of modeling problems and formulate measures to overcome them.

BPMN Process Model Correctness and Quality

The BPMN model of a recruiting process in Figure 1 illustrates typical problems we found in our industry sample. At first glance, the model seems reasonably well defined. A start event depicts 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 process's structure, layout, and labels.

Structural problems relate to how the model elements are connected. The application process displays 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 because an applicant can't be rejected and accepted at the same time.

BPMN offers two options to represent splits and joins: gateways or multiple incoming and outgoing arcs. Modelers must be aware that two outgoing arcs represent a parallel split, whereas two incoming arcs capture an unsynchronized merge. Many modelers appear to have difficulty making this distinction.

0740-7459/16/$33.00 © 2016 IEEE

Also, the association between the main process and subprocess is inconsistent because the role descriptions of the main process and subprocess don't match. Whereas the HR department performs the main process, the recruiting department executes the subprocess steps. Because the subprocess is semantically part of the main process, such a mismatch represents an inconsistency.

Furthermore, the model's layout could be improved. There's no reason why the control flow direction varies for the activities "Checking formal requirements" and "Application assessment." Although some users apply this practice to save space, it will likely confuse readers owing to the increased cognitive effort for recognizing the task order.[7,8] In our sample, we found several cases of "banana models," which run from left to right to the end of the modeling canvas and then run from right to left in the space underneath.

Another class of problems relates to inconsistent use of natural language. The activity "Reject applicant" uses an imperative verb at the beginning to indicate the action. In contrast, the activity "Application assessment" uses the noun "assessment" at the end to indicate the action. Such grammatically inconsistent labels can confuse model readers.[9] To avoid inconsistencies and ambiguities, guidelines recommend using verb–object structures for activities (for example, "Reject applicant"), object–participle structures for events (for example, "Applicant rejected"), and object–participle questions for gateways (for example, "Applicant rejected?").[6,10,11]

BPMN Use at the Six Companies

The six companies came from different industries and varied in size and modeling experience. However, they were all concerned about process model quality and had received basic training on creating proper models. Also, they all used the Signavio process-modeling editor (www.signavio.com) to create and maintain their models. That is, all models were created under the same technical circumstances.

Using various automated techniques, we developed a tool for checking a set of 35 well-known BPMN guidelines and correctness rules. In particular, this set covered the guidelines proposed by Bruce Silver[10] and Thomas Allweyer[11] as well as the recommendations by Stephen White and Derek Miers.[12]

PRIOR BPMN STUDIES

Jan Recker and his colleagues examined BPMN using a representational analysis and identified issues related to pools and lanes.[1,2] (A pool delineates a business process; a lane subdivides a pool.) They also investigated actual BPMN use by surveying 590 BPMN users worldwide. Their findings provide insights into the modeling tools those practitioners employed, the tool functionalities they used, and their problems and desires regarding BPMN.[3] Michael zur Muehlen and Danny Ho presented detailed findings about BPMN's usefulness in industry.[4] They reported their experiences applying BPMN to redesign a truck dealership's service management process.

Although these studies provided valuable insights, they didn't discuss the quality issues of BPMN models in practice. To the best of our knowledge, zur Muehlen and Recker conducted the only study of actual BPMN models from industry.[5] They analyzed 120 BPMN process models and reported on the frequency of use of the different BPMN constructs. However, they didn't discuss quality issues, either.

To close this gap, we analyzed 585 BPMN 2.0 process models from six companies to investigate quality issues (see the main article).

References

1. J. Recker et al. "How Good Is BPMN Really? Insights from Theory and Practice," *Proc. 2006 European Conf. Information Systems* (ECIS 06), 2006; http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1075&context=ecis2006.
2. J. Recker et al., "Do Ontological Deficiencies in Modeling Grammars Matter?," *MIS Q.*, vol. 35, no. 1, 2011, pp. 57–79.
3. J. Recker, "Opportunities and Constraints: The Current Struggle with BPMN," *Business Process Management J.*, vol. 16, no. 1, 2010, pp. 181–201.
4. M. zur Muehlen and D.T. Ho, "Service Process Innovation: A Case Study of BPMN in Practice," *Proc. 41st Ann. Hawaii Int'l Conf. System Sciences* (HICSS 08), p. 372.
5. M. zur Muehlen and J. Recker, "How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation," *Advanced Information Systems Eng.*, LNCS 5074, Springer, 2008, pp. 465–479; http://web.stevens.edu/ses/documents/fileadmin/documents/pdf/MIZU-JARE-BPMN-CAiSE-2008.pdf

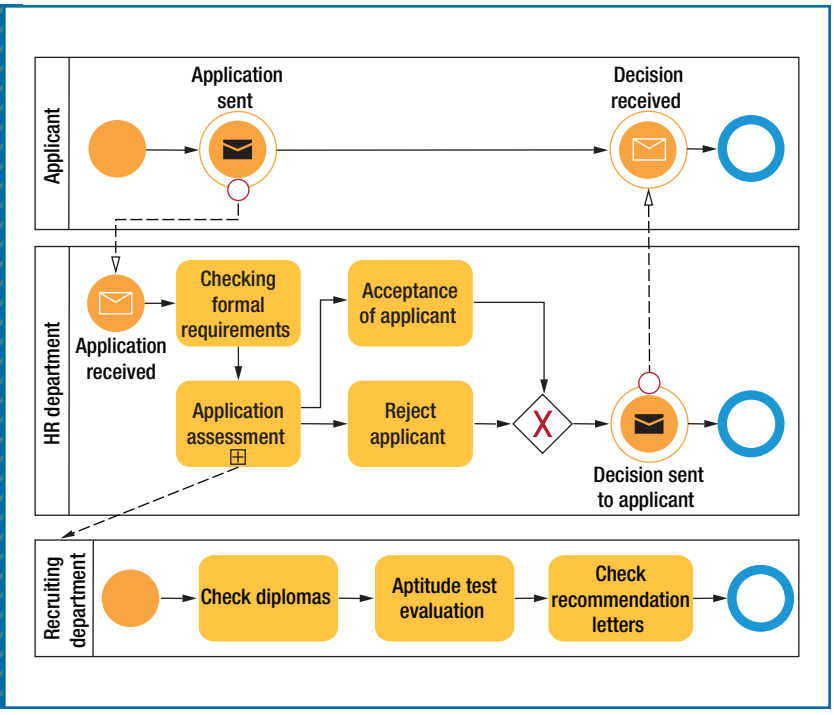


FIGURE 1. A BPMN (Business Process Model and Notation) model of a recruiting process, showing typical errors made in practice. The errors involve the model's structure, layout, and labels.

Figure 2 gives an overview of the 15 most frequent quality issues (no overlap existed among them). They fall into three categories: structure, layout, and labeling.

Structure

This category refers to the consistent and correct use of modeling elements such as activities, gateways, events, pools (which delineate business processes), and flow connectors. Table 1 lists the encountered structural inconsistencies.

On the positive side, approximately 99 percent of the investigated models had no syntactical errors. Apparently, the general notion of syntactical correctness was well understood and didn't represent a problem.

On the negative side, several more specific concepts weren't well respected; 22 percent of the models contained deadlocks, and 42 percent contained multimerges. Both errors typically result from using implicit splits and joins and dramatically affect the understanding and automated processing of models.

More issues were related to message flows. Modelers often connected the message flow to the wrong elements. In particular, the throwing (sending) message event appeared to cause confusion. Modelers often attached an incoming message flow to a throwing message event. They didn't necessarily seem to clearly understand the active notion of a throwing message event. We encountered issues for 48 percent of all message flow arcs.

The biggest problem, however, was due to the inconsistent association of main processes and subprocesses. In 86 percent of models with subprocesses, the subprocess's roles didn't match the main process's corresponding role. This showed that many modelers were unaware of or didn't sufficiently take care of the links between process models. They still focused on single models instead of considering their model as part of a companywide process architecture.

Layout

This category deals with positioning the process model elements to ensure cognitive effectiveness. So, this category's rules and guidelines aim to guarantee that a model can be easily read and understood. Table 2 lists the inconsistencies we found.

Only a few models suffered from issues such as inappropriate spacing, arcs flowing in the wrong direction, or inconsistent incoming and outgoing behavior. Apparently, many of the modelers understood the importance of these basic layout requirements and applied them appropriately.

Nevertheless, not all aspects were respected and implemented to the same degree; 27 percent of all models contained overlapping edges or nodes. Such overlap often originates from the intention to save space. Still, because overlap affects the understanding of a model, it should be avoided.

The biggest layout issue concerned excessive diagram size. Approximately 47 percent of the models exceeded the maximum diagram size; that is, they didn't fit on an A3 page. This shows that the modelers didn't sufficiently consider using subprocesses or decomposing a model into several parts to reduce its complexity. Instead, they tried to capture all the details in a single model.

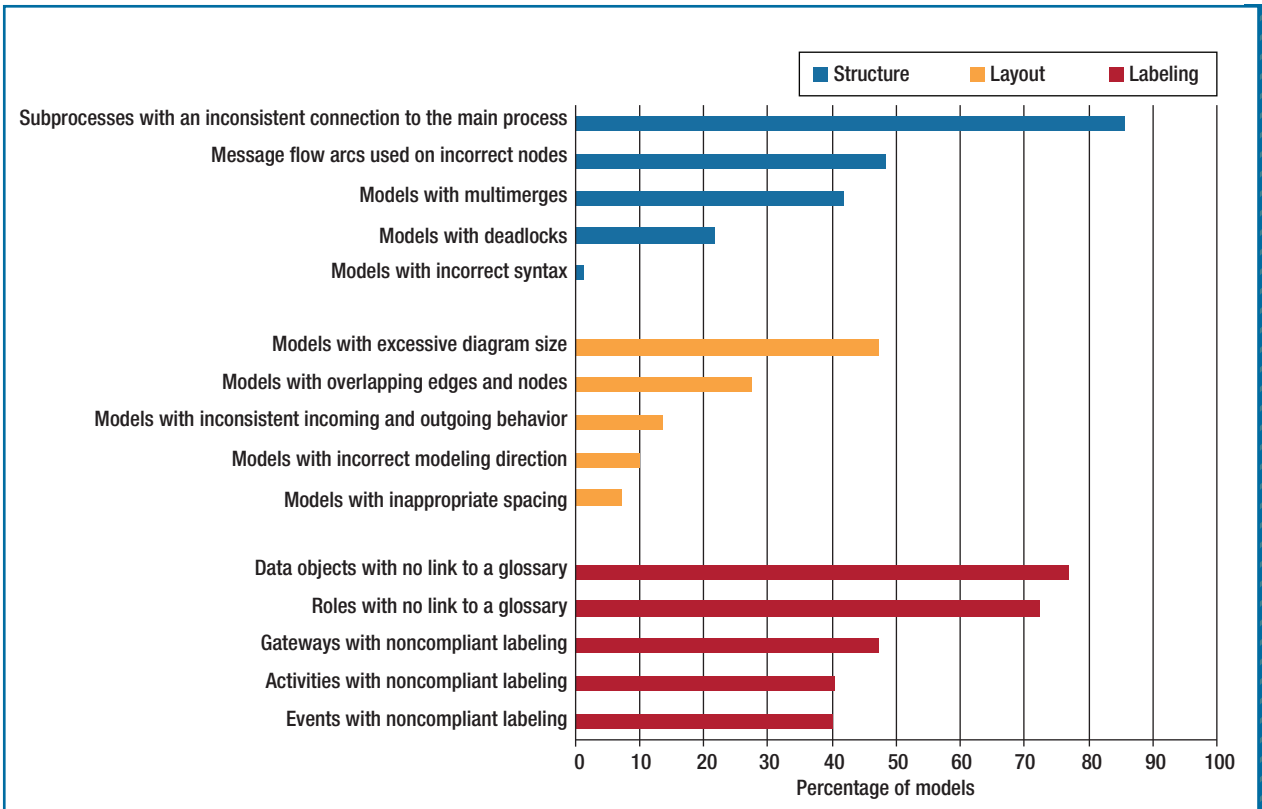


FIGURE 2. Violations of modeling guidelines in practice. Although the companies were trained in modeling and aware of modeling guidelines, they still committed a number of violations.

TABLE 1

Structural inconsistencies.

Inconsistency	Explanation
An inconsistent connection between a subprocess and the main process	The main process's pools must have the same role as the subprocess's pool.
Using message flows on incorrect nodes	Message flows should be attached only to the designated and semantically correct elements.
Multimerges	Multimerges lead to multiple executions of the subsequent flow.
Deadlocks	Deadlocks block the continuation of the process; they arise if gateways are erroneously combined.
Incorrect syntax	Elements must be appropriately linked according to the Business Process Model and Notation syntax rules.

Labeling

As we mentioned earlier, labeling refers to the proper use of natural language. Table 3 lists the inconsistencies.

A major issue was structure. About 42 percent of the labels followed potentially ambiguous syntactic patterns that might have

TABLE 2

Layout inconsistencies.

Inconsistency	Explanation
Excessive diagram size	The diagram must fit on an A3 page (297 × 420 mm).
Overlapping edges and nodes	Edges and nodes must not overlap.
Inconsistent incoming and outgoing behavior	Control flow and message flow arcs should be consistent. For example, control flow arcs shouldn't be attached to the bottom of activities.
Incorrect modeling direction	Left-to-right modeling should be consistently applied.
Inappropriate spacing	The distance between connected elements should be at least 50 percent of the element size.

TABLE 3

Labeling inconsistencies.

Inconsistency	Explanation
Data objects with no link to a glossary	The data object description isn't linked to a glossary.
Roles with no link to a glossary	The role description isn't linked to a glossary.
Gateways with noncompliant labeling	A violation of the object–participle question convention occurred.
Activities with noncompliant labeling	A violation of the verb–object convention occurred.
Events with noncompliant labeling	A violation of the object–participle convention occurred.

negatively affected understanding of the model.[9] Apparently, linguistic aspects and their impact weren't clear to all modelers. So, they used varying and sometimes even ambiguous structures for labeling process model elements.[13] The problem was further complicated by the fact that the guideline of labeling an activity with an imperative verb and an object wasn't sufficiently intuitive to every modeler.

Another major issue was reuse, which is related to glossary use. Only a fraction of the modelers used glossaries. Approximately 72 percent of all roles and 77 percent of all data objects weren't linked to a glossary. Although the need to link to glossaries might not have been intuitive to every modeler, glossaries' importance can hardly be overestimated. Because modeling initiatives typically include several modelers, not using glossaries often leads to inconsistent use of roles and data objects. So, the integrity of the entire process architecture is at risk.

Five Ways to Improve BPMN Modeling

We identified five major problem areas, for which we developed the following recommendations.

First, avoid implicit splits and joins (see Figure 3a). Implicit splits and joins through multiple outgoing and incoming arcs are the major cause of deadlocks and multimerges. This problem occurs because BPMN offers several options to represent such semantics. Although users can easily remember the gateway symbols' meanings, they often don't realize that multiple outgoing arcs represent an AND-split and multiple incoming arcs represent an XOR-join. So, they erroneously model an XOR-join when they intended to model an AND-join. Representing the same semantics in multiple graphical ways is called *concept excess* and has been found to negatively affect understandability.[4] So, we recommend prohibiting the use of multiple arcs. Modelers can use gateways to clearly and unambiguously define split and join semantics.

Second, provide tool support for proper model decomposition (see Figure 3b). Our empirical results show that modelers might struggle with model decomposition. The models might be either too big or not fully consistent. Modeling tools with an appropriate mechanism could

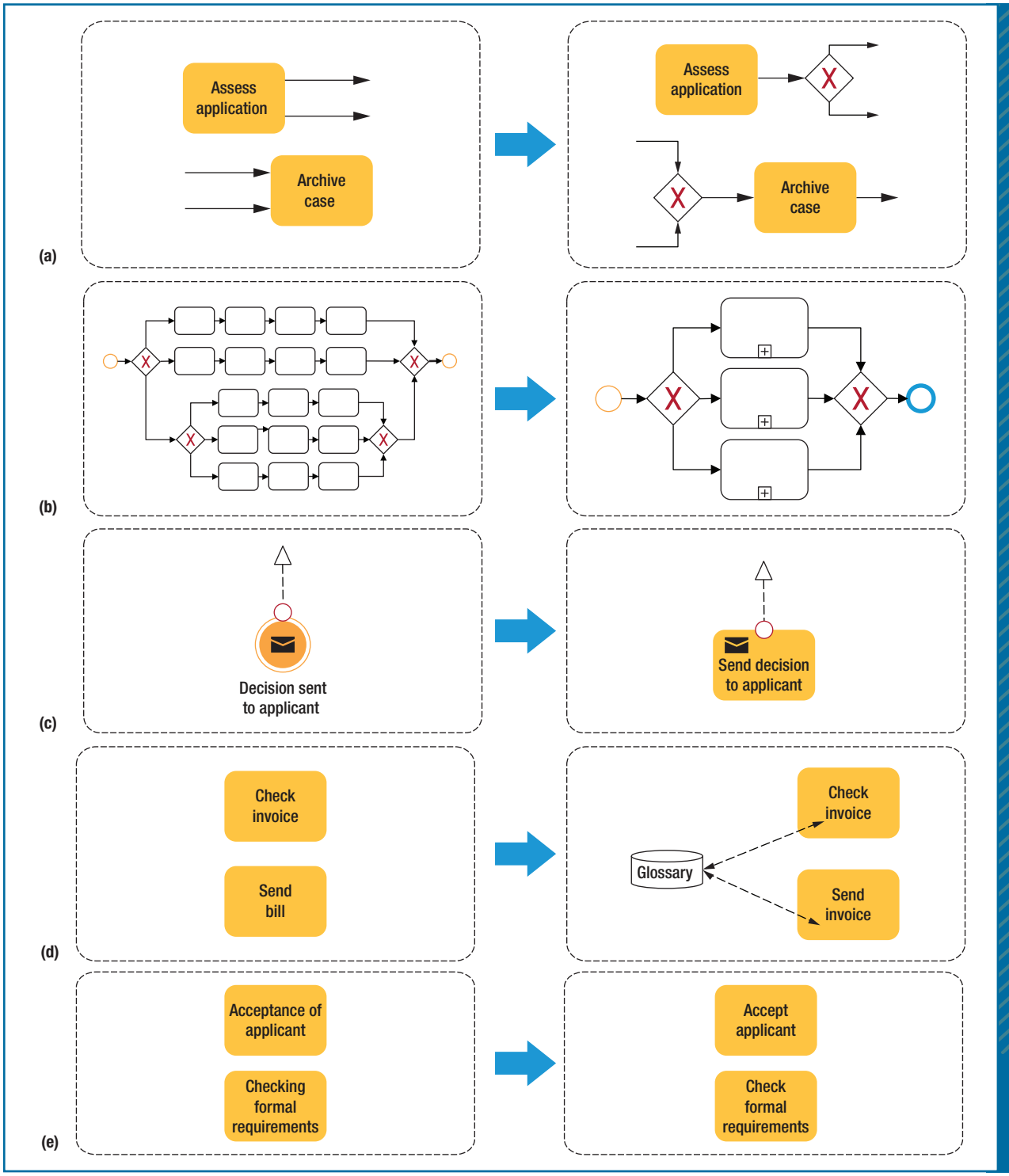


FIGURE 3. Five ways to improve BPMN modeling. (a) Avoid implicit splits and joins. (b) Provide tool support for proper model decomposition. (c) Omit the throwing message event. (d) Establish a centrally maintained glossary. (e) Provide tool support for linguistic checks during modeling.

ABOUT THE AUTHORS

HENRIK LEOPOLD is an assistant professor in Vrije Universiteit Amsterdam's Department of Computer Science. His research interests include business process modeling, natural-language-processing techniques, process model matching, and process architectures. Leopold received a doctorate in information systems from Humboldt-Universität zu Berlin. His doctoral thesis received the German Targion Award 2014 for the best dissertation in the area of strategic information management. Contact him at h.leopold@vu.nl.

JAN MENDLING is a full professor and the head of the Institute for Information Business at Wirtschaftsuniversität Wien. His research areas include business process management, conceptual modeling, and enterprise systems. Mendling received a PhD (Dr. rer. soc. Oec) from Wirtschaftsuniversität Wien. He's a founder of the Berlin Business Process Management Community of Practice (www.bpmb.de) and a board member of the Austrian Gesellschaft für Prozessmanagement. His PhD thesis won the Austrian Computer Society's Heinz Zemanek Award and the German Targion Award 2008 for dissertations in the area of strategic information management. Contact him at jan.mendling@wu.ac.at.

OLIVER GÜNTHER is the president of the University of Potsdam. His areas of expertise include IT strategy, IT business architectures, and security and privacy. He has been an IT strategy consultant and executive in several start-up companies, including Poptel, Germany's first voice-over-IP company. Günther received a PhD in computer science from the University of California, Berkeley. He previously was president of the German Informatics Society. Contact him at oliver.guenther@uni-potsdam.de.

prevent both problems. Such tools could notify users when their models become too big or give advice on how to consistently split their models. This way, modelers could avoid inconsistencies such as the use of different roles in subprocesses.

Third, omit the throwing message event (see Figure 3c). Our study suggests that message flow arcs might cause several problems. As we mentioned before, modelers struggled with correctly applying them and often connected them to incorrect elements. The throwing message event appeared to particularly cause confusion. Because people generally perceive events as passive components of a process, it seems they often misunderstand the active notion of a throwing message event. It's easy to use activities instead of throwing events. That way, the notion of an event is more consistent, and modelers can often avoid wrong message flow.

Fourth, establish a centrally maintained glossary (see Figure 3d). The consistent reuse of central concepts such as roles and data objects is necessary for a sound process architecture. However, not all modelers will consistently use and maintain a glossary on their own. For example, the modeling tool in our study offered a glossary functionality, but hardly anyone used it. 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 a consistent process architecture.

Finally, provide tool support for linguistic checks during modeling (see Figure 3e). Achieving consistent structural use of natural language seems difficult. As we mentioned before, guidelines such as compliance with the verb–object structure appear to be unintuitive to many modelers. So, they don't implement them. The most effective measure seems to be to communicate such inconsistencies during modeling. Modeling tools could use techniques such as refactoring to automatically suggest a correct version of a noncompliant label.[13] This could prevent inconsistent natural-language formulations right from the start.

A closer look at these recommendations reveals that particularly the first and third recommendations are connected to the representational

choices of BPMN. The modelers in the six companies struggled with correctly dealing with these choices and incorporated errors that should be avoided. Our recommendations don't restrict BPMN's expressive power. Instead, they help modelers select a preferable representation when they must express a specific behavior pattern. The second, fourth, and fifth recommendations refer to quality issues that might also occur in other process notations such as event-driven process chains or UML activity diagrams.

Although our study doesn't let us argue about the general quality of BPMN process models in industry, it provides valuable insights into which aspects might require particular attention.

Our recommendation list isn't exhaustive. However, according to the models we analyzed, compliance with these recommendations could avoid over 90 percent of the problems we observed. We're convinced that this represents an important step toward a consistent, unambiguous, and understandable process architecture.

References

1. M. zur Muehlen and J. Recker, "How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation," *Advanced Information Systems Eng.*, LNCS 5074, Springer, 2008, pp. 465–479; http://web.stevens.edu/ses/documents/fileadmin/documents/pdf/MIZU-JARE-BPMN-CAiSE-2008.pdf.
2. M. zur Muehlen and D.T. Ho, "Service Process Innovation: A Case Study of BPMN in Practice," *Proc. 41st Ann. Hawaii Int'l Conf. System Sciences* (HICSS 08), p. 372.
3. J. Recker, "Opportunities and Constraints: The Current Struggle with BPMN," *Business Process Management J.*, vol. 16, no. 1, 2010, pp. 181–201.
4. J. Recker et al., "Do Ontological Deficiencies in Modeling Grammars Matter?," *MIS Q.*, vol. 35, no. 1, 2011, pp. 57–79.
5. K. Figl, J. Mendling, and M. Strembeck, "The Influence of Notational Deficiencies on Process Model Comprehension," *J. Assoc. for Information Systems*, vol. 14, no. 6, 2013, pp. 312–338.
6. J. Mendling, H.A. Reijers, and W.M.P. van der Aalst, "Seven Process Modeling Guidelines (7PMG)," *Information and Software Technology*, vol. 52, no. 2, 2010, pp. 127–136.
7. K. Figl and M. Strembeck, "On the Importance of Flow Direction in Business Process Models," *Proc. 9th Int'l Conf. Software Eng. and Applications* (ICSOFT-EA 14), 2014, pp. 132–136; http://wi.wu.ac.at/home/mark/publications/icsoft-ea14.pdf.
8. P. Effinger, N. Jogsch, and S. Seiz, "On a Study of Layout Aesthetics for Business Process Models Using BPMN," *Business Process Modeling Notation*, J. Mendling, M. Weidlich, and M. Weske, eds., Springer, 2011, pp. 31–45.
9. J. Mendling, H.A. Reijers, and J. Recker, "Activity Labeling in Process Modeling: Empirical Insights and Recommendations," *Information Systems*, vol. 35, no. 4, 2010, pp. 467–482.
10. B. Silver, *BPMN Method and Style, with BPMN Implementer's Guide*, 2nd ed., Cody-Cassidy Press, 2011.
11. T. Allweyer, *BPMN 2.0—Business Process Model and Notation*, 2nd ed., Books on Demand GmbH, 2009.
12. S.A. White and D. Miers, *BPMN Modeling and Reference Guide: Understanding and Using BPMN*, Future Strategies, 2008.
13. H. Leopold, S. Smirnov, and J. Mendling, "On the Refactoring of Activity Labels in Business Process Models," *Information Systems*, vol. 37, no. 5, 2012, pp. 443–459.