Analysis of Prevalent BPMN Layout Choices on GitHub

Daniel Lübke[1,2][https://orcid.org/0000-0002-1557-8804] and Daniel Wutke[3]

[1] Digital Solution Architecture, Hanover, Germany
[2] Leibniz Universität Hannover, Germany
daniel.luebke@digital-solution-architecture.com
https://www.digital-solution-architecture.com
[3] dwutke@gmail.com

Abstract. Layout of BPMN diagrams greatly influences their understandability. The primary objective of this study is to understand prevalent choices of modelers for their design of BPMN diagrams. As a research method we use repository mining to analyze BPMN diagrams we found on GitHub. We found that BPMN diagrams on GitHub are mostly laid out from left-to-right and that layout direction choices differ by the modeling tool, process model type (pool vs. no pools) and purpose (toy vs. non-toy).

Keywords: BPMN Layout · GitHub Mining · Repository Mining

1 Introduction

Layout is one of the influencing factors of understandability of BPMN diagrams [3]. While some empirical research exists on this topic, we want to explore real-world BPMN processes and analyze the use of layouts – and influencing factors of those; for example, what layout direction (left-right vs. top-bottom) is dominantly used?

While GitHub has been used in software engineering research [6], its use for BPMN-related research is only in the beginning [4, 5]. Within this paper we re-use the dataset by Heinze et al. [4] and present a preliminary analysis of layout direction choices made for the BPMN diagrams contained therein.

We present a preliminary study, which is structured as follows: First we present our research design in Sect. 2 before we explain how we mined GitHub and how we handled the obtained models in Sect. 3. Results of our statistical analysis are presented in Sect. 4 for which we give our interpretation in Sect. 5. Finally, we discuss threats to validity in Sect. 6 before we conclude and give possible future research topics.

2 Research Questions

We want to answer the following research questions related to the layout direction of BPMN diagrams found on GitHub:

J. Manner, S. Haarmann, S. Kolb, N. Herzberg, O. Kopp (Eds.): 13th ZEUS Workshop,
ZEUS 2021, Bamberg, held virtually due to Covid-19 pandemic, Germany, 25-26 February 2021,
published at http://ceur-ws.org

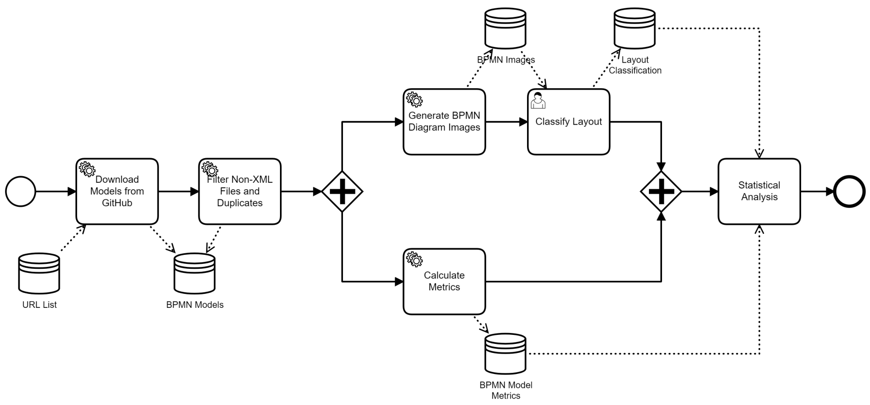


Fig. 1: The Research Process Followed

RQ1: What layout directions are used and how is their usage distributed?
Because existing research predicts better understandability for horizontal layouts [2, 3] and existing modeling guidelines mandate them [1, 9], we hypothesize that left-right diagrams are in the majority.

RQ2: What influence does the modeling tool have on layout direction?
Differences due to modeling tools have been shown for BPEL [8]. We expect that such differences also exist with BPMN.

RQ3: What influence has project ownership on layout direction?
We expect that (explicit or implicit) modeling guidelines and shared authorship within a project will lead to uniformity of layouts in any given project. Thus, we expect that the majority of diagrams within a project will have the same layout.

RQ4: Are "toy" diagrams laid out differently?
Because it has been demonstrated before that models exhibit different properties based on different purposes [7] (e.g., productive vs. example), we expect toy diagrams to be smaller and thus to have simpler layouts (horizontal & vertical only).

RQ5: Are diagrams with pools laid out differently?
Because we expect that laying out pools with more complex layouts is difficult, we expect pools to have significantly more left-right and top-bottom layouts. Because we think that pools lead to even more left-right modeling, we expect that the proportion of this layout is even larger in the diagrams with pools.

In order to answer our research questions we followed the following research process as illustrated in Fig. 1: We downloaded all files in the list of [4] (not all of which were still available online) and filtered those files according to the steps described in Sect. 3.

Later, both authors manually classified the layout of the diagrams and calculated BPMN metrics with a self-written tool called *BPMN Layout Analyzer* for various diagram metrics[4]. All classification data and metric data was stored in CSV files on which statistical analysis was performed with R. These steps are described in Sect. 4.

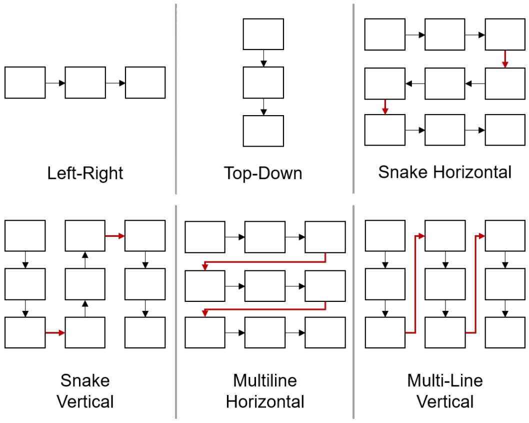


Fig. 2: Names of Layout Directions

During analysis of layout directions, we labeled BPMN diagrams as shown in Fig. 2.

3 GitHub Mining and Data Cleansing

We started by downloading the BPMN files from GitHub as they have been identified by Heinze et al. [4]. This means that we did not mine GitHub per se but downloaded all models by the list of models identified by Heinze et al. Although the original list contained 8904 unique BPMN files, only 8467 files were still available as of 2020-04-06.

For each diagram we generated a PNG file by using BPMN.io's bpmn-to-image. This failed for some files due to missing diagram interchange information or other file format compliance issues. This left us with 5299 unique processes.

In addition BPMN DI layout information or the XML itself were corrupt, which was ignored by BPMN.io so that after merging in the results from our BPMN Layout analyzer tool, only 4638 diagrams were left. In order to exclude "junk diagrams" we removed diagrams from the data set, which did not have a) at least two activities (neither counting events nor gateways), and b) were

[4] Freely available at: `https://github.com/dluebke/bpmnlayoutanalyzer/`

Table 1: Distribution of Diagram Layouts

Layout	% total	% non-toy	% toy	% no pools	% pools
left-right	79.52	73.35	86.30	78.21	86.80
multiline-horizontal	0.55	0.67	0.42	0.41	1.32
multiline-vertical	0.10	0.19	0.00	0.12	0.00
other	9.34	10.07	8.54	9.12	10.56
snake-horizontal	1.96	2.88	0.95	2.19	0.66
snake-vertical	0.20	0.29	0.11	0.24	0.00
top-down	8.33	12.56	3.69	9.71	0.66

connected enough. Too low connectedness is found in diagrams that are just used for placing all BPMN elements without any sequence flows, which was probably done to make illustrations.

Therefore, we used the following threshold for the number of subgraphs sg:

$$p = \begin{cases} 2 \times |pools^{expanded}| & : pools^{expanded} > 0 \\ 2 & : pools^{expanded} = 0 \end{cases} \tag{1}$$

$$s^e = |subprocesses^{expanded}| \tag{2}$$

$$e^e = 2 \times |eventsubprocess^{expanded}| \tag{3}$$

$$e^c = |eventsubprocess^{collapsed}| \tag{4}$$

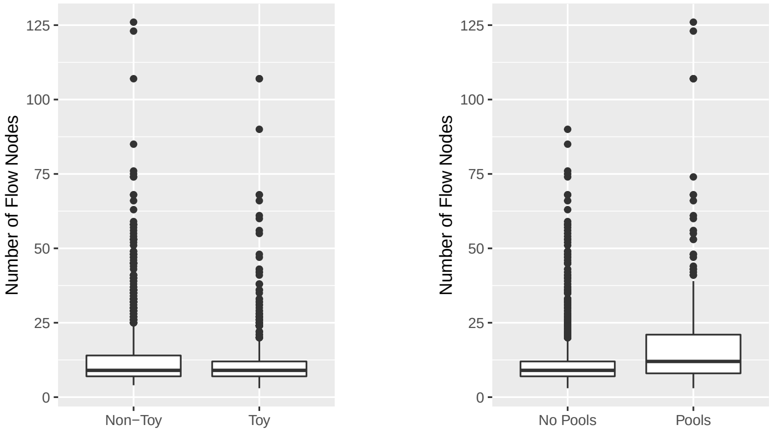
$$sg \leq p + s^e + e^e + e^c \tag{5}$$

First we define how many subgraphs our process-flow is allowed to have (equation 1): We want to exclude diagrams in which the main process falls apart into more than 2 subgraphs. For diagrams with pools we allow 2 subgraphs per expanded pool. Next, we allow an additional subgraph for an expanded subprocess (equation 2) because a new process must be contained in it. Event subprocesses are different because they are not connected to the main process flow. As such, an additional subgraph must be granted for each event subprocess, if the event subprocess is collapsed (equation 4). If the event subprocess is expanded two additional subgraphs are allowed: one for the event subprocess and one for the process contained in it (equation 3).

This left us with 1992 diagrams for analysis. The analysis of the influence of project ownership on layout directions includes duplicates and is based on a total of 7396 processes and 2745 processes after determining metrics and relevance filtering.

4 Execution & Statistics

For getting process and layout direction counts both authors manually and independently classified the diagram layout direction. After the first round, approx. 10% differences between layout direction classifications had been found which were resolved later in a shared session to reach a unified understanding.



(a) Toy vs. No-Toy (b) Pools vs. No Pools

Fig. 3: Comparison of Flow Node Count for different Diagram Subsets

The total distribution of layout directions is shown in the first data column in Table 1: The most common layout direction was left-to-right, followed by "other" layouts, which describe chaotic or too unclean layout directions, and top-down layouts. More advanced layouts like snake or multi-line layouts are rarely used.

We further classified if a BPMN model is a "toy" model or not by searching for the key words "test", "dummy", and "example" in the complete file name including path. The distribution of layout directions with regards to toy vs. non-toy processes are shown in the middle columns of Table 1. Left-right layouts are used even more frequently in toy diagrams, while top-down layouts are used more often with non-toy diagrams. We performed a simulated Fisher's Exact Test (100,000 rounds) to test whether the distributions of layout directions is significantly different between the toy and non-toy subsets.

In the following we calculated the number of pools and flow nodes in the processes with the *BPMN Layout Analyzer*. The distribution of layout directions for BPMN diagrams with or without pools are shown in the last two columns in Table 1. There are more left-right layouts used in conjunction with pools than without and more top-down layouts are used without pools than with pools. We again used a simulated Fisher's Exact Test (100,000 rounds) in order to check whether the distributions of layout directions is significantly different between diagrams with or without pools. This test again yields a highly significant p-value ($p = 9.999\,9 \times 10^{-6}$).

In a next step we analyzed the sizes of diagrams measured in number of flow nodes for toy vs. no toy diagrams (see Fig. 3a) and for diagrams with or without pools (see Fig. 3b): The mean number of flow nodes of the toy subset is 11.24 and the mean of the non-toy subset is 13.4. A Wilcoxon hypothesis test for a

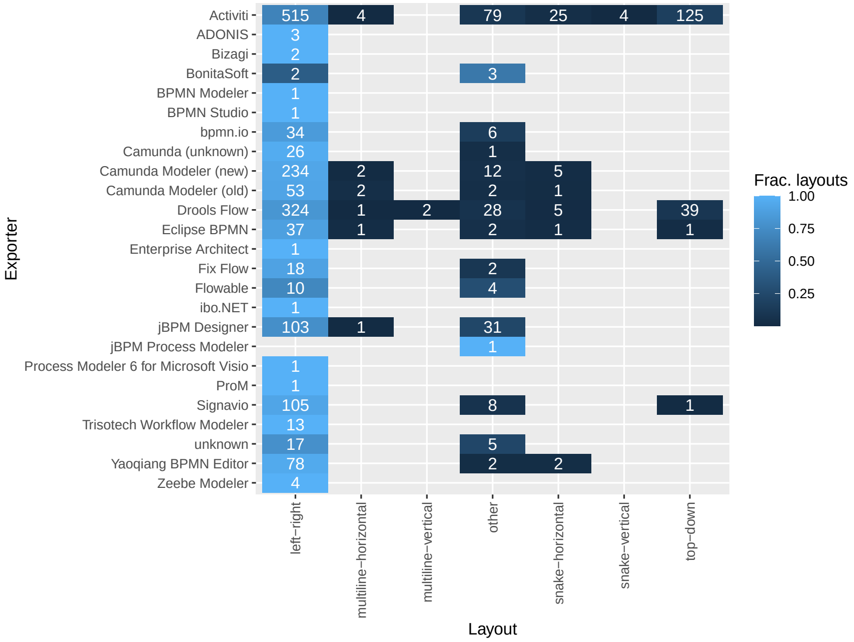


Fig. 4: Absolute and Relative Numbers of Processes by Layout and BPMN Editor

difference of means yields a p-value of 0.013 1. The mean number of flow nodes of the pools subset is 18.08 and 11.34 for diagrams without pools. A Wilcoxon hypothesis test for a difference of means yields a highly significant p-value of 1.514×10^{-19}.

The “BPMN Layout Analyzer” tool also extracts the exporter meta data (which describe the BPMN tool that wrote that file) from BPMN files. When no exporter meta data was found, some heuristics, e.g., namespace names, were used to find a potential BPMN editor. However, there are still some ambiguities, e.g., Camunda and bpmn.io have different names and we do not know for sure whether these name changed in different releases or whether the exporter info was set incorrectly by some other BPMN tool.

We broke down the number of diagrams grouped by layout direction and the BPMN editor as shown in Fig. 4: Nearly all tools have left-right or other layouts only with small exceptions. However, both Activiti and Drools also have a large number of top-down layouts. They are practically the only editors, which have been used to create top-down layouts, although the majority of diagrams created with these tools still follow a left-right layout. We performed a simulated Fisher’s Exact Test (100,000 rounds) to test whether the distribution of layout directions is independent from the modeling tool used. This test yields a highly significant p-value of $p = 9.999\,9 \times 10^{-6}$.

Lastly, we analyzed the layout direction "cleanliness" for the repositories. For each repository we calculated the most dominant layout direction for all diagrams contained therein. Then we calculated the percentage of diagrams that have this layout direction compared to all diagrams within this repository. Thus, cleanliness of 100% means that all diagrams in such a repository have the same layout direction. 85.02% of all repositories had the same layout direction for all their diagrams. Furthermore, we performed a simulated Fisher's Exact Test (100,000 rounds) for the different layout direction distributions against the repositories, which yields a highly significant p-value of $p = 9.999\,9 \times 10^{-6}$.

5 Interpretation

RQ1: What layout directions are used and how is the usage distributed?
We found that the majority (79.52 %) of diagrams are laid out left-right. Although we do not know what the causal reasons are, the left-right layout is predominantly used as recommended by theory, existing guidelines, and the BPMN specification.
RQ2: What influence does the modeling tool have on layout direction?
The hypothesis test is highly significant indicating that the modeling tool has an impact on the diagram layout direction. Interestingly, the Activiti and Drools modelers are responsible for nearly all top-down layouts. However, it is unclear at this point, why these tools have been used for top-down layouts, which warrants further investigation. Many editors have preference for left-right layouts, e.g., Camunda and Signavio. Thus, investigating editor preferences and linking them to actually used layouts can possibly give more insights.
RQ3: What influence does the project ownership has on layout direction?
Layout directions differ highly significantly depending on the project ownership, i.e., the owning repository. While we could not dive deep into the data yet, the differences are highly significant: 85.02% of the repositories followed only one layout paradigm; others had diagrams with different layouts. This means that there are forces which will make diagrams in a projects more similar. Future research can investigate what those forces are (e.g., same developers, guidelines, ...).
RQ4: Are "toy" diagrams laid out differently?
Within the dataset toy diagrams have a highly significantly different layout distribution and are highly significantly smaller with regards to their flow node count. As such we conclude that "toy" diagrams are not representative for the set of "non-toy" BPMN diagrams and future research should be concerned whether to include or exclude those depending on the research questions.
RQ5: Are diagrams with pools laid out differently?
Within the dataset diagrams with pools have a highly significantly different layout distribution and are highly significantly larger with regards to their flow node count. As such we conclude that for future research into BPMN models and diagrams, pool and non-pool diagrams should be researched separately.

Because this is a exploratory study based on existing data without any control, all these correlations can be due to confounding reasons or because they are really causal. Further research is required to establish the relationship type.

6 Threats to Validity

Like in software engineering research a general threat is the usage of GitHub data that might not be representative and generalizable [6]. In fact, we have shown that further analysis must take care of diagram types. Also, due to manual nature of the layout classification other researchers might come to other results. We also experienced problems with the diagram interchange information in the BPMN files, which can skew the results to more reflect compliant editors. Lastly, the tool distribution found on GitHub does not match those found in organizations (e.g., IBM, SAP, and Oracle tools are missing; Signavio is underrepresented etc.). Some BPMN models had more than 1 diagram, which we did not evaluate. On a technical note, the exporter information in the BPMN diagrams itself are not as reliable as one would hope: Missing information and ambiguous names are possible sources of error.

7 Conclusion & Outlook

Within this paper we have shown that the majority of BPMN diagrams found on GitHub are laid out left-right. The results suggest that the type ("toy" or "non-toy") of a process model influences the size and the layout direction. We have found that further research into tool usage is warranted as nearly all top-down diagrams are laid out using only two editors. Furthermore, we have shown that most – although not as many as expected – repositories only contain diagrams with one layout.

This work opens up new research angles: 1) How can "real" models be separated from "toy" models automatically? Our heuristics of using key words in the file path is a first approximation but while manually classifying the layouts, we also encountered other diagrams (empty or default labels, unfinished diagrams, . . .) that should possibly excluded. 2) The exact causal relationships between model properties (size, pools), modeling tooling and the diagram layout needs to be researched. We showed correlations but no causal relationships in this work. 3) This study should be replicated and compared to model repositories from larger organizations created by process modelers in their respective environments. 4) Formalization of layout direction and automation of its detection in order to scale to larger datasets and make the classification more objective. All in all, this explorative work has laid the foundations for answering these questions.

Bibliography

1. Angela Birchler, Elisabeth Bosshart, Mike Märki, Peter Opitz, Jürg Pauli, Beat Rigert, Yves Sandoz, Marc Schaffroth, Nicki Spöcker, Christian Tanner, Konrad Walser, and Thomas Widmer. eCH-0158 BPMN-Modellierungskonventionen für die öffentliche Verwaltung. WWW: https://www.ech.ch/dokument/fb5725cb-813f-47dc-8283-c04f9311a5b8, September 2014.
2. Kathrin Figl and Mark Strembeck. On the importance of flow direction in business process models. In *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, pages 132–136. IEEE, 2014.
3. Kathrin Figl and Mark Strembeck. Findings from an experiment on flow direction of business process models. In *EMISA 2015*, 2015.
4. Thomas Heinze, Viktor Stefanko, and Wolfram Amme. Bpmn in the wild: Bpmn on github. com. In *Proceedings of the 12th ZEUS Workshop on Services and their Composition*, pages 26–29. CEUR-ws. org, 2020.
5. Thomas S. Heinze, Viktor Stefanko, and Wolfram Amme. Mining bpmn processes on github for tool validation and development. In Selmin Nurcan, Iris Reinhartz-Berger, Pnina Soffer, and Jelena Zdravkovic, editors, *Enterprise, Business-Process and Information Systems Modeling*, pages 193–208, Cham, 2020. Springer International Publishing.
6. Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. The promises and perils of mining github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 92–101, New York, NY, USA, 2014. Association for Computing Machinery.
7. Daniel Lübke, Ana Ivanchikj, and Cesare Pautasso. A template for categorizing empirical business process metrics. In *Business Process Management Forum - BPM Forum 2017*, 2017.
8. Daniel Lübke, Tobias Unger, and Daniel Wutke. Analysis of data-flow complexity and architectural implications. In Daniel Lübke and Cesare Pautasso, editors, *Empirical Studies on the Development of Executable Business Processes*, pages 59–80. Springer, 2019 (to be published).
9. Bruce Silver and Bruce Richard. *BPMN Method and Style*, volume 2. Cody-Cassidy Press Aptos, 2009.