Validation of Algorithmic BPMN Layout Classification

Elias Baalmann[1,*], Daniel Lübke[1,2]

[1]Digital Solution Architecture, Hannover, Germany
[2]Leibniz Universität Hannover, FG Software Engineering, Hannover, Germany

Abstract
For many use cases it is handy to clearly and possibly automatically classify the layout direction of BPMN processes, e.g., in empirical research. We want to validate whether our previously proposed classification and algorithm [1] delivers good, reproducible, and helpful results. To accomplish this, we compare the classification algorithm to a previously manually classified large data set of BPMN processes on GitHub. Our results show that the algorithm classifies BPMN layouts similar to manual classification and is suitable for large data sets due to its good run-time characteristics.

Keywords
BPMN, Diagram Layout, Diagram Layout Formalization, Diagram Layout Detection, Flow Layout, Algorithm Validation

1. Introduction

BPMN is the lingua franca for business process modeling and has many use cases. Its main purpose is to convey information between different stakeholders and as such understandability is a key quality feature of BPMN models. Consequently, much research has focused on analysing the impact of different model aspects [2]. This includes the influence of diagram layout [3]. Recently, analysis of large process model repositories like GitHub [4, 5, 6, 7] have become an interesting research direction because large process repositories allow for better statistical results. However, analysing large data sets require much manual work. To address this issue we started to formalize layout directions of BPMN models [1] for improving comparability of studies.In the next step we implemented a tool automating this classification. Within this paper we validate the formalization and the tool implementation by running it against the data set from a previous study [7], comparing the classification results and analyzing the run-time characteristics of our tool.

ZEUS 2023: 15th Central European Workshop on Services and their Composition, February 16–17, 2023, Hannover, Germany
*Corresponding author.
elias.baalmann@digital-solution-architecture.com (E. Baalmann);
daniel.luebke@digital-solution-architecture.com (D. Lübke)
0000-0002-1557-8804 (D. Lübke)

S. Böhm and D. Lübke (Eds.): 15th ZEUS Workshop, ZEUS 2023, Hannover, Germany, 16–17 February 2023, published at http://ceur-ws.org

2. Related Work

One of the first questions that arises in our context is how BPMN diagrams are laid out by practitioners. Effinger et al. [8, p. 400] state that "[i]n BPMN diagrams the flow direction is usually top-to-bottom or left-to right." This statement is empirically validated by Lübke & Wutke [7, p. 52], who found that 79.52% of BPMN diagrams on GitHub are laid out left-to-right. They also identified other layouts, like most prominently, top-down layouts and more complex layouts like multi-line and snake layouts.

A more theoretical approach is taken by Figl & Strembeck. [9, p. 60] who state that "[b]asically, there are four main options for the overall direction: left-to-right, top-to-bottom, bottom-to-top, right-to-left.", i.e., they take all four possible main directions as principal layout directions. However, they have also added that "zigzag models" should be subject to future research, thereby recognizing the use of more complex layouts in practice.

All modeling guidelines we found recommend left-to-right layouts, e.g., the Swiss standard eCH-0158 for eGovernment [10]. Even the BPMN specification itself favors left-to-right modeling [11, p. 42]. Also Corradini et al. [12, p. 49] define a guideline (number 43) that process modelers should make their models long and thin by aligning all edges with a general workflow direction as much as possible.

However, more recently, a study by Lübke et al. [3, p. 127] has shown that the understandability of large diagrams profits from more complex layouts like snake or multi-line layouts to avoid the penalty of scrolling these diagrams on screen. For the case of smaller diagrams, this experiment found a slight advantage for left-to-right layouts in contrast to top-down layouts, affirming Figl & Strembeck's earlier experiment. However, the findings are either minimal (some understandability metrics in the former experiment) or not significant (some metrics in the former experiment and all metrics in the latter experiment).

3. Research Questions

In this paper, we want to answer the following research questions.

RQ1: Does the algorithm proposed by Baalmann & Lübke [1] classify diagrams comparable to manual classification?

RQ2: Is the classification tool suitable to analyze large data sets?

4. Automatic Flow Layout Classification

In his thesis, Baalmann proposes a hierarchy of flow layouts with three levels [13]. The first level describes the base layout: Straight, L, Multi-Line, Stairs, Snake, U, and Z.

The second level differentiates the layouts by orientation. Possible orientations vary for each base layout based on its symmetry. For example, the Straight layout has four orientations: Straight-N, Straight-E, Straight-S, and Straight-W. In this context, compass directions describe the layout direction of the diagram, with E representing left-right, S representing top-down, etc. Table 1 (appendix) lists the possible orientations for each base layout, along with a brief explanation.

In the third level of the hierarchy, minor variations in the layout are examined. These variations typically involve deviations from the general flow of the diagram. However, we will not be considering this level here, as it would go beyond the scope of this paper.

To automatically classify a BPMN diagram based on its flow layout, Baalmann and Lübke use a modular algorithm that reads the BPMN file, splits it into layout paths that connect a start event to an end event which are then processed individually before the results for the paths are finally combined to an overall flow layout of the diagram. To classify a path, it is further split into a vector chain that connects the elements on the path. After simplifying the chain and discretizing the directions of the vectors, a number of regular expressions are used to determine the flow layout of the path.[1]

5. Automatic vs. Manual Classification

In our comparison we will be compare the classification of our tool with the manual classification of 5297 diagrams by Lübke & Wutke [7]. It should be noted that the automation is not intended to replicate the results exactly: diverging results should not necessarily be interpreted as errors on either side. Rather, the goal is to identify reasons for deviations to distinguish the behavior of automated and manual classification.

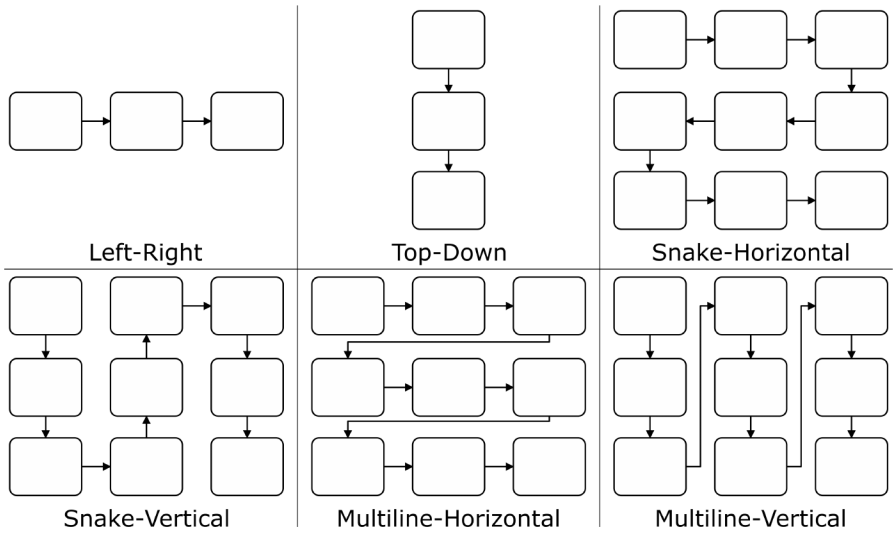

Figure 1: Different flow layouts according to Lübke & Wutke. Figure based on Fig. 2 from [7].

To make comparisons, the flow layouts under consideration must first be matched: The manual classification distinguishes six flow layouts, which are presented in Figure 1. Based on this representation, it is determined that Left-Right corresponds to the Straight-E flow layout. Analogously, Straight-S is the flow layout that is best represented by Top-Down. Since the authors of the manual classification do not specify more precisely which requirements must be met for a particular flow layout, both Snake-ES and Snake-EN are equated with the Snake-Horizontal layout considered by Lübke & Wutke. Following this principle, Snake-Vertical

corresponds to the flow layouts Snake-SE and Snake-NE. In addition, Multiline-Horizontal is compared to Multiline-ES and Multiline-EN, and Multiline-Vertical is compared to Multiline-SE and Multiline-NE. The flow layouts L, Stairs, U, Z, Straight-N, and Straight-W are not considered by Lübke & Wutke.

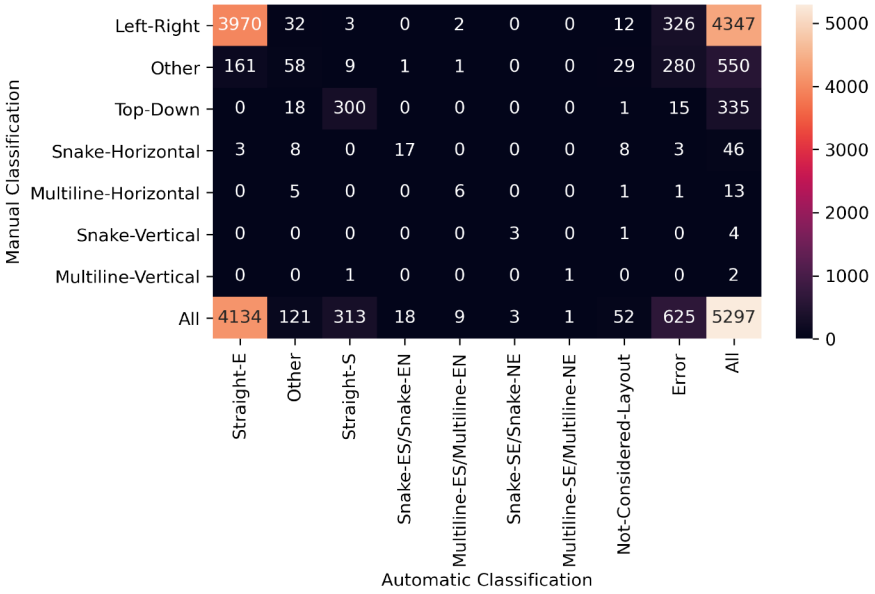

Figure 2: Manual classification of 5297 diagrams compared to automatic classification

Figure 2 shows the distribution of flow layouts per classification for the data set of 5297 diagrams. It is clear that the overwhelming majority of diagrams (manually $4347/5297 \approx 82\%$, automated $4134/5297 \approx 78\%$) are directed from left to right, i.e., classified as Left-Right or Straight-E. It is also notable that the automated variant was unable to classify a large portion of the diagrams ($625/5297 \approx 12\%$) mostly due to invalid BPMN files (for example containing sequence flows between undefined elements), and that a small proportion ($52/5297 \approx 1\%$) of the diagrams were assigned to a flow layout not considered by Lübke and Wutke (Not-Considered-Layout). This raises the question of whether the Not-Considered-Layouts occur frequently enough to be worth considering in future work. However, upon closer examination, it can be seen that in manual classification, only a similarly small proportion ($(46+13+4+2)/5297 \approx 1\%$) of the diagrams were assigned to a flow layout other than Left-Right or Top-Down. Therefore, it is clear that the distribution of the different flow layouts is so uneven that, if flow layouts beyond Straight-E or Straight-S are to be distinguished, a very small number of diagrams must be expected.

Comparing the two classifications, it can be seen that classifications for the straight diagrams (those with the flow layout Left-Right/Straight-E and Top-Down/Straight-S), are very similar. Out of the diagrams manually classified as Left-Right, 91% were automatically classified as Straight-E, and similarly, 90% of those manually classified as Top-Down were automatically classified as Straight-S. However, since 7% and 4% respectively of these diagrams could not

be classified automatically due to errors, it is likely that the agreement is even higher. It is also clear that many (51%) of the diagrams that could not be manually classified as one of the considered flow layouts caused errors in the automated classification. Furthermore, automated classification rarely (about 40%) confirms manual classification when manual classification is Snake-Horizontal, Multiline-Horizontal, or Multiline-Vertical.

Finally it can be seen that the automatic classification in most cases agrees with manual classification when automated classification is able to identify one of the considered flow layouts. The lowest agreement in this sense is for horizontal multi-line variants, at 67%, but it should be noted that many (52%) of the diagrams automatically classified as analyzable (no error) but not classifiable (Other) were manually classifiable.

6. Classification of Large Dataset

To verify that the classification tool is suitable for analyzing large datasets, a large GitHub data set consisting of 48,679 classifiable diagrams is used.

On a desktop PC with an AMD Ryzen 5 3600 CPU, classification of all models takes approximately two hours. However, it should be noted that there are large differences in the time required for each model. The classification of the slowest 10 models took about 99% of the time, while the diagram with the eleventh-longest classification time was classified in less than 30 seconds.

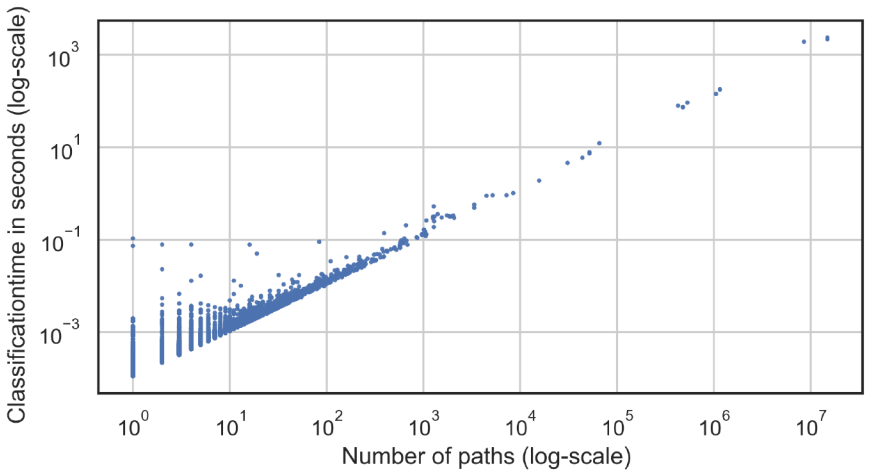

Figure 3: Classification time per model from the GitHub data set. Double-logarithmic representation of the relationship between the number of paths in the model and the classification time.

Figure 3 shows that the run-time of the classification depends on the number of paths in the analyzed model. There seems to be a power law, as the data forms a straight line in the double-logarithmic representation.

To better assess the usefulness of the tool, the statements that can be made about the data set based on automated classification are checked. Figure 4 shows the distribution of basic flow layouts and orientations for the Straight base layout. By far most diagrams have a Straight

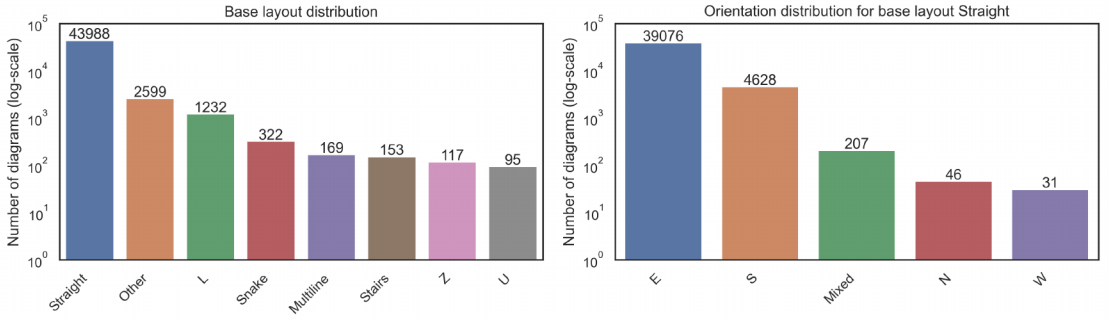

Figure 4: Distribution of flow layouts in large GitHub dataset.

layout (note the logarithmic scale). Furthermore, the orientation E has been assigned to the most diagrams. Specifically, $\frac{39076}{43988} \approx 89\%$ of all diagrams with straight layouts, and therefore $\frac{39076}{48675} \approx 80\%$ of all analyzable diagrams, were classified as Straight-E.

7. Conclusions & Outlook

We could confirm that automated layout classification can be used to analyze large data sets and yields results comparable to manual classification. Besides requiring less effort, automation based on a formalized definition of layouts has additional advantages over manual classification, e.g., it avoids errors due to carelessness and inconsistencies due to subjective perception. However, manual classification currently also has some advantages over automated classification. For example, our tool is not able to complete incomplete diagrams like users with BPMN experience can. In addition, a small number of diagrams are not classified by automation because they do not meet assumptions made in the formalization. It has also been observed that badly laid out diagrams can be better classified manually.

We identified some restrictions of our implementation. For example, models with many paths require a longer processing time possibly making manual inspection more suitable in these cases. However, we encountered hardly any diagrams in our data set, which require a lot of time. If, for example, a time limit of 30 seconds per model had been set, only 11 of the 48679 diagrams would not have been classified. In this case, the total time would have been reduced from about two hours to about five minutes.

Going forward, a way of improving the classification tool would be to reduce the number of not classifiable diagrams by making the implementation less vulnerable to small imperfections in the BPMN files. Another research direction is to extend the validation and get further insights into the problems of the algorithm by using a more diverse data set with diagrams distributed over many different flow layouts. As a side result we could replicate that real-world diagrams are mainly laid out Straight and especially Straight-E (left to right).

We hope that our formalization and tool helps researchers in their empirical studies with BPMN data sets and are open to any cooperation in this regard.

References

[1] E. Baalmann, D. Lübke, Algorithmic Classification of Layouts of BPMN Diagrams, in: CEUR Workshop Proceedings (Ed.), Proceedings of the 14th Central European Workshop on Services and their Composition (ZEUS 2022), volume 3113, 2022, pp. 42–50.

[2] K. Figl, Comprehension of procedural visual business process models, Business & Information Systems Engineering 59 (2017) 41–67.

[3] D. Lübke, M. Ahrens, K. Schneider, Influence of diagram layout and scrolling on understandability of BPMN processes: an eye tracking experiment with BPMN diagrams, Information Technology and Management 22 (2021) 99–131. doi:10.1007/s10799-021-00327-7.

[4] T. Heinze, V. Stefanko, W. Amme, Bpmn in the wild: Bpmn on github. com, in: Proceedings of the 12th ZEUS Workshop on Services and their Composition, CEUR-ws. org, 2020, pp. 26–29.

[5] T. S. Heinze, V. Stefanko, W. Amme, Mining bpmn processes on github for tool validation and development, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2020, pp. 193–208.

[6] J. Türker, M. Völske, T. S. Heinze, Bpmn in the wild: A reprise., in: ZEUS, 2022, pp. 68–75.

[7] D. Lübke, D. Wutke, Analysis of Prevalent BPMN Layout Choices on GitHub, in: CEUR Workshop Proceedings (Ed.), Proceedings of the 13th European Workshop on Services and their Composition (ZEUS 2021), volume 2839, 2021, pp. 46–54.

[8] P. Effinger, M. Siebenhaller, M. Kaufmann, An Interactive Layout Tool for BPMN, in: IEEE Computer Society (Ed.), 2009 IEEE Conference on Commerce and Enterprise Computing, Wien, 2009, pp. 399–406. doi:10.1109/CEC.2009.36.

[9] K. Figl, M. Strembeck, Findings from an experiment on flow direction of business process models, in: J. Kolb, H. Leopold, J. Mendling (Eds.), Enterprise modelling and information systems architectures, Gesellschaft für Informatik e.V, Bonn, 2015, pp. 59–73.

[10] A. Birchler, E. Bosshart, M. Märki, P. Opitz, J. Pauli, B. Rigert, Y. Sandoz, M. Schaffroth, N. Spöcker, C. Tanner, K. Walser, T. Widmer, eCH-0158 BPMN-Modellierungskonventionen für die öffentliche Verwaltung, WWW: https://www.ech.ch/dokument/fb5725cb-813f-47dc-8283-c04f9311a5b8, 2014.

[11] Object Management Group, Business Process Model and Notation (BPMN), Version 2.0, 2011. URL: https://www.omg.org/spec/BPMN/2.0/PDF.

[12] F. Corradini, A. Ferrari, F. Fornari, S. Gnesi, A. Polini, B. Re, G. O. Spagnolo, Quality assessment strategy: applying business process understandability guidelines for learning, 2015. URL: http://pumax.isti.cnr.it/dfdownloadnew.php?ident=cnr.isti/cnr.isti/2015-TR-034&langver=it&scelta=Metadata.

[13] E. Baalmann, Algorithmische Klassifikation der Layouts von BPMN-Diagrammen, Masterarbeit, Leibniz Universität Hannover, Hannover, 01.04.2022.

A. Appendix

The sources for the automatic classification tool are available via GitHub.

Table 1
Flow Layout Hierarchy

Flow Layout	Directions
Straight-N	bottom-up
Straight-E	left-right
Straight-S	top-down
Straight-W	right-left
L-ES	right, then down
L-WN	left, then up
L-EN	right, then up
L-WS	left, then down
L-SE	down, then right
L-NW	up, then left
L-NE	up, then right
L-SW	down, then left
Multiline-ES	lines left-right, each line below previous line
Multiline-WN	lines right-left, each line above previous line
Multiline-EN	lines left-right, each line above previous line
Multiline-WS	lines right-left, each line below previous line
Multiline-SE	lines top-down, each line right of previous line
Multiline-NW	lines down-top, each line left of previous line
Multiline-NE	lines down-top, each line right of previous line
Multiline-SW	lines top-down, each line left of previous line
Stairs-NE	diagonal bottomleft-topright
Stairs-SE	diagonal topleft-bottomright
Stairs-SW	diagonal topright-bottomleft
Stairs-NW	diagonal bottomright-topleft
Snake-ES	first line left-right, each line below previous line
Snake-WN	first line right-left, each line above previous line
Snake-EN	first line left-right, each line above previous line
Snake-WS	first line right-left, each line below previous line
Snake-SE	first line top-down, each line right of previous line
Snake-NW	first line down-top, each line left of previous line
Snake-NE	first line down-top, each line right of previous line
Snake-SW	first line top-down, each line left of previous line
U-ES	left-right, then top-down, then right-left
U-WN	right-left, then bottom-up, then left-right
U-EN	left-right, then bottom-up, then right-left
U-WS	right-left, then top-down, then left-right
U-SE	top-down, then left-right, then bottom-up
U-NW	bottom-up, then right-left, then top-down
U-NE	bottom-up, then left-right, then top-down
U-SW	top-down, then right-left, then bottom-up
Z-ES	left-right, then top-down, then left-right
Z-WN	right-left, then bottom-up, then right-left
Z-EN	left-right, then bottom-up, then left-right
Z-WS	right-left, then top-down, then right-left
Z-SE	top-down, then left-right, then top-down
Z-NW	bottom-up, then right-left, then bottom-up
Z-NE	bottom-up, then left-right, then bottom-up
Z-SW	top-down, then right-left, then top-down