A User Evaluation of Different Visual Representations for Visualizing Differences in Process Models

Firstname Lastname and Firstname Lastname

Universität Wien
Währinger Strasse 29
1090 Wien

Abstract. Finding differences between two processes can be a complex, time consuming, and expensive task. Our work is based on the difference graph approach which calculates the differences between two process models. In this paper we evaluate different possibilities for visualizing these differences. For this purpose we have selected some common visual elements such as color, font size, and line stippling and evaluated these different representations with 31 participants through an online survey. Our results show that color coding was the preferred method of the participants for depicting difference in a graph visualization.

Keywords: Difference Graph, Visualization, Process Differences, Process Model

1 Introduction

Processes are an indispensable part of today's businesses. From visualizing processes for communication, up to optimization companies constantly use processes to gain additional business intelligence.

Using process mining algorithms on data collected during process execution allows for further analysis [6] such as detection of bottlenecks, problems, and violations. Conformance checking [3] is a part of process mining which assesses if a log file deviates from a process model. Unfortunately conformance checking focuses only on detection of deviations between log files and process models. Kriglstein et al. [1], on the other hand, computed the difference model between two process variants and visualized the resulting differences using [XXX] notation. The input models iteself can be generated through process mining or created manually. Visualizing data is a very important task to enhance the users understanding of the data. However, Kriglstein et al. [1] did not evaluate how to best visualize the differences to allow for an effective interpretation.

In this work we address this issue by investigating which visual depictions of differences suit the interpretation of differences in process models best. To answer this question we conducted an online survey with company employees and students and nine different representations gathered through a literature search.

Difference calculation and visualization can be used for different applications in business process management. For example, to compare process models generated with process mining techniques from two processes. This helps finding deviations between these processes. Consider, for example, a company which executes the same task at two different locations. Unfortunately in one location the execution takes twice as long. Comparing the process models of these locations can reveal why the execution time is doubled. When comparing two process models one can also gain additional information if these processes can be merged. Finding deviations allows to determine where problems may occur. Another area of application is to assess one process at different points in time. For example, to evaluate how one process has evolved from one year to the next.

The remainder of this paper is structured as follows. Section 2 gives an overview of the difference graph model and its calculation as well as an outline why visualization is beneficial. Section 3 covers our evaluation. Related work is shortly discussed in Section 4. Section 5 concludes this paper including possible directions for future work.

2 Difference Graph Model and Visualization

The difference graph concept [1] consists of two parts: the model and the visualization of the model. In this section we will first describe the difference model and then focus on visualizing this model.

Elementary component for the difference graph model is a process model. Following [1] a process model can be viewed as a directed connected graph $PM = (N, E \subseteq N \times N)$ where N is a set of nodes and E is a set of directed control edges. Each node has a unique identifier, a label, and a type (e.g., activity, event, or gateway node). Optionally, nodes and edges may have weights assigned which represent the instance traffic. Instance traffic measures how often a specific activity is executed in real world. A process model consists of one start node which has no incoming edges and one end node which has no outgoing edges. Except from start and end node each node is connected by at least one incoming and outgoing edge. Every node has to be on a path between start and end node. As an example, Figure 1 shows two versions of a process model. When visually comparing these two variants one can observe that, for example, that B has been deleted from the right model while a new node E has been added. However, with increasing size of the model, manually finding the differences and commonalities can become increasingly cumbersome and time-consuming.

For generating the difference graph two process models are needed. These models can be described in many different process modeling languages, for example, Petri Nets [2], UML [4], EPC [5]. Generally, all languages which allow to view a process as (weighted) directed graph can be used for the difference graph calculation. For this purpose, the method takes two graphs as input, in the following referred to as PM_1 and PM_2, and generates a new graph $PM_2 - PM_1$

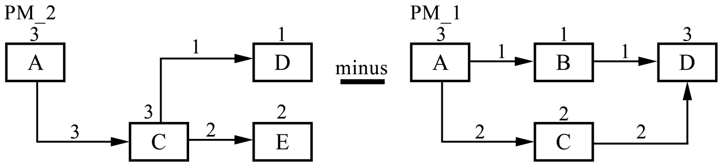


Fig. 1. Two process models.

with markings associated with its nodes and edges (for an in-depth discussion see [1]). The following list shows the different markings and describes in which case these markings are assigned.

New a node/edge is marked as New if the node/edge was added from PM_1 to PM_2.

Increased a node/edge is marked as Increased when the weight has increased from PM_1 to PM_2.

Unchanged a node/edge is Unchanged if it appears in both input models (and has the same weights).

Decreased a node/edge is marked as Decreased when the weight has decreased in PM_2.

Deleted a node/edge is mark as Deleted if the node/edge was deleted from PM_2.

As a side note it should me emphasized that the markings are only used if the input models have weights assigned to nodes and edges.

Figure 2 shows the resulting graph when the left model in Figure 1 is subtracted from the right model. For example, Node B was deleted from PM_2 and therefore is marked as deleted while the weight of node C has increased from 2 to 3 and thus C has been marked as Increased.

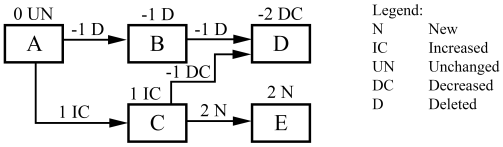


Fig. 2. Calculating the differences from Figure 1 leads to this difference graph. On top of each edge and node calculated weights and applied markings are shown.

The main focus of this paper is how to best visualize the difference graph to promote an effective interpretation by users. The basic idea is to depict the various markings differently, for example, by using icons or color-coding. To do so we conducted an evaluation which is shown in the next section.

3 Evaluation

To address the research question, stated in Section 1, an online survey was conducted. We used the information found with literature research as input for our survey. The survey should show which from the found visualizations fits the difference graph best.

3.1 Literature Research

In order to identify different visual possibilities to distinguish between different markings we conducted a literature search. Our first step was to find relevant keywords addressing the topic of difference visualization. These keywords where used within the search engines Google scholar and Microsoft academic. This led to nine paper which where used as a basis for snowballing method where on one side other relevant paper where collected and on the other side new keywords where extracted, all used keywords are shown in Table 1. Overall snowballing led to additional 22 paper. We read those 31 paper and extracted every visualization which allows five different styles. Five styles are necessary to be able to visualize each marking with an own style e.g. different colors, shapes, symbols. From all collected papers we were able to extract nine visualization approaches. Figure 3 shows all nine approaches used in our evaluation.

Graph differences	Calculate graphical differences	Matrix difference calculation
Delta Analysis	Union of Models	Difference computation
Visual Difference computation	UML diagram differences	Merging business processes
Merging processes	Structural differences between graphs	Visual graph comparison
Visualizing Differences between Graphs	Change detection	Change visualization
Graph evolution visualization	Visualization properties	

Table 1. List of keywords used for literature research.

3.2 Design

To avoid massive scrolling and simplify navigation through the survey we used single pages. Single pages come with the advantage that on each commit of a page we are able to validate the answers and give hints to the user where answers are missing. Overall the survey was divided into three groups: introduction, styles, and advanced/demographic questions.

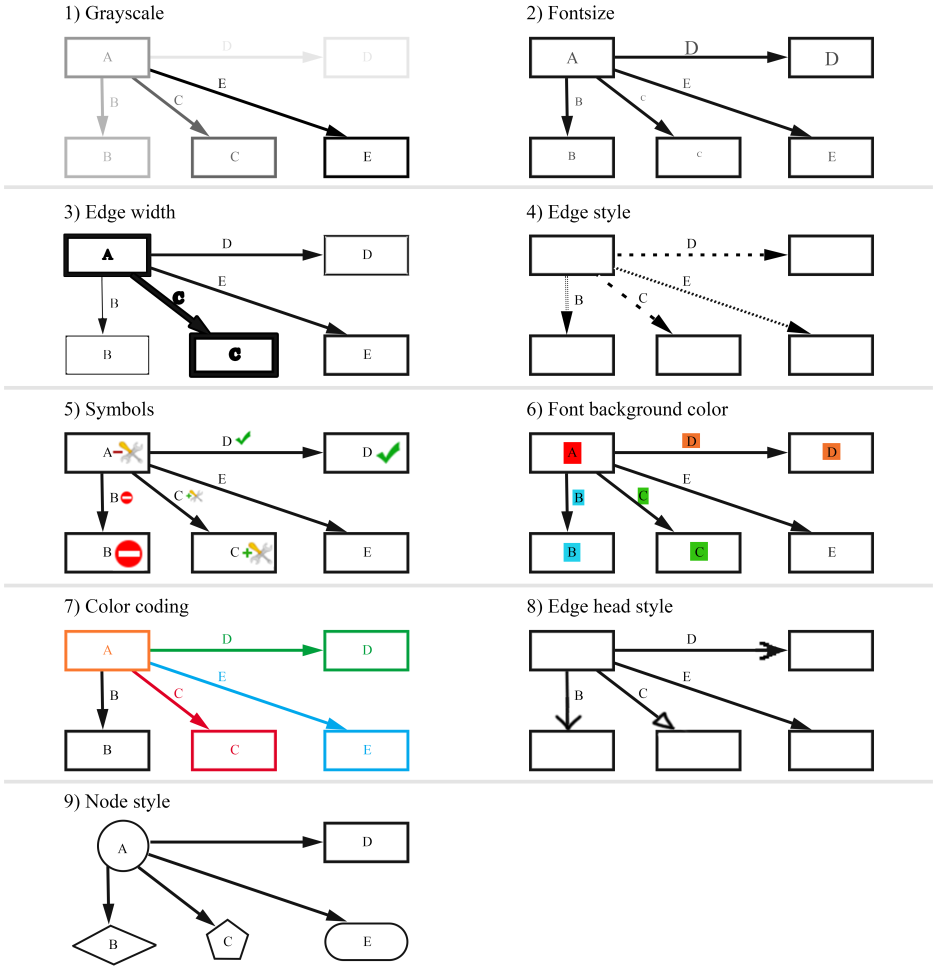


Fig. 3. Visualizations

The introduction gave an example how the difference graph is calculated and contained a question specific to this example to verify if the participant has understood the difference graph concept or not. Further questions on the introduction page asked about knowledge about graphs.

Following the introduction are questions about the visualizations themselves. Each visualization represents each of the five markings in a different way. For example, the visualization *Color coding* uses the colors green, blue, black, orange, and red to distinguish between the markings. Each of this visualizations is presented on a single page which contains two questions related to the visualization. One question required the participants to rate the expressiveness of the visualization on a five-point scale from good to poor. The second question asked the participants to assign each visual element to the marking for which they think the element is best suited. For example, the participants had to assign a red colored node to the marking they deem most suitable for it, such as DELETED.

In the advanced/demographic section we asked the participants to rank the visualizations according to their expressiveness. In addition, participants were asked if they prefer to use the same visual representation regardless of the graph's size or if the representation should be determined by the size of the graph. On our last page participants where asked if they have worked with business processes before. Afterwards the survey concluded with demographic questions about, for example, age, gender, and employment.

3.3 Procedure

First, a two-stage pretest to validate the design of the survey was conducted to ensure the questionnaire is understandable and time required to complete the survey is adequate. In the first pretest a discussion with two people took place to asses the wording of the questionnaire. After changing the survey the user feedback a second pretest with five participants was conducted. During their survey they where encouraged to think aloud, to ask questions, and to provide feedback for possible improvements. All their questions and comments where noted and analyzed after the pretest. After finishing the second pretest minor changes to the questions were made.

After finishing the two pretests the link to the survey was distributed by e-mail to 159 company employees and students. Additionally the link was posted on Facebook addressing students on an private group initiated from and for business informatics students with 537 likes. Overall 73 attendees started the survey from whom 31 attendees finished. Incomplete surveys were excluded from our analysis. In the end, we received 31 complete responses.

3.4 Results and Discussion

Our participants where spread from 19 to 46 years. One participant was below 20 years, 11 participants where in range from 21 to 25 years, although 11 participants from 26-30, one was 36 years, three where 44 and one 46 years. Three people did not fill in their age. We had 24 men (77.4 %), 6 women (19.4 %) and 1 participant (3.2 %) with no answer, participating in the survey. Overall 19 participants (61.3 %) had worked with graphs before and 12 participants (38.7) had not. As expected a little less participants had worked with process models. From 31 participants 13 (41.9 %) had worked with process models, 17 (54.8 %) had not worked with process models before and one participant (3.2 %) did not answer the question.

Before we go into details which representation should be used for the difference graph we want to show that the difference graph can be understood by novices as well as experienced persons. 61.3% of the participants understood the concept by the very first example. Another analysis shows that 73.6% of the participants with fundamental graph knowledge understood the example. In contrast, only 41.6% of the participants without graph knowledge understood the example.

To find the best suiting representation we analyze the intuitive understanding and the ranking for each style. The intuitive understanding is important to allow users to gain an overview of the visualization without the need to look which marking is represented by which style. The goal is to find a visualization where each style can be intuitively allocated to a marking.

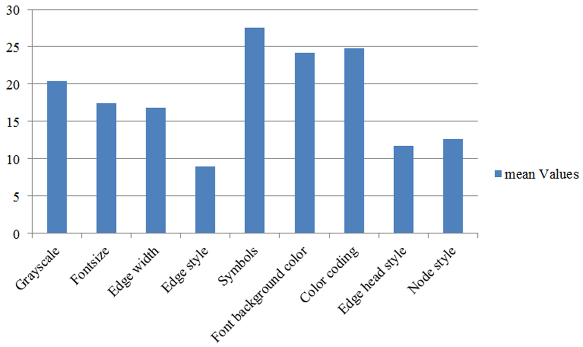
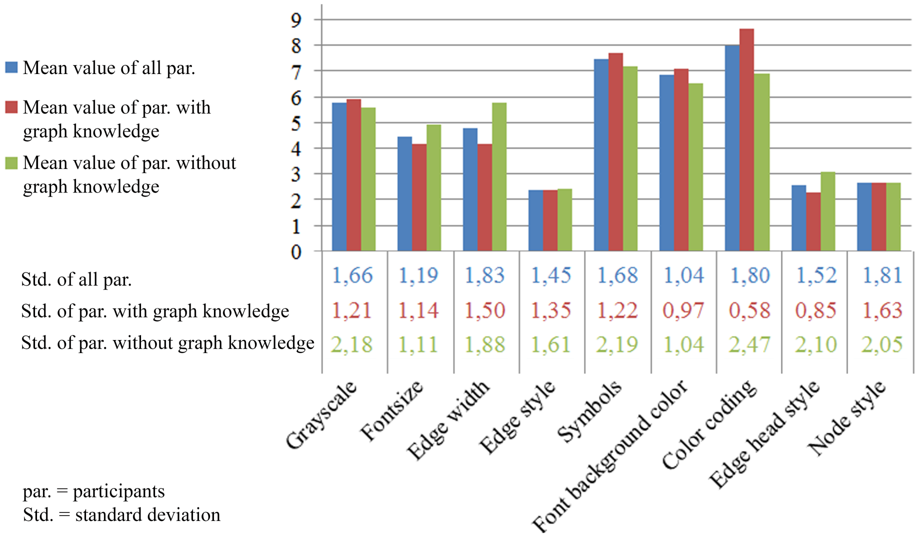


Fig. 4. Intuitive understanding of all styles.

Figure 4 shows all visualizations in terms of intuitive understanding. The best visualization is *Symbols* followed by *Color coding* and *Background color*. The maximum intuitive understanding possible is 31, which can only be achieved if every participant assigns the same style to the marking. Symbols range from 27 to 29 with a mean value of 27,6. For example, the green tick (Figure 3.5 Symbols) was assigned by 27 participants as marking NEW. Color coding has a mean value of 24,8 and *Background color* of 24,2.

After seeing all visualization approaches the users task was to order them according to their expressiveness. The ranking was transformed into numbers best ranking counts 9 worst 1. To compare the rankings the mean-value and



	Grayscale	Fontsize	Edge width	Edge style	Symbols	Font background color	Color coding	Edge head style	Node style
Std. of all par.	1,66	1,19	1,83	1,45	1,68	1,04	1,80	1,52	1,81
Std. of par. with graph knowledge	1,21	1,14	1,50	1,35	1,22	0,97	0,58	0,85	1,63
Std. of par. without graph knowledge	2,18	1,11	1,88	1,61	2,19	1,04	2,47	2,10	2,05

Fig. 5. Ranking of all styles with standard deviation.

standard deviation are calculated. Figure 5 shows the mean value for all representations as bar chart. Besides the mean value of all participants the figure although shows the mean value accumulated for people with and without graph knowledge. As before the figure shows the representations *Color coding*, *Symbols* and *Background color* as best three. This time the order is different now *Color coding* is on top followed by *Symbols* and *Background color*. Figure 5 although shows for each visualization the standard deviation. An interesting observation is that except from *Font size* the standard deviation is lower by participants with graph knowledge then by participants without. A good example to underline this is the representation *Color coding* where the deviation by participants without graph knowledge is 2.47 and 0.58 by participants with graph knowledge.

Within both evaluations only three of nine visualizations are on top. *Background color* was found in both evaluations on third place. Participants commented that *Background color* is alike *Color coding* with the hint that *Color coding* is more useful then *Background color*. Participants mentioned *Color coding* is more pleasant to look at and more suitable for small as well as big graphs. While *Background color* is not suitable for big graphs. The reason for this is *Background color* only colorizes the label of a node. In big graphs finding the correct label for a node can be challenging. Therefore only the visualizations *Color coding* and *Symbols* are considered for further analysis.

Both visualizations offer advantages over the other. For *Symbols* the same negative aspect as for *Background color* can be encountered. Within big graphs finding the symbol assigned to a specific edge can be challenging. One solution would be to change visualizations with the size of the graph. Our evaluation showed that only 29% of the participants recommend to change the visualization with the graphs size.

We consider *Color coding* as the best visualization. The visualization offers a good intuitive understanding, scored best in the overall ranking and is suitable for small and big graphs. The standard deviation in the overall ranking shows that participants with graph knowledge had a common understanding. As commented by our participants we recommend using a legend to show which color represents which marking.

Figure 6 shows an example how the final representation can look. The figure shows the graph which was calculated in the previous example with applied color coding.

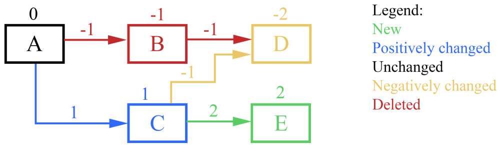


Fig. 6. Color coded representation of the difference graph

4 Related Work

Conformance checking [3] is a part of process mining [6, 7] which deals with detection of inconsistencies between a process model and a log file. Conformance checking lacks of the possibility to compare two process models generated from log files. Kriglstein et al. [1] proposed the difference graph concept where instance traffic of two processes is compared. They implemented a prototype where differences are visualized through color coding. They did not evaluate if color coding is a suitable option for visualizing differences within process models.

One of the first visualization approaches for process models where Petri Nets [2]. Since then many different languages emerged [4, 5, 8]. From process mining perspective the topic of visualization has also been addressed in [9]. For the difference graph we need to show one process model which visualizes the differences of two process models. Therefore we extend previous work in this field by evaluating how differences can be visualized within process models.

5 Conclusion

In our work we combine the difference graph concept with a visualization component. In the concept paper for the difference graph, color coding was mentioned as the preferred method for visualization. In our survey color coding was one of nine styles which can be used for difference visualization. Our evaluation confirmed there thoughts and showed that color coding is the best method for visualizing differences between two process models. To support the users intuitive understanding we suggest to use a legend which states the meaning of each color.

With these results in hand a prototype implementation of the difference graph concept enhanced with color coding was implemented within the ProM framework. The plug in is currently available in the nightly build of ProM.

The color coded visualization is not limited to only show differences between two process models. Color coding can be applied to show differences in many scenarios. For example, conformance and compliance checking which both in some way try to find differences.

Future work in terms of visualization can be done in several areas. Process models often allow clustering but how does clustering affect the color coded visualization? Our approach only visualizes differences between two models. Therefore another interesting topic would be the visualization of process evolution.

References

1. S. Kriglstein, G. Wallner, and S. Rinderle-Ma. A visualization approach for difference analysis of process models and instance traffic. In *Business Process Management*, volume 8094 of *Lecture Notes in Computer Science*, pages 219–226. Springer Berlin Heidelberg, 2013.
2. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, Apr 1989.
3. A. Rozinat and W.M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64 – 95, 2008.
4. J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education, 2004.
5. W.M.P. van der Aalst. Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639 – 650, 1999.
6. W.M.P. van der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.
7. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713 – 732, 2007.
8. W.M.P. van der Aalst and A.H.M. ter Hofstede. Yawl: yet another workflow language. *Information Systems*, 30(4):245 – 275, 2005.
9. B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The prom framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer Berlin Heidelberg, 2005.