Differencegraph - A ProM Plugin for Calculating and Visualizing Differences between Processes

Manuel Gall[1], Günter Wallner[2], Simone Kriglstein[3], Stefanie Rinderle-Ma[1]

[1] University of Vienna, Faculty of Computer Science, Austria
{manuel.gall, stefanie.rinderle-ma}@univie.ac.at
[2] University of Applied Arts Vienna, Institute Art & Technology, Austria
guenter.wallner@uni-ak.ac.at
[3] Technical University of Vienna, Institute for Design & Assessment of Technology, Austria
simone.kriglstein@tuwien.ac.at

Abstract. The analysis of differences and commonalities between process models or between instances which progressed through the model (henceforth referred to as instance traffic) plays an important role in companies. For example, companies are often confronted with different versions or variants of a process model and hence need methods to identify redundancies or inconsistencies between them. *Differencegraph* is a plugin for ProM which supports the identification of differences and commonalities between process models as well as between their instance traffic. For this purpose a so-called difference graph between two process models and their instance traffic is calculated and visualized. This generated difference graph supports decision making in various business cases such as finding deviations between processes.

Keywords: Differences between Processes, Visualization, Process Model, Process Visualization, Process Mining, Instance Traffic

1 Introduction

Since companies frequently need to analyze and manage different process model versions and variants (e.g., to adapt to new requirements or to optimize business processes) [7], it is necessary to support analysts in finding differences and commonalities between process models. In today's business many processes are orchestrated by information systems. Since such systems often store performed activities within log files, process mining techniques have become increasingly important. In its simplest form such a log file contains information about tasks and traces. Each task represents an activity. A trace represents one process instance and consists of several tasks. All tasks within one trace are stored according to their execution order. These log files can then be used by business

analysts for process model generation. Therefore, not only the analysis of differences and commonalities between process models itself but also between the instances which progressed through the model (i.e., instance traffic [6]) is of interest. For example, the analysis of instance traffic allows analysts to detect more or less executed paths.

In this paper we introduce the plugin *Differencegraph* which has been implemented for ProM [2] – an open-source framework that includes most of the existing process mining techniques as plugins. The plugin allows analysts to compare two process models together with their instance traffic. In contrast to other approaches which focus on comparing process models generated from real world behavior with hand crafted process models, the focus of our plugin is to compare two real world process models with each other. The plugin offers the following main features:

- **Comparison** of process models represented as *Heuristic net* or *Petri net*. These process models can either be generated directly within our plugin by using the heuristics [9] or alpha miner [1] offered by ProM or can be generated by other mining/transformation plugins and then passed as input to our plugin. The process models itself can either be based on two distinct process logs or can be derived from a single process log by splitting it into two parts (determined by user-specified dates).
- **Visualization** of differences with color coding and/or symbols. The visualization also includes two model types which can be visualized: *Heuristic net* (see Figure 1) or *Petri net* (see Figure 2).
- **Interactions** like zooming, panning, and brushing and linking allow for an easy navigation and orientation within our visualization.

In the following we sketch two use cases to give an impression about possible tasks for which the plugin can be of interest:

Use Case 1: As a result of reorganization two departments were merged under a joint management. Prior to that both departments developed and applied their own process variants. After both departments were merged, different process variants for the description of the same process existed. To reduce the number of process variants it is necessary to identify redundant ones or find ways to properly merge them. Analysts can use the plugin to compare the different process variants in order to identify their commonalities and differences.

Use Case 2: A company has a process log which spans two years and is interested in how the process execution has changed from the first to the second year. In this case the comparison of instance traffic enables the analysts to see how the process has changed, e.g., which tasks or paths were executed more or less often. This also allows to see trends which, in turn, can help to optimize and coordinate processes in order to avoid bottlenecks.

2 *Differencegraph* Plugin

The plugin – written in Java – is based on the difference graph and instance traffic concept introduced by Kriglstein et al. [6]. For the calculation of the difference graph, two process models are necessary as input. These models can be generated based on different process logs or by splitting one process log into two parts (as mentioned in the second use case). The resulting difference graph uses five different markings to represent the differences and commonalities: *New*, *Unchanged*, *Deleted*, and – if instance traffic is of interest – *Decreased* as well as *Increased*. A description of the calculation itself can be found in [6]. For the visual representation of these markings color-coding and/or symbols can be used.

Installation. To install the plugin the ProM nightly build [8] is required. After extracting ProM, run the PackageManager and download the *Differencegraph* plugin under the category *not installed*. All required dependencies are automatically downloaded by ProM.

Usage. After starting the plugin the user can either select two log files or a single log file and split it into two parts. Afterwards some initial configuration can be made using a wizard. This includes, choosing which mining algorithm (heuristic or alpha miner) should generate the process models for difference calculation. Furthermore, the visual properties for visualizing the difference graph can be selected (these properties can also be changed later within the visualization). Since traces may not be stored in order of processing, the plugin offers the possibility to order the entries of the log files in ascending order using the *Sort Log* option. This can be of interest when one log should be split into two parts or only specific time spans of two distinct log files should be compared. In the latter case date pickers can be used to select the dates which should be considered for difference calculation.

After the configuration is completed the plugin passes the log files to the chosen mining algorithm to generate the two process models. These process models are then used for the difference calculation (more information about the calculation can be found in [6]) and the resulting difference graph is visualized.

Interface. Figure 1 shows the interface of the plugin and the visualization of the two input models and the difference graph with selected options *Weights on Nodes*[4], *Color Coding*, and *Symbols*. Each node represents a task while each edge represents the control flow from one task to the next. To ease orientation for the users the plugin tries to preserve the mental map [3] as much as possible by trying to arrange the nodes in such a way that their locations do not change significantly between the different views. A legend at the bottom right shows which color/symbol represents which marking. For example, in Figure 1 node *D* (labeled ① in the image) is represented as *Decreased* and therefore drawn in orange along with a symbol. *Decreased* means that the weight associated with

[4] Weights reflect the amount of instances executing a task, i.e., the instance traffic.

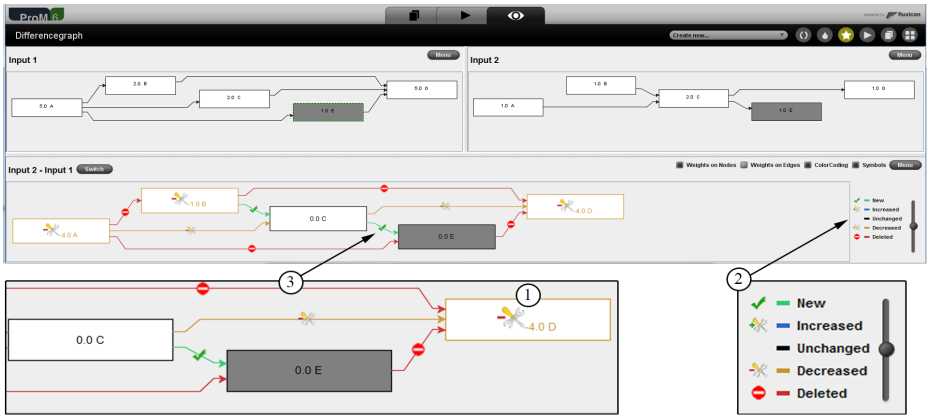

Fig. 1. Screenshot of two input process models (top) and the resulting difference model (bottom). In this particular example, both visual properties *color coding* and *symbols* are used to highlight the differences. ① Node *D* is marked as *Decreased* because the weight was reduced from 5 (*Input 1*) to 1 (*Input 2*). ② A close-up of the legend. For example, the marking *Decreased* is represented in orange. The slider on the right allows to zoom in or out on the graph. ③ A close-up of a few nodes.

B was reduced from *Input 1* to *Input 2* (specifically by -4.0). Each visualization allows zooming and panning. Brushing and linking is also supported, that is, if one or multiple nodes are selected within a graph then the respective nodes are also selected within the other graphs. For example, node *E* was selected in *Input 1* and thus *E* is also selected in the other two views.

A more detailed introduction about the plugin, survey data, screencasts, and an installation guide can be found at the website of the plugin [4].

3 Maturity and Significance to the BPM field

For the visual representation of the differences we conducted a user study [5] to find out which representations are suited to highlight differences in a graph. We investigated nine different types of representations based on different visual properties (e.g., color, symbol, and size). Among these representations, color-coding and symbols were ranked highest by the participants and achieved the highest average rating in terms of expressiveness (see ② in Figure 1 for an example of these properties). As shown in Figure 1-③ our plugin also allows to use both properties simultaneously to highlight the differences.

During development the plugin was tested with log files with up to 25.000 traces and 250 tasks. Figure 2 shows an example of a difference graph visualized as *Petri net* with 23 tasks generated from 2000 traces. However, a higher number of traces and tasks affects response time of the graphical interface and leads to longer computation times for process model generation done by the alpha or heuristics miner.

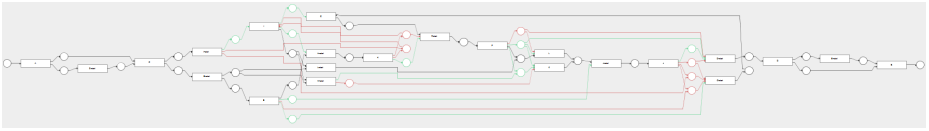

Fig. 2. An example of a large difference graph visualized as *Petri net* with 23 tasks generated from 2000 traces. Differences are visualized using color coding.

Currently we are not aware of any other tool that allows comparison and visualization of instance traffic differences. Conformance checking is closely related to the purpose of our plugin but does not focus on comparing real world processes for insight generation. Due to its focus on real world execution semantics of business processes stored within log files we believe that our *Differencegraph* plugin can contribute to the analysis of process models in various ways, some of which we have shortly outlined above.

Acknowledgments. Simone Kriglstein was supported by CVAST (funded by the Austrian Federal Ministry of Science, Research, and Economy in the exceptional Laura Bassi Centres of Excellence initiative, project nr: 822746).

References

1. van der Aalst, W.: Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media (2011)
2. van Dongen, B., de Medeiros, A., Verbeek, H., Weijters, A., van der Aalst, W.: The ProM framework: A new era in process mining tool support. In: Applications and Theory of Petri Nets 2005, LNCS, vol. 3536, pp. 444–454. Springer (2005)
3. Eades, P., Lai, W., Misue, K., Sugiyama, K.: Preserving the mental map of a diagram. In: Proceedings of COMPUGRAPHICS. vol. 91, pp. 24–33 (1991)
4. Gall, M., Rinderle-Ma, S.: Differencegraph (2015), `http://gruppe.wst.univie.ac.at/projects/diffgraph/`
5. Gall, M., Wallner, G., Kriglstein, S., Rinderle-Ma, S.: A study of different visualizations for visualizing differences in process models. In: Proceedings of Workshop on Event Modeling and Processing in Business Process Management (EMoV2015) in conjunction with 34th International Conference on Conceptual Modeling (ER 2015). Springer (2015)
6. Kriglstein, S., Wallner, G., Rinderle-Ma, S.: A visualization approach for difference analysis of process models and instance traffic. In: Business Process Management, LNCS, vol. 8094, pp. 219–226. Springer (2013)
7. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems: A survey. Data Knowl. Eng. 50(1), 9–34 (2004)
8. Verbeek, E., Günter, C.: Prom nightly build (2010), `http://www.promtools.org/prom6/nightly/`
9. Weijters, A., van der Aalst, W., Alves de Medeiros, A.K.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34 (2006)