

# Process Model Simplification based on Probabilities in Process Tree (Extended Abstract)

Sabya Shaikh, Mohammadreza Fani Sani

*Process and Data Science (PADS) Chair, RWTH-Aachen University, Germany*

sabya.shaikh@gmail.com, fanisani@pads.rwth-aachen.de

**Abstract**—Process discovery aims to describe how a process is actually executed based on recorded data. However, most automated process discovery algorithms result in complex and imprecise process models due to the existence of outlier behavior in the real event logs. The process discovery procedure usually has an exploratory nature and should be done interactively considering users' preferences. This demo paper proposes an interactive ProM plug-in that allows users to simplify the discovered process model. Using this tool, modifying the event log based on the simplified process model is also possible.

**Index Terms**—Process Mining, Process Discovery, Event log preprocessing

## I. INTRODUCTION

Process discovery is one of the sub-fields of process mining that aims to describe process models based on recorded events [1]. The discovered process model is used by the business analysts to comprehend the information available in event logs to strengthen their process. Several process discovery algorithms [2]–[4] discover a process model assuming that event logs are noise-free. However, in reality, event logs contain noise due to wrong or incorrect entry that leads to a complex process model. Therefore, several filtering algorithms are put forth to eliminate these behaviours at a preprocessing stage. The filtered event logs must further be provided to the process discovery algorithms to obtain a simple process model, hence, making it a wearisome procedure. There are also other process discovery techniques that have the in-build capability to eliminate the noise and infrequent behaviour, nevertheless, all of these algorithms use frequency of variants or activities for filtering which is not effective when the event log contains unique variants.

To address this problem, in this demo paper, we aim to provide an interactive tool, that would enhance the process model with probabilistic information and allow the business analysts to adjust the complexity by modifying the process model interactively to improve the process model's precision and simplicity. The pruned process model can be used to filter the event log. In other words, the user can have the modified event log based on the modified process model.

We use the process trees discovered by the Inductive miner [5] as the base process model that is further pruned based on the probabilities of nodes computed by the tool and the threshold provided by the user. A process tree is a process model that represents a process in the form of a tree (however, we could not describe all processes with the process tree

notation). This tool lies in both the discovery and enhancement phases of process mining. It exploits the process exploration feature to provide an interactive and iterative plug-in. The developed plug-in is an extension in the ProM Framework [6] that is a framework that allows the development and supports various process mining algorithms.

This paper is structured as follows. Section II discusses some of the previous related works and motivation to develop the tool. The highlights of the tool, the case studies carried out to evaluate the tool, and the link to demo is mentioned in section III, while Section IV concludes the paper.

## II. RELATED WORK AND MOTIVATION

Numerous process discovery algorithms are applied on an event log assuming all behaviours in the event log are appropriately ignoring the noise. However, this approach generates a complex process model that is difficult to understand [7] resulting in the analysis being skewed. Some basic process discovery algorithms represent all the behaviours in a process model. Hence, this calls for pre-processing of the event log using filtering techniques such as [7]–[10] in order to discover a comprehensible process model. Few of the improvised process discovery algorithms, e.g., [5], have embedded techniques to filter the infrequent behaviours prior to discovering the process model. However, these techniques do not guarantee a simple model always.

Usually, a user provides multiple input parameters and discovers a process model that could be unsatisfactory. This process might have to be repeated until a satisfactory model is discovered which could be exhausting and time-consuming. Furthermore, we usually need to first understand the general behaviour of the process and later dig deeper into the process. Thus, having an interactive and iterative tool would be helpful in many scenarios. In commercial tools, there exist some basic interactive filtering methods that simply work based on the frequency of variants or activities. However, in many applications, because of the existence of lots of unique variants, this type of filtering approach is not beneficial.

## III. TOOL

The tool is developed in the ProM framework that eases the use and integration with other process mining plug-ins. It is a part of the research available at [11].

Fig 1 shows an outlook of the developed tool. The input for the tool is an event log and interactive user-provided

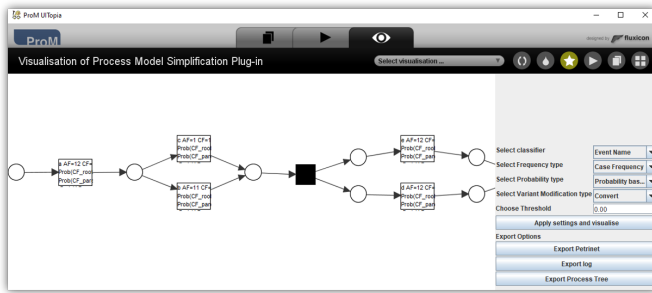


Fig. 1. A screenshot of the developed tool

parameters and the output is a (simplified) process model along with a filtered event log. This tool allows users to set various parameters interactively in the right panel which include the threshold for pruning the process tree, event classifier, frequency types, probability types, and different ways the log can be filtered. We have three resulting outputs, namely, a simplified and enriched Petri net and process tree, and a filtered event log. The Petri net output is displayed on the left panel. Each node in the process tree holds the following information: probabilities of its occurrence in the event log, frequency of the node's execution, and list of traces that the node executes. This information is used to get rid of the noisy and infrequent behavior resulting in a noise-free filtered event log. We formally described how to compute the probabilities and simplify a process tree based on them in [11]

A tutorial video demo of our tool is provided here<sup>1</sup>. A more comprehensive tutorial of this tool is available in [11].

### A. Tool Capabilities

The main capabilities of the developed tool are as follows.

- Simplifying the process model while potentially improve its quality
- Enhancing process model with probabilistic information which is used for simplifying the process model
- Filter and modify the given event log based on the simplified and enhanced process model

### B. Use Cases

We have used the developed tool to simplify the process models of four real event logs, i.e., [12]–[15]. The results show that our tool improves the quality of the process model when simplified. We have found that using the developed tool, it is possible to improve the precision and simplicity of process models of these event logs. Moreover, the developed tool performs in a reasonable time that makes it useful for real applications. For detailed information on the event logs used and the evaluation procedure carried out, please refer to [11].

### C. Installation

Prom 6 nightly build can be used to run and test the plug-in.

- Download and install Prom 6 nightly build available at <sup>2</sup>.

<sup>1</sup><https://www.youtube.com/watch?v=geDKNyq96pA>

<sup>2</sup><http://www.promtools.org/doku.php?id=nightly>

- Open the Prom Manager and install the 'LogFiltering' package.
- Open the Prom tool and import an event log
- Use plug-in "Process model simplification and log filtering" as shown in the recorded demo.

The source code of this project is available at <sup>3</sup>. In this way, it is also possible to modify the approach and also execute it.

## IV. CONCLUSION

We present an interactive tool that enhances the process model with frequency and probability information and enables the simplification of the process model. Users can further filter the event logs based on the simplified process model. The significant contribution of this tool is the simplification of the process model in a novel way such that high fitness and precision are maintained while complexity is reduced.

## REFERENCES

- [1] W. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.
- [2] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, 2004.
- [3] S. J. J. Leemans, D. Fahland, and W. van der Aalst, "Discovering block-structured process models from event logs - A constructive approach," in *Application and Theory of Petri Nets and Concurrency - 34th International Conference, 2013. Proceedings*, vol. 7927. Springer, 2013, pp. 311–329.
- [4] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, "Process discovery using integer linear programming," in *Applications and Theory of Petri Nets, 29th International Conference*, vol. 5062. Springer, 2008, pp. 368–387.
- [5] S. J. J. Leemans, D. Fahland, and W. van der Aalst, "Discovering block-structured process models from event logs containing infrequent behaviour," in *Business Process Management Workshops - BPM 2013 International Workshops*, vol. 171. Springer, 2013, pp. 66–78.
- [6] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. van der Aalst, "The prom framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets 2005*, ser. Lecture Notes in Computer Science, vol. 3536. Springer, 2005, pp. 444–454.
- [7] M. Fani Sani, S. J. van Zelst, and W. van der Aalst, "Applying sequence mining for outlier detection in process mining," in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, ser. Lecture Notes in Computer Science, vol. 11230. Springer, 2018, pp. 98–116.
- [8] R. Conforti, M. L. Rosa, and A. H. M. ter Hofstede, "Filtering out infrequent behavior from business process event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 300–314, 2017.
- [9] M. Fani Sani, S. J. van Zelst, and W. M. P. van der Aalst, "Repairing outlier behaviour in event logs using contextual behaviour," *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.*, vol. 14, pp. 5:1–5:24, 2019. [Online]. Available: <https://doi.org/10.18417/emisa.14.5>
- [10] J. Wang, S. Song, X. Lin, X. Zhu, and J. Pei, "Cleaning structured event logs: A graph repair approach," in *31st IEEE International Conference on Data Engineering*. IEEE Computer Society, 2015, pp. 30–41.
- [11] S. Shaikh, "Process model simplification based on probabilities in process tree," in *Master thesis*. RWTH-Aachen University, 2020.
- [12] F. Mannhardt, "Hospital billing - event log," [https://data.4tu.nl/articles/dataset/Hospital\\_Billing\\_-\\_Event\\_Log/12705113/1](https://data.4tu.nl/articles/dataset/Hospital_Billing_-_Event_Log/12705113/1), Aug 2017.
- [13] Felix.Mannhardt, "Sepsis cases - event log," [https://data.4tu.nl/articles/dataset/Sepsis\\_Cases\\_-\\_Event\\_Log/12707639/1](https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639/1), Dec 2016.
- [14] M. M. de Leoni and F. Mannhardt, "Road traffic fine management process," [https://data.4tu.nl/articles/dataset/Road\\_Traffic\\_Fine\\_Management\\_Process/12683249/1](https://data.4tu.nl/articles/dataset/Road_Traffic_Fine_Management_Process/12683249/1), Feb 2015.
- [15] B. van Dongen, "Bpi challenge 2017 - offer log," [https://data.4tu.nl/articles/dataset/BPI\\_Challenge\\_2017\\_-\\_Offer\\_log/12705737/1](https://data.4tu.nl/articles/dataset/BPI_Challenge_2017_-_Offer_log/12705737/1), Feb 2017.

<sup>3</sup><https://svn.win.tue.nl/repos/prom/Packages/LogFiltering>