

Combining Process Mining and Simulation to Guide Users Towards Process Improvement in mpmX

Nina Kinkelin^{1,2}, Clemens Schreiber¹ and Josua Reimold²

¹Karlsruhe Institute of Technology, Karlsruhe, Germany

²MEHRWERK GmbH, Karlsruhe, Germany

Abstract

Process mining and simulation can be a powerful combination to analyze and improve business processes. While process mining is commonly applied to analyze past process executions (as-is process), simulation allows a user to explore process executions, which might occur in the future (to-be process). Yet, the combination of process mining and simulation is not commonly used in practice. We see two main reasons for this, which we attempt to solve: (1) missing tool support for the creation and execution of process simulations based on event logs, (2) missing guidance for the user-based adaptation of simulation scenarios. Hence, we are introducing a simulation extension to the Mehrwerk ProcessMining software mpmX, which on the one hand enables the automatic discovery and execution of process simulation models based on an event log, while at the same time providing suggestions for the creation of alternative simulation scenarios. The simulation scenarios thereby cover a subset of the discovered process variants from the as-is process and enable a user to analyze the change in process performance based on a more standardized process.

Keywords

Process mining, interactive process simulation, business process reengineering

1. Introduction

The combination of process mining and simulation allows a user to compare process executions in the past (as-is process) with process executions, that might occur in the future (to-be process) [1, 2]. This also allows a user to simulate how process reengineering based on process mining might impact the process performance in the future [3]. As shown in [3] such an approach can improve business processes across multiple domains. Yet, the user support for an integration of process mining and simulation is lacking. We are, therefore, introducing a simulation extension to the existing mpmX process mining tool [4], which supports the integration of process mining and simulation on multiple levels, i.e., the analysis of the as-is process, the creation of a simulation scenario, the execution of the simulation, and the comparison between as-is and to-be process. A user can thereby actively influence the creation of the simulation scenario based on the selection of relevant process variants, the number of cases to be simulated, the maximum trace length for each simulated case, and the case arrival ratio. We show in a use

ICPM 2023, October 23–27, 2023, Rom, Italy


✉ nina.kinkelin@mehrwert.net (N. Kinkelin); clemens.schreiber@kit.edu (C. Schreiber);

josua.reimold@mehrwert.net (J. Reimold)

🆔 0000-0002-6256-9708 (C. Schreiber)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

case, that our tool is able to generate suggestions for simulation scenarios, which can lead to process improvement in terms of throughput and waiting time, and allows for a comprehensive comparison between as-is and to-be process. A video tutorial with a demonstration of the tool is available at <https://vimeo.com/user167009028/mpmxsim?share=copy>.

2. Guided Process Simulation

Our guided process simulation approach consists of five main steps (Fig. 1): 1. load and analyze an event log in mpmX, 2. define the simulation scenario, 3. create the simulation model, 4. execute the simulation, and 5. analyze the simulation result. The user is guided through each step by the mpmX tool and the extension. In the following, we will shortly describe each step.

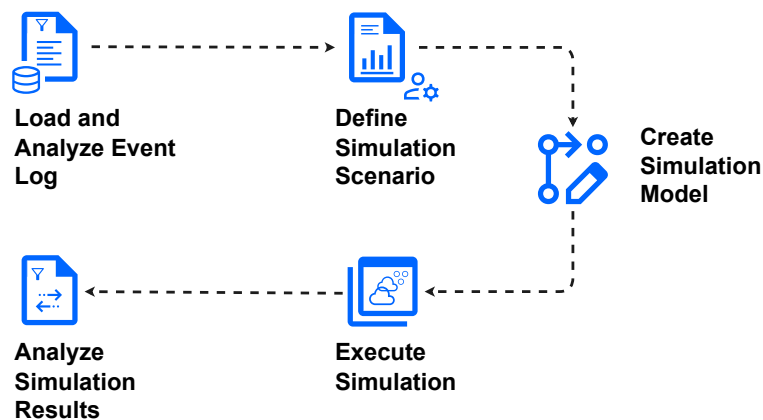


Figure 1: Process mining and simulation as implemented in mpmX.

2.1. Load and analyze event log

After loading an event log into mpmX, the mpmX process variant analyzer detects all existing process variants, i.e., event sequences in the event log. In addition, the process variants are analyzed based on their frequency, i.e., the number of occurrences in the event log, and their average lead time. This information is provided to the user in a process variant overview (see Fig. 2).

2.2. Define simulation scenario

Based on the process variant overview, the user is able to select relevant process variants for the simulation. The user can for example exclude process variants with a low frequency and a high lead time, if she assumes, that these variants could be omitted in the future. In this way the process simulation model ends up being a more standardized version of the as-is process. By hovering the cursor over the different variants, the user can also see the specific event sequences. In this way the user is able to keep relevant variants that might be essential to the process, while at the same time eliminating non-relevant variants with low performance. The

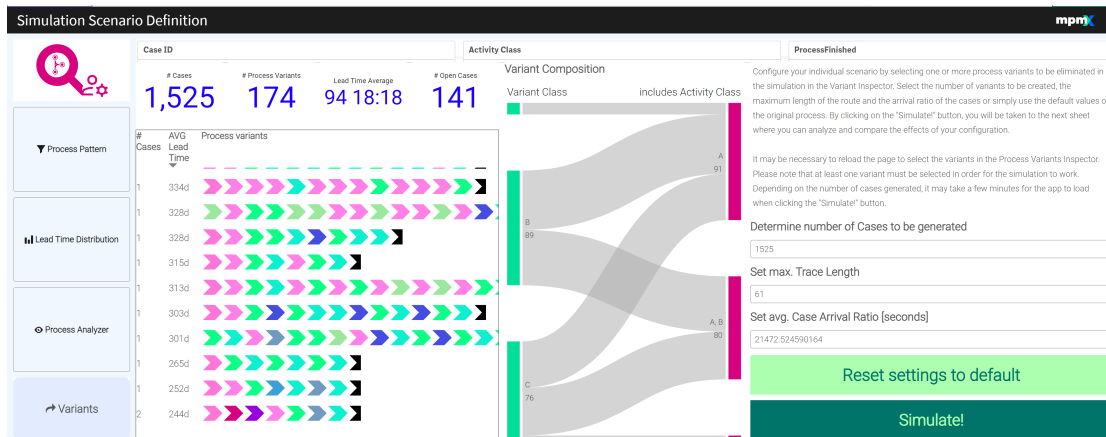


Figure 2: Process variant overview for the generation of simulation scenarios.

effect of the selection of process variants on the overall process performance thereby depends on how the eliminated variants are replaced by alternative variants (see Sect. 2.3).

Further simulation parameters, which can be adjusted by the user, are: the number of cases to be simulated, the case arrival ratio, and the maximal trace length. These parameters allow the user to investigate how the process performance changes, based on varying basic simulation conditions and constraints. It is important to notice that the simulation also considers the probabilistic distribution of process variants. Hence, the higher the number of cases to be simulated, the more similar will be the process variant distribution to the distribution observed in the event log. The default values for the respective parameters are calculated based on the provided event log.

2.3. Create simulation scenario

The simulation model is created based on the selected process variants in step 2 (see Sect. 2.2). The elimination of process variants requires a redistribution of relative case frequencies among the remaining process variants. This redistribution is solved based on a mapping algorithm, i.e., the relative case frequency of the eliminated variants is mapped to the most similar variants, in terms of some distance measure. In this way the relative case frequency of the eliminated process variant is added to the most similar process variant. This approach assumes that if a process variant is eliminated, it will be replaced by the most similar process variant in the future. In the current implementation, the Levenshtein distance is used to identify the similarity between the process variants in the event log, but also other distance measures are possible. If there are multiple process variants with an identical minimum distance to an eliminated variant, then the relative case frequency is randomly distributed to one of them.

2.4. Execute simulation

Based on the selected process variants in step 3 (see Sect. 2.3) a Petri net is discovered, which is then used for discrete event simulation. Our code is based on the implementation provided

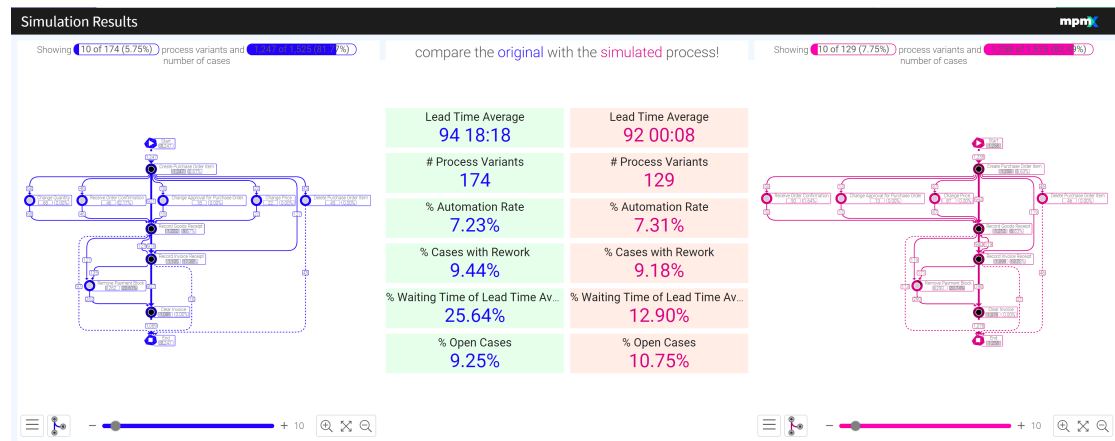


Figure 3: Overview of simulation results and comparison to as-is process.

by PMSIM [2] and uses the python libraries PM4Py [5] for process discovery and SimPy¹ for discrete event simulation. The main difference between our implementation and PMSIM is that we do not want to create new process variants during the simulation, which were not selected by the user during the creation of the simulation scenario. This is achieved by selecting only those execution paths in the Petri net during the simulation, which also occur in the event log. In this way it is assured that the simulation actually considers a more standardized (“improved”) version of the as-is process and does not involve new deviations.

2.5. Analyze simulation result

Finally the simulation results are evaluated, i.e., to which extent the standardization, based on the selected process variants, might lead to a performance improvement. The simulated data is reloaded into mpmX and the resulting process model, together with different performance indicators is shown in an overview. This overview also allows for a direct comparison between the as-is process and the simulated process (see Fig. 3).

3. Maturity

The mpmX simulation extension was tested based on a real-life event data set (BPIC’19). The data set consists of 1525 cases and 174 process variants. By eliminating the 10 process variants with the highest lead time and a frequency of one, and without changing any other parameters, the simulation shows an improvement in average lead time from 95 days to 92 days, an increase of the automation rate by 0.08 percentage points and a reduction of wait time by 12.74 percentage points. The experiment was run on an Intel i5 CPU @ 1.60GHz machine with 8GB RAM. The generation and execution of the simulation model took about 74 seconds. When no process variants are eliminated, i.e., in case of highest computational complexity, the generation and simulation took about 80 seconds. This might indicate that our approach is also feasible for the

¹<https://simpy.readthedocs.io>

application in industry, although further testing is needed.

4. Outlook and Conclusion

As future work we would particularly like to add two features to the current version: 1.) the ability to add additional new process variants to the simulation model, which were not in the as-is process based on the event log, 2.) an evaluation of the mapping accuracy, i.e., if the similarity between the traces is rather high or rather low. This would also provide a better assessment on the reliability of the simulation. Furthermore, we would also like to integrate additional process perspectives, such as the resource [6] and the data perspective [7]. However, based on the current version of the mpmX simulation extension we could show that our proposed simulation approach is viable and can provide additional guidance and insights to existing process mining analysis.

References

- [1] van der Aalst, W. M. (2018). Process mining and simulation: a match made in heaven!. In Proceedings of the 50th Computer Simulation Conference (pp. 1-12).
- [2] Pourbafrani, M., Vasudevan, S., Zafar, F., Xingran, Y., Singh, R., and van der Aalst, W. M. (2021). A Python Extension to Simulate Petri nets in Process Mining. arXiv preprint, arXiv:2102.08774.
- [3] Märuşter, L., and Van Beest, N. R. (2009). Redesigning business processes: a methodology based on simulation and process mining techniques. Knowledge and Information Systems, 21, 267-297.
- [4] Meyer, J., Reimold, J., and Wehmschulte, C. (2019). An introduction to MPM - MEHRWERK ProcessMining, CEUR Workshop Proceedings, vol. 2374.
- [5] Berti, A., Van Zelst, S. J., and Schuster, D. (2023). PM4Py: A process mining library for Python. Software Impacts, 17, 100556.
- [6] López-Pintado, O., Halenok, I., and Dumas, M. (2022). Prosimos: Discovering and Simulating Business Processes with Differentiated Resources. In International Conference on Enterprise Design, Operations, and Computing (pp. 346-352). Cham: Springer International Publishing.
- [7] Fritsch, A., Schüler, S., Forell, M., and Oberweis, A. (2023). Modelling and Execution of Data-Driven Processes with JSON-Nets. In International Conference on Business Process Modeling, Development and Support (pp. 29-43). Cham: Springer Nature Switzerland.