

Simod: A Tool for Automated Discovery of Business Process Simulation Models

Manuel Camargo^{1,2}[0000-0002-8510-1972], Marlon Dumas¹[0000-0002-9247-7476],
and Oscar González-Rojas²[0000-0002-8296-6620]

¹ University of Tartu, Tartu, Estonia, {manuel.camargo, marlon.dumas}@ut.ee

² Universidad de los Andes, Bogotá, Colombia, o-gonzal@uniandes.edu.co

Abstract. Business process simulation is a widespread approach for quantitative analysis of business processes. However, the creation of accurate business process simulation models is a laborious and error-prone task, due to the numerous parameters that need to be carefully tuned. Additionally, the accuracy of a simulation model is inherently limited by the accuracy of the process model that is used as a starting point. This paper presents Simod: A tool to automatically generate simulation models from event logs. Simod uses an automated process discovery technique to extract a process model from an event log and then enhances this model with simulation parameters extracted via a combination of trace alignment, replay, and curve-fitting techniques. The tool incorporates a Bayesian hyperparameter optimization technique to fine-tune the accuracy of the resulting simulation model.

1 Significance

Business process simulation (BPS) is a widespread technique for quantitative analysis of business processes [3]. Traditionally, business process simulation models are manually created by domain experts via data gathering techniques such as interviews, observation, and sampling. This approach makes the creation of simulation models time-consuming and error-prone [5], while subordinating the accuracy the simulation model to that of the process model that is used as a starting point. Yet, process models produced by domain experts generally do not capture all possible process execution paths.

Several Process Mining (PM) approaches to construct BPS models from business process event logs have been proposed [7, 4]. These approaches generally assume that the event log perfectly fits the process model given as input, which is not generally the case. Moreover, the objective of these approaches has been to demonstrate the feasibility of extracting simulation parameters from event logs, and not to fully automate the extraction of BPS models from event logs.

In this paper we present a tool, namely Simod, to automatically generate simulation models from event logs. Simod assembles various PM techniques to discover all the components of a BPS model. The tool also implements pre-processing techniques to ensure conformance between the event logs and the discovered models, and a hyperparameter optimization technique to fine-tune the accuracy of the resulting model.

2 Maturity

This paper presents the first version of Simod, which automatically creates simulation models that can be executed by the BIMP [2] simulator. We used three event logs (one synthetic and two real-life)³ to validate the process performance measures computed by the automatically generated simulation models against the actual performance measures observed in the original logs. The synthetic log corresponds to a purchase-to-pay process. The real event logs come from an Academic Credentials Recognition process and from a Manufacturing Production process. Figure 1 presents the basic architecture and the steps for the creation of a process simulation model.

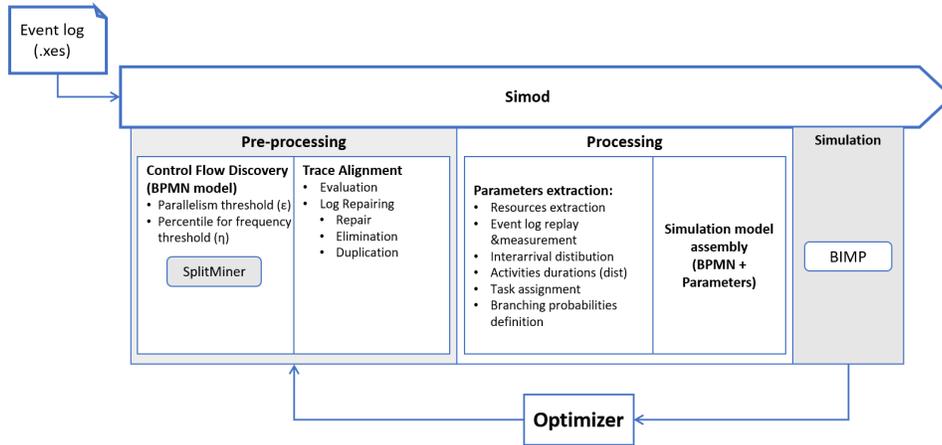


Fig. 1. Architecture of Simod

Pre-processing. This stage extracts a BPMN model of the business process from data, and guarantees its quality and coherence with the event-log.

In the *Control Flow Discovery* step a BPMN model is mined using solely an event-log in XES format as input. The mined model constitutes the basis of the simulation model and describes activities, decision gateways and the way they are related in the process. Simod integrates with the SplitMiner algorithm [1], which allows extracting different structures by varying the parameters ϵ and η . The η parameter is the percentile for frequency threshold and acts as filter over the incoming and outgoing edges. The ϵ parameter is the parallelism threshold and determines the quantity of concurrent relations between events to be captured. This feature enables the exploration of multiple BPMN structures to find the

³ Logs available at <https://github.com/AdaptiveBProcess/Simod/tree/master/inputs>

most suitable in relation with data, something almost impossible to achieve in the traditional way of create simulation models.

In the *Trace alignment* step Simod measures the conformance between the process model and the event log, and provides the option of managing the non-conformances. Non-conformances are often caused by the search for balance between precision and simplicity of process mining techniques which relies on models that do not fit with the 100% of the event log traces. Simod replays the event log over the generated BPMN structure filtering the non-conformant traces. Simod also handles these non-conformant traces through its removal, replacement or repairing. Removal consists of deleting the non-conformant traces from the event log, using only those that can be reproduced for later analysis. Replacement consists in the change the non-conformant traces by the conformant most similar ones. The Similarity between traces is determined using the Damerau-Levinstein edit distance algorithm. Repair consists in making changes to the event log so that every trace can be replayed by the BPMN model (which is necessary to compute the branching probabilities of the decision gateways). For this repairing phase, Simod uses a conformance checking tool [6] that efficiently computes optimal alignments between each trace in the log and the closest corresponding trace produced by the process model.

Processing. In this stage Simod extracts the simulation parameters and assembles them with the process structure to create a BPS model. The extracted set of parameters was chosen according to the most common ones required by the existing commercial simulators. In this tool version, the simulation model complies with the parameters required by the BIMP simulator.

In the *Parameters Extraction* step all the simulation model parameters are calculated. The resource pools involved in the process, which are extracted using the algorithm of Song and Van der Aalst [8], are assigned to the different activities according to the frequency of execution. Likewise, the definition of probability distributions (PDF) of inter-arrival times and activities durations is carried out by fitting a collection of possible distribution functions to the data series and by selecting the PDF that yields the minimum standard error with respect to the data series. Finally, the branching probabilities definition is calculated from the frequencies of traversal of the conditional branches computed during the process replay.

The *Simulation Model Assembly* step is then performed to merge the simulation parameters and the BPMN model into a single data structure. Simod creates simulation models for the BIMP simulator, which receives the same BPMN structure with an additional XML data about all the simulation parameters.

Post-processing. In this phase Simod measures the similarity of the generated simulation model in relation to the original event log. Simod also explores different pre-processing options to find the most optimal combination.

The simulation model *similarity assessment* is carried out using the Demerau-Levinstein (DL) algorithm with a modification which includes a time penalty.

The DL algorithm measures the distance between sequences in terms of the number of editions necessary for one string character to be equal to another. This basic version of the algorithm penalizes each time actions such as insertion, deletion, substitution, and transposition are carried out. Thanks to the sequential nature of event logs this version of the algorithm is commonly used to measure the distance between task sequences. In Simod we add also a penalty in case of found differences in time between two activities of a trace allowing us to measure not only the discrete variables, but also the continue ones. Simod executes the simulation model and evaluates the similarity between the results of the generated model and the real event log. This information is useful for the user, since it allows him to decide whether or not to use the model for future analysis.

The accuracy of the generated simulation model depends to a large extent on the accuracy of the business process model used as a starting point, and this, in turn, depends on the parameters used in the pre-processing phase. Simod provides the option to use a Bayesian hyperparameter optimizer to efficiently explore the search space composed of all possible combinations of pre-processing options. In this way, Simod can discover a suitable combination of parameters without requiring the user to manually test a large number of possible models.

3 Simod Interface

Simod was developed in Python 3.6 and offers a user interface for Jupyter Notebooks. The user interface allows to select an event-log in XES or MXML format and to decide how to generate and analyze the model (see Figure 2). Simod requires the event log to include start and complete time-stamps otherwise is impossible to determine the activities duration. From the interface it is possible to define the pre-processing parameters manually, or using the hyper-parameter optimizer. In both cases, Simod provides information on the execution of the discovery steps, and on the results obtained from the similarity evaluation.

4 Screencast and Links

A screencast is available at <https://youtu.be/i9X5jwjuipk>. This video illustrates two typical scenarios. In the first scenario the user explores manually the different pre-processing options of the tool to generate a simulation model. In the second scenario the user defines a search space and the tool automatically explore the combination looking for the optimal one.

The source code, installation and usage tutorial, and example event logs can be downloaded from <https://github.com/AdaptiveBProcess/Simod.git>.

5 Conclusion

An automated construction of business process simulation models allows the efficient exploration of exceptional paths unknown by the process experts. The

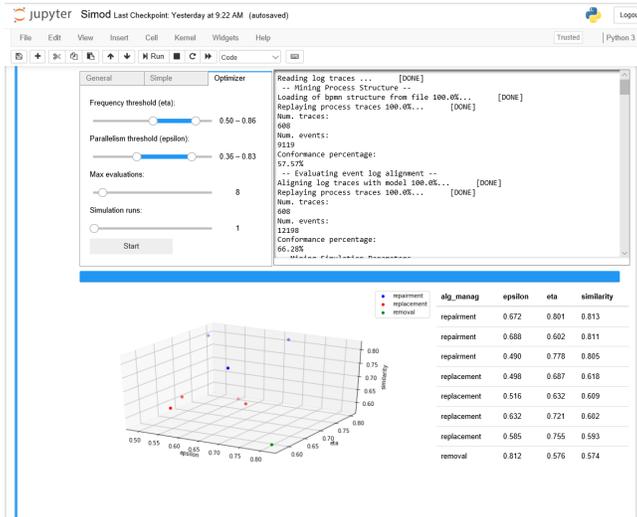


Fig. 2. Simod interface

tool presented in this paper enables this capability using as an only entry an event log. Future work may include the use of multiple mining algorithms for the extraction of simulation parameters.

Acknowledgments This research is funded by the Estonian Research Council (IUT20-55) and the European Research Council (Project PIX).

References

1. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L.: Split miner: Discovering accurate and simple business process models from event logs. In: IEEE Data Mining. pp. 1–10. IEEE, New Orleans, LA, USA (2017)
2. BIMP: BIMP - The Business Process Simulator (2016), <http://bimp.cs.ut.ee/>
3. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, second edition edn. (2018)
4. Martin, N., Depaire, B., Caris, A.: The use of process mining in business process simulation model construction. *Bus. Inf. Syst. Eng.* **58**(1), 73–87 (2016)
5. Marușter, L., van Beest, N.R.T.P.: Redesigning business processes: a methodology based on simulation and process mining techniques. *Knowl. Inf. Syst.* **21**(3), 267–297 (2009)
6. Reißner, D., Conforti, R., Dumas, M., Rosa, M.L., Armas-Cervantes, A.: Scalable conformance checking of business processes. In: Proc. of the OTM 2017 Conferences. pp. 607–627. Springer (2017)
7. Rozinat, A., Mans, R., Song, M., van der Aalst, W.: Discovering simulation models. *Inform. Syst.* **34**(3), 305 – 327 (2009), doi:10.1016/j.is.2008.09.002
8. Song, M., van der Aalst, W.M.: Towards comprehensive support for organizational mining. *Decis. Support. Syst.* **46**(1), 300 – 317 (2008)