GenCPN: Automatic CPN Model Generation of Processes (Extended Abstract)

Mahsa Pouarbafrani*, Shubham Balyan†, Mehmood Ahmed†, Sonam Chugh†, Wil M.P. van der Aalst*
*Process and Data Science Group, RWTH Aachen,{mahsa.bafrani , wvdaalst}@pads.rwth-aachen.de
†RWTH Aachen Univeristy,{shubham.balyan, mehmood.ahmed, sonam.chugh}@rwth-aachen.de

***Abstract*—To model business processes, a variety of modeling tools and techniques are used. Colored Petri Nets (CPN) are one of the modeling notations capable of presenting different patterns in activity flows, e.g., concurrency of activities in the process. CPN Tools enables the creation and execution of business process simulation models based on CPN notation and *Standard Machine Language* (SML) w.r.t. different aspects of processes. For instance, assigning different resources to different tasks, generating attributes for cases, and modeling stochastic Petri nets for choices are possible. We can replay and capture process behavior in the form of event logs for what-if analyses using simulation models of business processes. However, modeling the processes and implementing SML functions for generating event logs are complicated. We designed and implemented a tool that extracts process parameters directly from an event log and converts the event log using process mining techniques to CPN models along with generated SML functions. The tool allows for the straightforward modeling of business processes without user interaction, as well as the ability to change via a user interface.**

I. Introduction

Process mining insights using event data of processes are put into action by designing simulation models at detailed [1] and aggregated levels [2], [3] for future analyses. Process simulation models are crucial for replaying and regenerating process behavior in different scenarios. CPN Tools is a powerful tool for designing and simulating CPN models. It is based on two main components: (1) a graphical editor in which users are possible to design their models interactively, and (2) a simulation engine as shown in Figure 1. The engine is responsible for simulating the designed model and checking for the syntactical errors [4].

The power of CPN Tools in providing such an extensive simulation model for business processes was the motivation of various tools and techniques to generate simulation models of processes [5]. In [6], an SML framework is provided to access CPN models and enables the extension and capabilities of the tool. The provided platform for extending CPN Tools capabilities is used for a business process to represent declarative business processes. Authors in [1] exploit process mining techniques and generate CPN models which are executable via CPN Tools. The tool is a ProM plugin and covers all the aspects of process mining as one of the first automatic simulation model generation based on historical data of business processes. However, the tool is dependent on the ProM platform and requires multiple user interactions. Moreover, the option for generating the event logs of the simulation results directly from CPN Tools is not considered. Other techniques propose the generation of process simulation models without the use of CPN Tools and in the form of independent platforms, such as [7], [8]. However, they miss the flexibility of the CPN Tools for considering different aspects of a process. Our goal is to design a tool that addresses two main purposes: (1) generating CPN models from event logs based on real-world parameters for what-if analysis, and (2) more importantly, capturing execution results in the form of event logs for various purposes, e.g., research or process mining, without designing and programming efforts.

In this paper, we introduce a new tool that is able to generate a ready to simulate CPN models directly from an event log. Our tool (GenCPN) is a web application based on Python, which reduces all the user interaction and efforts for designing the models from scratch. Moreover, the process mining insights are put into action to extract all the required process simulation parameters. The strong contribution of this tool is providing the implemented SML functions that directly include all the interactions of the simulation engine (daemon) for the CPN models. It also provides the ready-to-run CPN models in CPN Tools.

II. GenCPN: CPN Model Generator

To design the corresponding CPN model of a given process (in the form of an event log), the results of process mining techniques are transformed into the XML notation of CPN models. The substantial information in this step is the discovered process model, i.e., Petri net, probability of choices in the process model, resource pooling and their capacity, the performance information, and the arrival rate for generating new tokens. As shown in Figure 2, the process model (Petri net) is enriched with the gained knowledge from process mining. The implementation of process discovery is based on [9] which is extended w.r.t. our framework. In the next step, the simulation parameters required for generating CPN models, i.e., *.CPN* files executable by the CPN Tools simulator, and the corresponding SML functions and files for automatic generation of event logs are generated. SML functions are also provided for delay of arrival rate, business hours, working days, and functions for distribution of activities' duration. In addition to the tool's user interface, user interaction is possible via CPN Tools' graphical editor.

A sample result of the generated CPN model for an event log is shown in Figure 3. For generating new tokens (cases), the delay function is designed, which is automatically filled

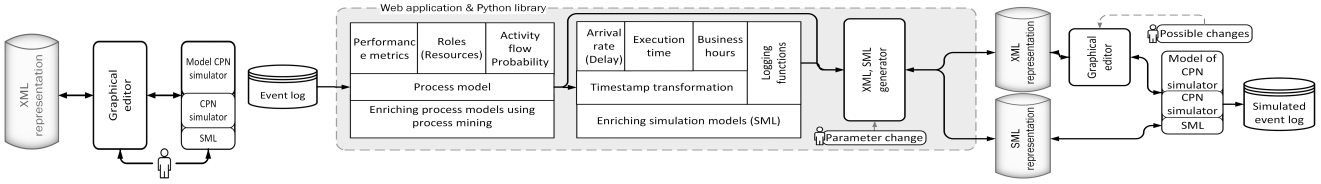

Fig. 1. Architecture of CPN Tools.

Fig. 2. The general framework of the tool for generating a CPN simulation model using process mining techniques from an event log. The results will be executable model in the CPN Tools structure.

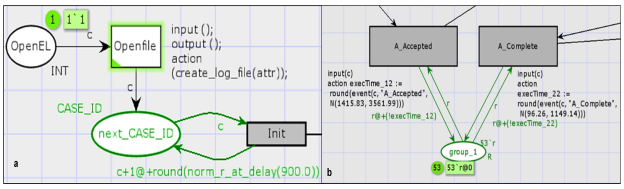

Fig. 3. The generated CPN model for a real event log. The function for generating new tokens (cases) based on the discovered arrival rate and capturing the new cases in the event log (a). Resource pooling between two activities (b).

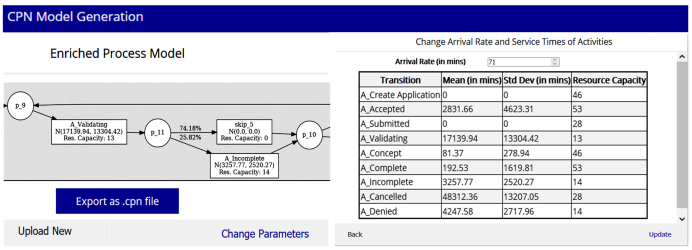

Fig. 4. The screenshot of the GenCPN including the enriched discovered CPN and the option to change the parameters.

with the extracted arrival rate from the event log. For logging generated events in the form of an event log, the function *create log file* in the *Openfile* transition is designed (Figure 3 (a)). Figure 3 (b) represents the generated function for executing each activity based on their extracted duration from the event log, as well as a sample scenario of sharing resources (resource pool) between two activities.

III. Tool Maturity

CPN Tools is being used for both academic and industrial purposes. GenCPN which generates CPN models automatically based on the event logs of processes is used for generating event logs of different projects such as IOP project[1] to regenerate event logs of production lines in a different setting. Moreover, different simulation models are designed in the previous and current researches, e.g., [7].

GenCPN is publicly available in the form of web application[2] along with all the codes and the screencast as open source[3] for further uses. Figure 4 is part of the tool's user interface, where the enriched discovered CPN model from the event log is shown (left). The discovered parameters are also shown (right) and possible to be changed by the user if required, e.g., arrival rate or an activity service time. The output of the tool is a zip file including the CPN file and the corresponding SML file, which is ready to be stimulated via the graphical interface of the CPN Tools.

IV. Conclusion

Designing executable and close-to-reality prescription models is a challenging task. Process mining is used to generate and design these prescribing models as close to reality as possible. We presented a tool in this paper that automatically converts event logs into executable simulation models. The simulation model parameters are filled in with real values from previous process executions. The model is created as an enriched CPN model, which can be run using CPN Tools, a sophisticated tool for simulating Petri nets. The required SML functions are also generated together with the model in order to capture simulation results in the form of event logs and to allow the user to make additional changes while running the simulation, e.g., adjusting the working hours or the density of arrival rate for different months.

1 www.iop.rwth-aachen.de

2 https://cpn-model-process-discovery-1.herokuapp.com/generate-cpn-model/

3 https://github.com/mbafrani/AutomaticCPNModelGenerator.git

Acknowledgment

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2023 Internet of Production-Project ID: 390621612. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

[1] A. Rozinat, R. S. Mans, M. Song, and W. M. P. van der Aalst, "Discovering colored Petri nets from event logs," *Int. J. Softw. Tools Technol. Transf.*, vol. 10, no. 1, pp. 57–74, 2008.

[2] M. Pourbafrani and W. M. P. van der Aalst, "Extracting process features from event logs to learn coarse-grained simulation models," in *CAiSE 2021, Proceedings*, vol. 12751. Springer, 2021, pp. 125–140. [Online]. Available: https://doi.org/10.1007/978-3-030-79382-1_8

[3] M. Pourbafrani and W. M. P. van der Aalst, "PMSD: data-driven simulation using system dynamics and process mining," in *Proceedings of the Demonstration(BPM)*, vol. 2673, 2020, pp. 77–81. [Online]. Available: http://ceur-ws.org/Vol-2673/paperDR03.pdf

[4] A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen, and K. Jensen, "CPN Tools for editing, simulating, and analysing coloured Petri nets," in *24th International Conference, ICATPN 2003*, 2003, pp. 450–462.

[5] M. Jansen-Vullers and M. Netjes, "Business process simulation–a tool survey," in *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, vol. 38, 2006.

[6] M. Westergaard and L. M. Kristensen, "The Access/CPN framework: a tool for interacting with the CPN Tools simulator," in *Applications and Theory of Petri Nets*, 2009, pp. 313–322.

[7] M. Pourbafrani, S. Jiao, and W. M. P. van der Aalst, "SIMPT: process improvement using interactive simulation of time-aware process trees," in , *RCIS 2021, Proceedings*, vol. 415, 2021, pp. 588–594. [Online]. Available: https://doi.org/10.1007/978-3-030-75018-3_40

[8] M. Pourbafrani, S. Vasudevan, F. Zafar, Y. Xingran, R. Singh, and W. M. P. van der Aalst, "A python extension to simulate Petri nets in process mining," *CoRR*, vol. abs/2102.08774, 2021. [Online]. Available: https://arxiv.org/abs/2102.08774

[9] A. Berti, S. J. van Zelst, and W. M. P. van der Aalst, "Process mining for python (pm4py): bridging the gap between process- and data science," *CoRR*, vol. abs/1905.06169, 2019.