

# An Agent-Based Workflow Simulator

Eric, Amann<sup>1</sup>, Patrick, Delfmann<sup>1</sup>

<sup>1</sup>University of Koblenz, Universitätsstr. 1, 56072 Koblenz, Germany

## Abstract

Event logs form the basis for many process mining activities. However, oftentimes appropriate event logs are not available, for researchers and practitioners. Synthetic log generation can provide a suitable alternative in some cases. In this paper we present a new approach to process log generation by presenting a tool which allows the creation of agent-based simulation environments in which processes can be executed. Our approach relies on the interaction of simulated agents with the Camunda workflow engine. Simulations can be built with a large number of configuration properties and are enhanced through the integration of context data. This allows users to simulate a vast number of different scenarios.

## Keywords

Workflow Simulation, Process Simulation, Agent-Based Simulation, Camunda

## 1. Introduction

The field of process mining (PM) is an active and growing research domain. At the center of all PM activities is the event log, containing all available information on the execution of a selected process [1]. However, obtaining suitable event logs can be a challenge for researchers, as companies are often not willing to share their data. But also in practice insufficient event log data or missing digitization of processes can be obstacles to PM adoption [2]. This problem can be addressed through the use of synthetic log generators. Several mostly petri-net-based tools and frameworks have been proposed [3, 4, 5, 6]. Other approaches use the Business Process Model and Notation (BPMN) specification as the basis for the log generation. [7, 8, 9, 10, 11].


Our approach to process simulation relies on an agent-based simulation (ABS) of the environment the process is run in. Instead of propagating tokens directly through the process model, we simulate agents that interact with the process through a workflow management system. Agents have their own predefined behavior and search for, claim and complete process tasks correspondingly. Simulations can be further enhanced by including context data, which can be configured to influence many of the simulation parameters, thus allowing for more complexity and realism. Our goal is to enable high quality workflow simulations and to support a large amount of different application scenarios for the tool.


---

*Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 22nd International Conference on Business Process Management (BPM 2024), Krakow, Poland, September 1st to 6th, 2024.*

✉ amann@uni-koblenz.de (E. Amann); delfmann@uni-koblenz.de (P. Delfmann)

ORCID 0009-0005-6241-6081 (E. Amann); 0000-0003-4441-0311 (P. Delfmann)

 © 2024 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)

## 2. Tool Description

The proposed simulator is a java-based application that uses the MaDKit engine for the underlying simulation [12]. MaDKit was chosen as the simulation engine because of its agent-group-role model that coordinates the agents' behavior. This model takes an organizational approach in which the interaction of agents between each other is directed by the groups and roles they are assigned to. Because processes are usually part of organizations with such groups and roles, the configuration real life scenarios are well supported by the simulation engine.

The simulation of the process execution is orchestrated through the interaction with the Camunda workflow engine. Simulated agents interact with the workflow engine like real humans would do. They try to claim tasks and complete them once they are done with them. These interactions are facilitated through Camunda. Thus, the event log is created by the interaction with the workflow system, just as it would be in a real-world setting. Any characteristic that can be observed in the event log is there as a result of the agents' interaction with their simulated environment and the workflow engine. The tool<sup>1</sup> and a accompanying demonstration video<sup>2</sup> are publicly available.

### 2.1. Configuration

Simulations are defined through a configuration file. This configuration is provided in the YAML<sup>3</sup> format. Depending on the goal the configuration can either be very detailed or undetailed in which case default values are used to allow for the fast creation of a simulation. There are seven main configuration elements:

- *Simulation Configuration* - One configuration file can contain several simulation configurations. If deployed, each simulations will be executed simultaneously with the others. In this configuration element base attributes such as the start and end dates and the time increment per simulation tick are defined.
- *Workers* - Workers are agents that claim and complete tasks in the configured processes. Several worker configurations can be provided. Each worker configuration spawns a defined amount of identical agents. They can be assigned a work schedule and specific tasks to process.
- *Processes* - Process configurations include the BPMN file, which is automatically deployed to the connected Camunda instance. Just like workers they can be assigned a schedule which controls the times at which the process is "active" and creates new process instances. Also, the interval at which process instances are started can be set.
- *Tasks* - For each task in the deployed processes the processing time can be set. Tasks are referenced through the ID specified in the BPMN file.
- *Process Variables* - Any task completion by a worker can add variables to the process. These can, for example, be used to direct the sequence flow after an exclusive gateway. The supported data types include numeric, boolean, and string. Variables are either

---

<sup>1</sup><https://uni-ko.de/workflow-simulator>

<sup>2</sup><https://uni-ko.de/workflow-simulator-demo-video>

<sup>3</sup><https://yaml.org/spec/>

randomly generated, or taken from context data. For the random value generation of numeric values options are available to define the range and distribution.

- *Schedules* - Schedules can be configured for the assignment to workers and processes. Each schedule includes configurations for the days of the week and months. For each day of the week the start time and an active duration can be defined. For each month inactive days can be configured.
- *Context* - Context data is provided through a CSV file. The file must include a timestamp for each entry and can be applied to workers, processes, tasks, process variables and schedules to modify configured values such as a workers performance or the interval at which new process instances are started. An example for the application of context data could be the number of orders in an online shop that affects the number of started processes, or health data, which influences the workers' schedules to simulate an epidemic among the them.

A detailed description of all configuration values, as well as an example configuration can be found in the project's Git repository.

## 2.2. Execution

Because only the environment surrounding the process is simulated, with the process execution being supported through an external workflow management system, several separate software components are involved in the execution of the simulation. Figure 1 provides an overview over the components and their interaction between each other. A full run of a simulation can be separated into three distinct parts.



**Figure 1:** Interactions of the simulator components with each other.

First, the configuration file must be created. Many configuration properties are assigned a default value if left empty by the user to support the fast creation of new scenarios. Once a suitable configuration is created, it can be deployed and executed.

The actual running of the simulation is the second part. The processes and agents are created as defined in the configuration and interact with their environment. During each simulation step all agents perform the activity they are programmed to do. This can be, for example, the processing of an assigned task, the search for a new task to perform, or relaxing outside of their

working hours. The simulation ends either by reaching the end date or a predefined number of simulated process instances.

Because the simulated time is much faster than the real time, the timestamps saved in the Camunda event log are not conforming to the ones in the simulation. Therefore, in the last step of the simulation, all timestamps pertaining to the simulated processes are modified. This is done through the direct interaction with the Camunda database, as there are no API endpoints for these operations. For this reason, the simulation cannot be used with SaaS Camunda instances, but must be self-hosted.

### **3. Application Scenarios**

There are several main application scenarios in which our tool could be used. An obvious one is the generation of event logs for the testing of predictive process monitoring (PPM) applications. Users can generate large logs for the training of models. As opposed to other synthetic log generators our tool does not provide logs with specific, predefined characteristics, but ones that conform to a predefined environment. Because the simulated agents mimic human behavior all log features arise organically based on the simulation configuration.

The second usage scenario can be the verification and testing of process models. Once a simulation scenario is configured a process model can be simulated. By comparing different versions of a process model or altering the scenario configuration between simulations, insights on the expected real-world performance of the process can be generated. This can help to find issues in the process execution or inefficiencies beforehand. Also, processes can be tested against outside influences by using different context data sets.

Lastly, the tool could be used to pre-train a PPM model in cases where a process's event log is too small for the application of PM. For example, if a new process is introduced, it can take weeks or even months depending on the circumstances until a sufficient number of instances have been completed. By simulating the process beforehand, an arbitrary number of instances can already be executed and thus enable the use of PPM tools at the moment the process is introduced. However, this approach requires a deep understanding of the environment the process is deployed in and all outside influences.

### **4. Conclusion and Outlook**

Our tool presents a way to directly simulate processes in the Camunda workflow engine. The agent-based simulation approach provides a new approach to the generation of realistic event logs. There are several usage scenarios for the tool, even outside of the field of PPM. Still, more testing needs to be done, especially with regard to the event log realism. We aim to evaluate the tool through the comparison with real world logs and against conventional BPMN simulators.

Also, there are more features which we want to integrate in the future development. First and foremost, the ability to use the simulator with any BPMN process. At the moment processes are limited to using elements which are not affected by the processing time such as timer events. These elements can currently distort the simulation because of the mismatch between the simulated and the real time. We plan on addressing this issue in future releases. Additionally,

we plan on providing a user interface for the creation, monitoring and analysis of simulations to improve the tool's usability and accessibility to users without programming knowledge.

## Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 456415646, 432399058.

## References

- [1] W. M. van der Aalst, Process mining: a 360 degree overview, in: *Process Mining Handbook*, Springer, 2022, pp. 3–34.
- [2] T. Grisold, J. Mendling, M. Otto, J. vom Brocke, Adoption, use and management of process mining in practice, *Business Process Management Journal* 27 (2021) 369–387.
- [3] A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen, K. Jensen, Cpn tools for editing, simulating, and analysing coloured petri nets, in: *International conference on application and theory of petri nets*, Springer, 2003, pp. 450–462.
- [4] O. Kummer, F. Wienberg, M. Duvigneau, J. Schumacher, M. Köhler, D. Moldt, H. Rölke, R. Valk, An extensible editor and simulation engine for petri nets: Renew, in: *Applications and Theory of Petri Nets 2004: 25th International Conference, ICATPN 2004*, Bologna, Italy, June 21–25, 2004. *Proceedings* 25, Springer, 2004, pp. 484–493.
- [5] J. Nakatumba, M. Westergaard, W. M. van der Aalst, Generating event logs with workload-dependent speeds from simulation models, in: *International Conference on Advanced Information Systems Engineering*, Springer, 2012, pp. 383–397.
- [6] J. Grüger, T. Geyer, D. Jilg, R. Bergmann, Sample: A semantic approach for multi-perspective event log generation, in: *International Conference on Process Mining*, Springer, 2022, pp. 328–340.
- [7] A. P. Freitas, J. L. M. Pereira, Process simulation support in bpm tools: The case of bpmn, in: *International Conference on Business Sustainability. 2100 Projects*, 2015.
- [8] J. L. Pereira, A. P. Freitas, Simulation of bpmn process models: current bpm tools capabilities, in: *New advances in information systems and technologies*, Springer, 2016, pp. 557–566.
- [9] A. A. Mitsyuk, I. S. Shugurov, A. A. Kalenkova, W. M. van der Aalst, Generating event logs for high-level process models, *Simulation Modelling Practice and Theory* 74 (2017) 1–16.
- [10] L. Pufahl, T. Y. Wong, M. Weske, Design of an extensible bpmn process simulator, in: *Business Process Management Workshops: BPM 2017 International Workshops*, Barcelona, Spain, September 10-11, 2017, *Revised Papers* 15, Springer, 2018, pp. 782–795.
- [11] P. Bocciarelli, A. D'Ambrogio, A. Giglio, E. Paglia, Bpmn-based business process modeling and simulation, in: *2019 Winter Simulation Conference (WSC)*, IEEE, 2019, pp. 1439–1453.
- [12] O. Gutknecht, J. Ferber, The madkit agent platform architecture, in: *Workshop on infrastructure for scalable multi-agent systems at the international conference on autonomous agents*, Springer, 2000, pp. 48–55.