

ProSiT: A Tool for Interactive and Transparent Process Simulations

Francesco Vinci¹, Gyunam Park^{2,3}, Wil M. P. van der Aalst^{2,4}, and
Massimiliano de Leoni¹

¹ University of Padua, Padua, Italy

`francesco.vinci.1@phd.unipd.it`, `deleoni@math.unipd.it`

² Fraunhofer Institute for Applied Information Technology (FIT), Sankt Augustin,
Germany

³ Eindhoven University of Technology, Eindhoven, The Netherlands

`g.park@tue.nl`

⁴ Process and Data Science, RWTH Aachen University, Aachen, Germany
`wvdaalst@pads.rwth-aachen.de`

Abstract. Business Process Simulation (BPS) provides a controlled environment to evaluate alternative process designs and perform “what-if” analyses. While several approaches exist for deriving simulation models directly from event logs, existing solutions are often limited in accessibility, configurability, and interpretability. We present ProSiT: *Process Simulation Tool*, a user-friendly environment for data-driven simulation. ProSiT supports (i) the ingestion of event logs and process models, (ii) automated discovery of control-flow, timing, resource, and trace attribute parameters, (iii) interactive configuration of alternative service scenarios, (iv) visualization of accuracy metrics, and (v) generation and statistical analysis of simulated event logs. ProSiT integrates state-of-the-art machine learning methods for parameter discovery in an explainable white-box form, contrasting with existing deep learning-based approaches that act as black boxes. By combining simulation accuracy with transparency, ProSiT empowers service architects and process analysts to experiment with alternative designs, evaluate service-level impacts, and explore optimization opportunities through an intuitive graphical interface.

Keywords: Business Process Simulation · Service Composition · Explainable Machine Learning · Graphical User Interface

1 Introduction

Business Process Simulation (BPS) has become a crucial instrument for organizations seeking to understand and improve their operational efficiency. As a foundational discipline in process management, BPS enables the risk-free evaluation of process changes without disrupting the real world [1]. These digital process twins allow organizations to conduct “what-if” analyses and quantitatively assess the impact of potential changes on key performance indicators such as cycle time, resource utilization, and throughput.

However, the effectiveness of BPS tools hinges on their ability to accurately capture complex process dynamics, while remaining accessible and interpretable to process analysts. Early approaches and many commercial tools still rely on manual model construction, which is time-consuming and error-prone [1]. More recently, several research tools have emerged to automate simulation model discovery from event logs [2,3,4,5].

While numerous research efforts have focused on data-driven BPS, many still face challenges related to the accessibility, configurability, and interpretability of their underlying models. These tools highlight a recurring trade-off: while data-driven methods can improve accuracy, they often sacrifice interpretability. Deep learning models, in particular, act as black boxes, making it difficult for users to understand or modify the underlying decision logic. This limits their practical utility in “what-if” analysis, where service architects must test and reason about alternative scenarios.

To address this gap, we present ProSiT (*Process Simulation Tool*), a web-based environment that integrates explainable, white-box machine learning techniques for simulation parameter discovery [7]. ProSiT combines simulation accuracy with full transparency, allowing users to inspect, validate, and interactively modify discovered parameters. By offering an end-to-end workflow—from log ingestion and model discovery to scenario configuration and results visualization—ProSiT enables process analysts to conduct reliable and interpretable “what-if” analyses, bridging the gap between data-driven automation and human-centric process design.

2 Tool Description

This Section describes the architecture, core functionalities, and API design of ProSiT. The modular architecture of the tool (Section 2.1) enables seamless integration of discovery and simulation components, while its workflow (Section 2.3) guides users from log ingestion to scenario analysis. The REST API (Section 2.2) provides interfaces for both interactive frontend communication and backend system integration.

2.1 System Architecture

The ProSiT tool follows a client-server architecture implemented as a web application. The overall architecture is depicted in Figure 1. The backend is built using Python with the Flask framework, handling all computational operations, including process discovery, parameter learning, and simulation execution.⁵ The frontend is implemented with standard web technologies (HTML, CSS, JavaScript) providing a responsive user interface. Communication between frontend and backend occurs through a RESTful API. Data persistence is managed via SQLite database. The core process mining and simulation capabilities

⁵ <https://flask.palletsprojects.com/>

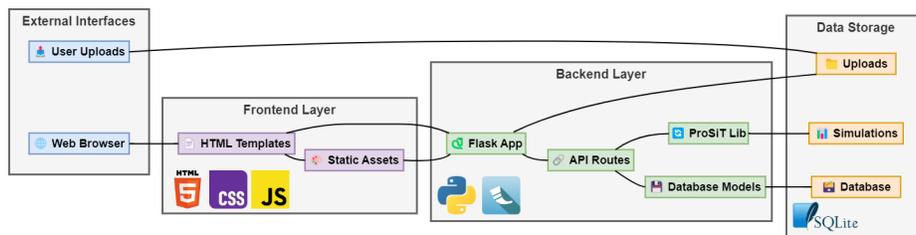


Fig. 1. High-level architecture of the ProSiT tool, showing the separation between the frontend layer (HTML/CSS/JS), backend layer (Python/Flask), and data storage (SQLite with SQLAlchemy ORM). The ProSiT library encapsulates the core discovery and simulation logic.

Endpoint	Method	Description
/api/upload	POST	Upload an event log or PNML file.
/api/discover	POST	Initiates the automated discovery of simulation parameters from an uploaded event log.
/api/parameters	GET	Retrieve discovered parameters.
/api/parameters	PUT	Updates specific simulation parameters to configure custom scenarios for “what-if” analysis.
/api/simulate	POST	Run a simulation with configured parameters and returns performance metrics and a synthetic event log.

Table 1. Core REST API endpoints of the ProSiT tool, supporting the complete simulation lifecycle from data ingestion to results generation.

are encapsulated in the ProSiT library, which implements the white-box discovery algorithms [7] and discrete-event simulation engine.⁶

For easy deployment and reproducibility, the tool is distributed as a Docker container, ensuring consistent execution across different environments.⁷

2.2 API Interface

The backend functionalities of ProSiT are programmatically accessible through a comprehensive RESTful API, enabling both interactive use via the web interface and integration with external systems. The API follows REST principles and returns JSON responses for all endpoints. Table 1 summarizes the core endpoints that support the complete simulation workflow.

The API design follows a sequential workflow: users first upload process data, then discover parameters, optionally modify them, and finally run simulations. This structured approach ensures reliable state management and supports reproducible “what-if” analyses.

⁶ <https://github.com/franvinci/prosit>

⁷ <https://www.docker.com/>

2.3 Core Functionalities

The ProSiT tool supports an end-to-end workflow for data-driven process simulation, guiding users through five key phases:

1. **Data Ingestion:** Users can upload event logs in XES format, and optionally provide process models in PNML format.
2. **Parameter Discovery:** This core functionality uses explainable machine learning models to automatically extract simulation parameters from event logs [7]. Users can tailor the discovery process by configuring the Inductive Miner algorithm with a specific noise threshold, opting for probabilistic decision trees with controllable depth, and activating incremental discovery with customizable grace periods [6]. The discovery covers *transition weights* for control-flow decisions; *resource* parameters including assignment probabilities, multitasking capabilities and availability calendars; *arrival time* distributions; *execution time* distributions per activity; *waiting time* distributions per resource; and distributions for *trace attributes*.
3. **Interactive Scenario Configuration:** An intuitive graphical interface allows analysts to modify the discovered parameters, facilitating the creation and comparison of alternative scenarios for comprehensive “what-if” analysis.
4. **Accuracy Assessment:** The tool evaluates the quality of the simulation by comparing the generated logs with the original event data. Assessment includes control-flow similarity measures (n-gram distance), temporal accuracy through distributional metrics, realism of resource handover behavior, and generalization capability through entropy analysis.
5. **Simulation Execution:** Users initiate simulations by defining the desired number of traces and a starting timestamp. The execution outputs a downloadable simulated event log, accompanied by visual analytics including process maps with performance data, cycle time distributions, resource utilization heatmaps, and detailed activity execution and waiting time analyses.

This structured workflow enables users to systematically advance from event data to validated simulation insights, with full transparency over simulation parameters and quantitative accuracy assessment.

3 Resources and Demonstration

ProSiT is distributed as a Docker container for easy deployment, ensuring consistent execution across different platforms. The Docker image includes all dependencies, including the ProSiT discovery engine. Developers can extend the tool by modifying the source code or integrating additional discovery algorithms through the modular API.

A demonstration video showcasing ProSiT capabilities is online available on <https://youtu.be/ZN7ajI9-2NI>. The video walks through a complete case study using a publicly available event log from the BPI Challenge 2017, demonstrating the entire workflow from log upload through parameter discovery, scenario configuration, simulation execution, and results analysis.

The complete source code, installation instructions, and documentation are available in <https://github.com/franvinci/prosit-tool>. The repository includes several publicly available event logs that users can directly employ for testing and evaluation purposes.

4 Tool Maturity and Conclusion

This paper introduces ProSiT, an interactive and explainable tool for data-driven business process simulation. By integrating state-of-the-art machine learning in a white-box form and providing a user-friendly graphical interface, ProSiT addresses a critical gap in existing solutions. Its focus on transparency and configurability allows process analysts and service architects not only to execute simulations, but also to explore, and configure the underlying parameters.

Built on probabilistic decision trees for explainable parameter discovery, ProSiT stands as a mature, fully functional research artifact suitable for both academic and industrial contexts. It demonstrates how white-box models can deliver predictive accuracy while remaining interpretable, thus supporting meaningful “what-if” analyses for service optimization.

As future work, we plan to conduct user studies to evaluate the tool’s usability and effectiveness in real-world process improvement scenarios. These evaluations will assess how well analysts can understand, trust, and practically apply the white-box simulation models for decision support.

References

1. van der Aalst, W.M.P.: Business Process Simulation Survival Guide, pp. 337–370. International Handbooks on Information Systems, Springer (2015)
2. Chapela-Campa, D., López-Pintado, O., Suvorau, I., Dumas, M.: SIMOD: automated discovery of business process simulation models. *SoftwareX* **30**, 102157 (2025)
3. Edh, J., Falk, T., Kanon, F., Simonsson, E., Sjödin, H., Wincent, J., Barar, E., Blümel, R., Kirchorfer, L.: A business process simulation tool bridging control-flow and resource-centric paradigms. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2025. CEUR-WS.org (2025)
4. López-Pintado, O., Halenok, I., Dumas, M.: Prosimos: Discovering and simulating business processes with differentiated resources. In: Enterprise Design, Operations, and Computing. EDOC 2022 Workshops - IDAMS, SoEA4EE, TEAR, EDOC Forum, Demonstrations Track and Doctoral Consortium. Springer (2022)
5. Meneghello, F., Francescomarino, C.D., Ghidini, C.: Rims_tool: a hybrid simulator for business processes. In: Doctoral Consortium and Demo Track 2023 at the International Conference on Process Mining 2023. CEUR-WS.org (2023)
6. Vinci, F., Park, G., van der Aalst, W.M.P., de Leoni, M.: Online discovery of simulation models for evolving business processes. In: Business Process Management - 23rd International Conference, BPM 2025, Proceedings. Springer (2025)
7. Vinci, F., Park, G., van der Aalst, W.M.P., de Leoni, M.: Reliable and configurable process simulations via probabilistic white-box models. In: Service-Oriented Computing - 23rd International Conference, ICSOC 2025, Proceedings. Springer (2025)