SMP Workbench: An Interactive Tool for Semi-Markov Process Engineering in Digital Twins

Mikkel Schmidt Andersen*
Aarhus University
Aarhus, Denmark
msa@ece.au.dk

Oliver Thomas*
Aarhus University
Aarhus, Denmark
202107067@post.au.dk

Prasad Talasila
Aarhus University
Aarhus, Denmark
prasad.talasila@ece.au.dk

Cláudio Gomes
Aarhus University
Aarhus, Denmark
claudio.gomes@ece.au.dk

Peter Gorm Larsen
Aarhus University
Aarhus, Denmark
pgl@ece.au.dk

Abstract

Digital Twins (DTs) must be updated and kept secure throughout their life cycle to remain consistent with their physical twins. Prior work proposed Semi-Markov Processes (SMPs) for probabilistic update scheduling, where updates are initiated when the system is likely to be in an update-suitable state. However, creating representative SMPs is a difficult, manual, and time-consuming process that requires domain expertise and knowledge of stochastic modeling techniques. This paper presents *SMP Workbench*, a tool for generating, visualizing, and refining SMP models for DTs. The tool derives an initial SMP from event logs, presents the resulting model in an editable visual form, and allows domain experts to inspect and refine the states, transitions, transition probabilities, and holding-time distributions before the model is used for stochastic analysis. We demonstrate the tool on logs from a case study and show how it supports a workflow from operational logs to creation of SMP models.

CCS Concepts

• **Software and its engineering** → *Software maintenance tools.*

Keywords

Semi-Markov processes, Digital twins, Cyber-physical systems, Model engineering, Log-based modelling, Interactive model refinement, Model visualisation, Tool support

ACM Reference Format:
Mikkel Schmidt Andersen, Oliver Thomas, Prasad Talasila, Cláudio Gomes, and Peter Gorm Larsen. 2026. SMP Workbench: An Interactive Tool for Semi-Markov Process Engineering in Digital Twins. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion 2026), October 04–09, 2026, Málaga, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3837062.3839090

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License.
MODELS Companion 2026, Málaga, Spain
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2903-4/2026/10
https://doi.org/10.1145/3837062.3839090

1 Introduction

Digital Twins (DTs) must be maintained and updated throughout their life cycle to remain consistent with their Physical Twins (PTs) [5]. As a physical asset undergoes maintenance or receives upgrades, its DT may also need to be updated to preserve its operational relevance. However, deciding when such updates should be applied is often a non-trivial task, particularly in industrial settings, where update decisions are constrained by uptime requirements, coordination costs from workers, service-level agreements, and the risk of interrupting ongoing operations. This is particularly important for DT-enabled cyber-physical systems, where an update applied at an unsuitable time may disrupt the interaction between the DT and PT, leading to unobserved anomalies in the PT. Platforms such as Digital Twin as a Service automate DT updates using DevOps pipelines [12]. The development of DT update schedulers can help DT platforms manage the rollout of updates to DTs.

Studies have proposed probabilistic update scheduling to identify favorable update windows for DT-enabled systems. Andersen et al. [2] modeled the operational behavior of a concrete batching plant using Semi-Markov Processes (SMPs), where states are classified according to their suitability for an update. In that approach, states are categorized as *safe* or *unsafe* for applying updates. The objective was to initiate an update when the probability of the system being in an update-suitable state was high. SMPs are well suited to this problem because they can represent probabilistic state transitions and non-exponential holding-time distributions, allowing update scheduling to account for the temporal behavior in real systems.

Despite the potential of SMPs for update scheduling, their construction for industrial DTs remains a practical barrier. The current modeling process is largely manual and requires domain knowledge and familiarity with stochastic modeling. Therefore, a practitioner must first identify any relevant operational states, estimate the transition probabilities, and then determine the holding-time distributions for each observed transition. This makes the approach difficult to adopt in practice, particularly when models must be derived from operational data and revised as the system evolves.

This paper presents *SMP Workbench*, an interactive tool for generating, visualizing and refining SMP models from observed state transitions and durations. The generated model is presented as a graph, allowing domain experts to inspect the transition structure, revise the generated states, and adjust the holding-time distributions. Thus, SMP Workbench lowers the practical barrier to creating

SMPs from operational DT data and supports the adoption of SMP-based stochastic analysis, including probabilistic update scheduling.

This study makes two contributions. (1) We present a tool-supported workflow for generating SMPs from event logs. (2) We provide an interactive application that allows domain experts to inspect and refine generated SMPs before they are used for downstream stochastic analysis.

2 SMP Workbench Overview

This section presents the architecture and use of the SMP Workbench. The tool supports a workflow for constructing, inspecting, and refining SMP models from operational logs, which can then support stochastic analysis, such as probabilistic update scheduling.

The SMP Workbench consists of two main components: a web-based visual editor for loading logs, inspecting generated SMPs, and refining the models, and a Python-based backend that parses log files, extracts state transitions and sojourn-times, applies outlier detection, estimates transition probabilities, and prepares sojourn-time observations for distribution selection and further refinement by the practitioner.

2.1 Log-Based SMP Generation

The first step in the workflow is to derive a candidate SMP from operational event logs. The current prototype expects JSON Lines with one record per observed transition. Listing 1 shows the required source state, destination state, and sojourn duration (in seconds). Logs in other formats must be mapped to these fields before they are imported. From these records, the SMP Workbench constructs the state set S, the observed state transitions, and the empirical sojourn-time observations associated with those transitions.

```
{
  "from_state": "OrderStarted",
  "to_state": "Dosing",
  "sojourn_sec": 2
}
```

Listing 1: Example SMP Workbench input record

The transition structure was derived by counting the observed transitions in the logs. For each pair of states $i, j \in S$, the tool computes a transition count n_{ij}, where n_{ij} is the number of times state j is observed immediately after state i. These counts were then normalized to estimate the transition probabilities as follows:

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{k \in S} n_{ik}}, \qquad i, j \in S. \tag{1}$$

This produces a transition matrix for the candidate SMP. The generated model is transition-conditioned: the empirical holding-time observations are attached to each observed transition $i \to j$, so the duration may depend on both the current and the next state. Transitions that were not observed in the logs were not included in the automatically generated model. Similarly, holding-time observations identified as outliers were excluded from the initial estimation step but quarantined for later inspection. This keeps the candidate model traceable to the observed operational behavior while allowing practitioners to re-include unusual but valid observations during the visual refinement phase of the model.

Therefore, the generated SMP should be interpreted as a candidate model: its state set, observed transitions, transition probabilities, and sojourn-time observations are derived from logs, whereas decisions about re-including quarantined observations and selecting holding-time distributions remain subject to domain-expert refinement, which is further elaborated in Section 3.

2.2 Outlier Detection

SMP Workbench applies outlier detection before generating the SMP to prevent abnormal logs or holding-time observations from distorting the generated model. The tool uses the Interquartile Range (IQR) method [11] at two levels: to identify anomalous log files and anomalous holding-times for each observed transition.

At the file level, the tool compares each file's count of each observed transition with the corresponding distribution across all imported files. A file is quarantined when at least one count exceeds the upper IQR fence. The reported flag identifies the transition and count that triggered quarantine.

The remaining log entries are grouped according to their source and destination states to compute the transitions. For each transition $i \to j$, the tool collects the corresponding sojourn-time observations and applies the IQR rule [11] to this transition-specific set. Let Q_1 and Q_3 denote the first and third quartiles of the observed holding times for a transition, with the IQR defined as

$$\mathrm{IQR} = Q_3 - Q_1. \tag{2}$$

A holding-time observation h_{ij} is flagged as an outlier if

$$h_{ij} < Q_1 - 1.5 \cdot \mathrm{IQR} \quad \text{or} \quad h_{ij} > Q_3 + 1.5 \cdot \mathrm{IQR}. \tag{3}$$

Equation 3 defines the transition-level filtering criterion used to quarantine unusually long holding-time observations before the initial SMP is generated.

Flagged entries are quarantined from the SMP generation step but remain available for inspection. This is important because an outlier may correspond to an incomplete execution or an unusual but valid operational case. Therefore, the generated SMP is created from the cleaned transition data, whereas the quarantined data support a human-in-the-loop refinement workflow.

2.3 Visual Refinement

After the initial SMP is generated, SMP Workbench presents the model as an editable directed graph. States are represented as nodes, and observed transitions are represented as edges annotated with estimated transition probabilities, as shown in Figure 1. Selecting a transition opens a refinement view that shows the corresponding sojourn-time observations, the selected holding-time distribution, and its parameters. This allows domain experts to inspect whether the generated model reflects the expected operational behavior of the system being modeled.

The refinement step is necessary because the log-derived SMP is an initial modeling artifact rather than a validated representation. Because IQR-based outlier detection may exclude valid observations or retain abnormal ones, domain experts must review quarantined data and refine the SMP before downstream analysis.

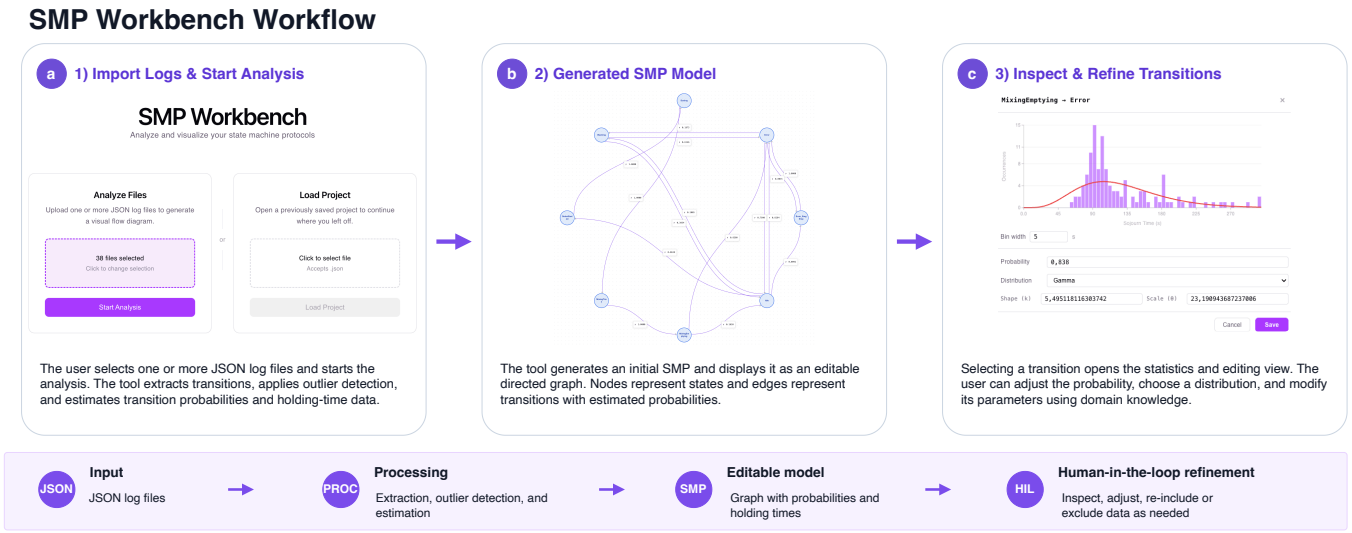


Figure 1: SMP Workbench workflow: (1) importing operational JSON log files, (2) generating an initial editable SMP, and (3) inspecting and refining transition-level information.

3 Demonstration of SMP Workbench

This section demonstrates the SMP Workbench workflow using operational logs from the fibo Collect concrete mixer[1][2]. The fibo Collect system is a semi-automated concrete mixer that produces ready-to-use concrete for customers. The case is suitable for demonstrating SMP Workbench because the system produces timestamped operational state transitions that can be used to derive an SMP. In the demonstration, we apply SMP Workbench to 38 JSON log files collected from the fibo Collect system over a two-month period, generating a candidate SMP with eight states. Each file covers one day and contains multiple, largely independent concrete mixing runs, although the runs are coupled through a shared material inventory. These states correspond to the operational phases in the concrete production process, such as dosing, mixing, and washing. Figure 1 summarizes the three-step workflow used in the demonstration.

3.1 Step 1: Import and analyze operational logs

The user first selects the JSON log files containing the observed state transitions in the format shown in Listing 1. SMP Workbench then extracts the transitions, applies outlier detection, and estimates the transition probabilities and sojourn-time observations. Of the 38 imported log files, 23 were used to generate the SMP, and 15 files were quarantined for practitioner inspection. The quarantined files contained 7,118 log entries. The tool reports the generated state-transition data and any quarantined files or log entries.

3.2 Step 2: Generate and inspect the SMP

The generated SMP is displayed as an editable directed graph. In the graph, the nodes correspond to the observed states, and the directed edges correspond to observed transitions between states. Each edge is associated with an estimated transition probability and a set of sojourn-time observations. This graph allows the practitioner to check whether the extracted states and transitions correspond to the expected fibo Collect workflow and to identify unexpected or missing transitions or implausible transition probabilities. For example, transitions between production phases such as dosing, mixing, and washing can be checked against the expected operating sequence of the fibo Collect system. Step 2 of Figure 1 shows the resulting editable graph with the associated states and transitions.

3.3 Step 3: Refine transition-level behavior

SMP Workbench also supports transition refinement. When the user selects a transition, the tool displays the empirical holding-time data for that transition, together with the estimated transition probability and distribution parameters. The practitioner can then adjust the transition probability, select the holding-time distribution, and modify the corresponding distribution parameters. This is important because a log-derived SMP is a candidate model that must be verified against domain knowledge. Some transitions may be valid but rare, whereas others may be artifacts caused by incomplete log data or unusual operating conditions.

The tool further exposes the quarantined files and log entries produced by the outlier detection step. These entries are excluded from the automatic SMP generation phase but remain available for inspection. Therefore, the practitioner can decide whether the excluded observations should remain outside the model or be re-included after inspection. This supports a human-in-the-loop workflow in which automated preprocessing is combined with domain expertise. After refinement, the candidate SMP provides the basis for downstream stochastic analysis, such as estimating favorable update windows for the DT-enabled system. However, this downstream use was not evaluated in this study.

[1] https://info.fibointercon.com/fibocollect
[2] https://shorturl.at/hDAqd

4 Related Work

Several tools and platforms have been proposed in recent years to support practitioners in creating, configuring, analyzing, and managing their DTs. Fiter et al. [4] introduce DTInsight, a tool that aims to support explicit, interactive, and continuous DT reporting, with the aim of making DT assumptions more visible to stakeholders through their threefold contribution. Fouquet et al. [6] introduce GreyCat, a framework for developing DTs over large-scale digital shadows, with support for scalable data management and efficient DT construction at scale. In addition to DT engineering tools, related studies have considered the lifecycle management, deployment, and reconfiguration of DT-enabled systems, particularly dynamic reconfiguration, which examines changes to software or production systems while remaining operational and is closely related to the problem of deciding when updates can be applied safely and are of interest for our tool. Muram et al. [10] considered dynamic reconfiguration in safety-critical production systems, where configuration changes must be assessed with respect to safety, performance, and production demands. Similarly, Alhozaimy et al. [1] quantified the trade-off between resilience and performance when dynamic reconfiguration is used as an adaptive mechanism. These studies motivate the need to determine when operational changes should occur, but they do not focus on the tool-supported construction of stochastic behavioral models from DT execution data.

Log-based model generation is also closely related to this paper, where process mining provides methods for discovering models from event logs analyzing process performance. Process mining techniques have demonstrated that operational traces can be used to derive behavioral models of the processes [3, 9]. Kalenkova et al. [8] specifically investigated the discovery of semi-Markov models from event logs for performance analysis. However, such approaches are typically concerned with automated model discovery and analysis rather than an interactive modeling workflow in which domain experts can inspect, correct, and refine the generated models. In contrast, our tool, SMP Workbench, treats log-derived models as initial modeling artifacts that can be visualized and adjusted before being used for analysis.

Tool support for SMPs has also been studied in dependability and safety-critical systems literature. Kaalen et al. [7] extended transition-timer-based SMP modeling with branching transitions and provided tool support for dependability analysis. Their work demonstrated the use of SMPs with transition timers and branching transitions for modeling timed transition behavior and supporting dependability analysis. The focus of SMP Workbench is different. Rather than providing analysis support for manually specified SMPs, it supports the generation, visualization, and practitioner-guided refinement of SMPs from operational DT logs. Taken together, the existing work provides an important foundation for DT engineering, process mining, dynamic reconfiguration and SMP analysis. However, there is limited tool support for bridging operational DT data and editable SMP models for stochastic analysis in DT engineering. SMP Workbench addresses this gap by supporting the import of log files, automatic generation of a candidate SMP, visual inspection of the resulting state-transition structure, and refinement of transition probabilities and holding-time distributions before stochastic analysis.

5 Conclusion

This case study illustrates the feasibility of using SMP Workbench to construct candidate SMP models from empirical operational data. The tool derives the operational states and transition probabilities from the observed event traces, providing a model for domain expert review. Therefore, the main contribution of the SMP Workbench is not full automation but a human-in-the-loop workflow for tool-supported model construction and refinement. By combining log import, state transition extraction, visualization, and interactive refinement, the tool provides a workflow in which practitioners can move from raw operational data to an SMP model. This is particularly important for update-scheduling use cases, where the model must not only be technically well-formed, but also meaningful to domain experts who understand the operational states, abnormal cases, and practical update constraints.

5.1 Limitations

In the case study, the IQR-based sojourn-time outlier detection helped identify anomalous observations, but it should not be treated as a complete validation mechanism. However, some anomalous cases may still be missed, reinforcing the need for manual inspection. Therefore, the results support domain experts in inspecting quarantined files and sojourn-time observations, excluding artifacts, and re-including observations that represent valid operational behavior. File-level outlier detection showed a similar behavior, but its accuracy was not evaluated against a labeled anomaly data set.

The current implementation also depends on the quality and consistency of the input logs. Missing events, inconsistent event names, duplicate entries, and changes in the logging format may affect the generated SMP structure and estimated transition behavior. This implies that the generated model should be interpreted as a data-derived modeling artifact rather than a complete ground-truth representation of the physical system itself. In practice, the usefulness of the generated SMP depends on the quality of the logs and practitioners' ability to validate the extracted states and transitions.

5.2 Future Work

Future work will focus on the analytical support provided by SMP Workbench. First, we plan to extend the tool with semi-automated selection of holding-time distributions by incorporating statistical goodness-of-fit methods and model-comparison criteria. This would provide practitioners with better support when choosing distributions, while still allowing domain experts to override or refine the suggested model.

We also plan to extend SMP Workbench with step-by-step simulation of the generated SMP. A simulation would allow practitioners to inspect the likely execution paths, state durations, and transition behaviors before using the model for downstream analysis.

Finally, we look into including a user study with domain experts and more case studies for further evaluation of SMP Workbench. This allows us to assess the usability of the interactive modeling workflow with different logs.

The source code for SMP Workbench is available at https://github.com/INTO-CPS-Association/SMP-Workbench.

References

[1] Sarah Alhozaimy, Daniel A. Menasc'e, and Massimiliano Albanese. 2024. Resilience and Performance Quantification of Dynamic Reconfiguration. *Future Generation Computer Systems* 160 (2024), 120–130. doi:10.1016/j.future.2024.05.040

[2] Mikkel Schmidt Andersen, Claudio Gomes, Sophia Thompson, and Peter Gorm Larsen. 2025. Probabilistic Update Scheduling for Digital Twins: A Semi-Markov Approach . In *2025 ACM/IEEE 28th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE Computer Society, Los Alamitos, CA, USA, 250–256. doi:10.1109/MODELS-C68889.2025.00043

[3] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. 2019. Automated Discovery of Process Models from Event Logs: Review and Benchmark. *IEEE Transactions on Knowledge and Data Engineering* 31, 4 (2019), 686–705. doi:10.1109/TKDE.2018.2841877

[4] Kérian Fiter, Louis Malassigné-Onfroy, and Bentley Oakes. 2025. DTInsight: A Tool for Explicit, Interactive, and Continuous Digital Twin Reporting. In *2025 ACM/IEEE 28th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 139–143. doi:10.1109/MODELS-C68889.2025.00030

[5] John Fitzgerald, Cláudio Gomes, and Peter Gorm Larsen (Eds.). 2024. *The Engineering of Digital Twins* (1 ed.). Springer, Cham. XXV, 386 pages. doi:10.1007/978-3-031-66719-0

[6] Francois Fouquet, Thomas Hartmann, Cyril Cecchinel, and Benoit Combemale. 2024. GreyCat: A Framework to Develop Digital Twins at Large Scale. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems* (Linz, Austria) *(MODELS Companion '24)*. Association for Computing Machinery, New York, NY, USA, 492–495. doi:10.1145/3652620.3688265

[7] Stefan Kaalen and Mattias Nyberg. 2020. Branching Transitions for Semi-Markov Processes with Application to Safety-Critical Systems. In *Model-Based Safety and Assessment*, Marc Zeller and Kai Höfig (Eds.). Springer International Publishing, Cham, 68–82.

[8] Anna Kalenkova, Lewis Mitchell, and Matthew Roughan. 2025. Performance Analysis: Discovering Semi-Markov Models From Event Logs. *IEEE Access* 13 (2025), 38035–38053. doi:10.1109/ACCESS.2025.3546033

[9] Sander JJ Leemans, Dirk Fahland, and Wil MP Van Der Aalst. 2013. Discovering block-structured process models from event logs-a constructive approach. In *International conference on applications and theory of Petri nets and concurrency*. Springer, 311–329.

[10] Faiz Ul Muram, Muhammad Atif Javed, Hans Hansson, and Sasikumar Punnekkat. 2020. Dynamic Reconfiguration of Safety-Critical Production Systems. In *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 120–129. doi:10.1109/PRDC50213.2020.00023

[11] Ronald Rousseau, Leo Egghe, and Raf Guns. 2018. Chapter 4 - Statistics. In *Becoming Metric-Wise*. Chandos Publishing, 67–97. doi:10.1016/B978-0-08-102474-4.00004-2

[12] Prasad Talasila, Cláudio Gomes, Lars B Vosteen, Hannes Iven, Martin Leucker, Santiago Gil, Peter H Mikkelsen, Eduard Kamburjan, and Peter G Larsen. 2025. Composable digital twins on Digital Twin as a Service platform. *Simulation* 101, 3 (2025), 287–311.