FrOG: An OCEL generator for fragmented supply orders

Jost Goette[1,*,†], Maximilian Harms[1,†] and Henrik Leopold[1,†]

[1]Kuehne Logistics University, Hamburg, Germany

Abstract

Process mining is widely used to analyze, understand, and improve business processes. Object-Centric Process Mining (OCPM) extends traditional approaches by capturing interactions between multiple object types within a process. In supply chains, these interactions exhibit domain-specific behavior, such as order fragmentation, for example, when a supplier cannot complete an order. Yet suitable event data are scarce due to multi-party privacy constraints. We introduce FrOG (Fragmented Order Generator), a configurable generator of object-centric event logs (OCEL) derived from a warehouse simulation with industry-standard stock policies. FrOG produces near-realistic supply-order data that capture fragmentation and cross-object dependencies, enabling OCPM studies without access to sensitive operational logs.

Keywords

OCPM, warehouse simulation, synthetic OCEL, data generator, fragmented orders

Metadata description	Value
Tool name	FrOG
Current version	1.0
Legal code license	MIT
Languages, tools and services used	Python, PM4PY, Dash, Pandas
Supported operating environment	Microsoft Windows, GNU/Linux, MacOS
Download/Demo URL	https://github.com/klu-ds/FrOG
Documentation URL	https://github.com/klu-ds/FrOG/blob/main/README.md
Source code repository	https://github.com/klu-ds/FrOG
Screencast video	https://youtu.be/MxbSjiawY30

1. Introduction

Global supply chains are complex, fast-moving, and inherently uncertain. As demand and supply fluctuate, suppliers often adapt by splitting large orders into multiple shipments based on what is available. This improves service, but it also complicates post-order evaluation, because it blurs lead times, fulfillment rates, and cost comparisons.

One established concept for analyzing supply orders is Process Mining [1]. It offers methods and techniques for analyzing and improving operational processes based on event data and

ICPM Doctoral Consortium and Demo Track 2025, October 20-24, 2025, Montevideo, Uruguay

*Corresponding author.
†These authors contributed equally.
jost.goette@klu.org (J. Goette); maximilian.harms@klu.org (M. Harms); henrik.leopold@klu.org (H. Leopold)

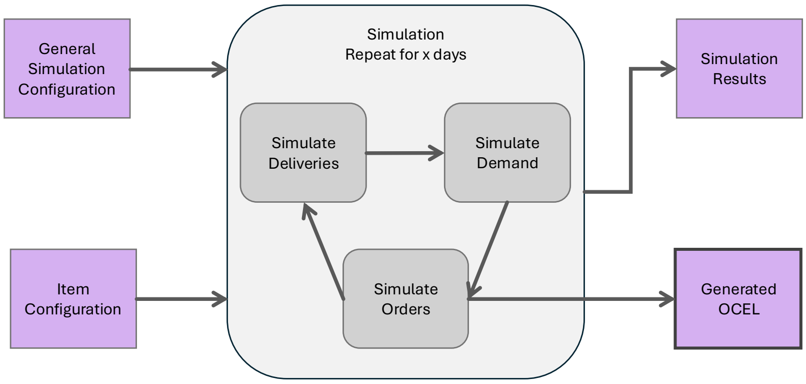

Figure 1: Overview of the pipeline of FrOG.

plays an important role in identifying deviations, inefficiencies, and improvement opportunities. Traditional process mining techniques typically assume the availability of flat event logs, where each event is associated with a single case identifier. While effective in many scenarios, this representation introduces limitations, for instance due to the duplication of events when multiple process entities interact within a single execution [2].

Object-centric process mining (OCPM) has emerged as a paradigm that accounts for multiple interacting objects within a process. Instead of relying on a single case notion, OCPM represents events in terms of the objects they affect, enabling a more accurate and flexible view of the execution of real-world processes [2]. However, restrictions such as privacy concerns hinder access to specialized event data of the supply chain management (SCM) domain [1]. Data that would be necessary to advance process mining techniques for the SCM field.

Existing tools for synthetic event log generation [3] [4] provide realistic control-flow and data perspectives but remain limited to flat case structures, thereby insufficiently capturing complex interactions between multiple objects. More recent object-centric generators [5] extend this by modeling multi-object relations, yet they lack the capability to represent domain-specific supply chain phenomena like repeated order fragmentation. Logistics-focused simulators [6] primarily address inventories, routes, and order executions without explicitly modeling the systematic splitting of customer orders into multiple partial deliveries. Consequently, existing approaches cannot realistically reproduce the fragmentation dynamics characteristic of split-order deliveries in event logs.

To facilitate the investigation of novel OCPM techniques that address the challenges of SCM, we propose FrOG (Fragmented Order Generator), a synthetic OCEL generation tool, that uses a warehouse simulation to create authentic orders based on common stock-policies. The tool lets it users customize the simulation and especially the degree to which orders are split. It offers access to basic OCPM techniques to analyze the resulting OCEL as well as extract it.

2. The tool

FrOG is implemented in Python. The frontend is built with DASH [7]. For simulation and OCEL generation we use Pandas [8] and PM4PY [9]. Figure 1 shows the high level workflow

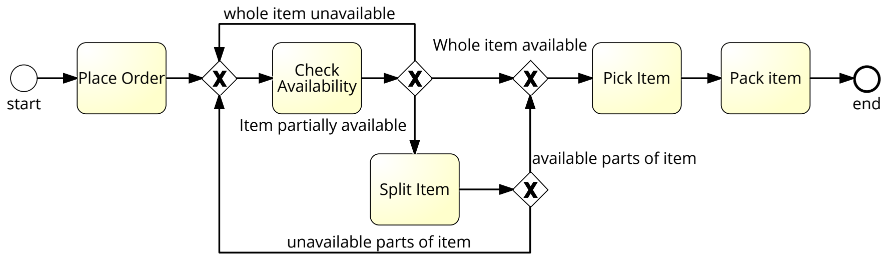

Figure 2: Overview of the order process we use for our evaluation as BPMN.

of the FrOG. It lets the user configure various parameters of a common warehouse simulation on the *simulation* page. Then, users can define various items with different behavior that the warehouse manages. After that, the simulation runs for the set amount of days. Over the course of the simulation, the warehouse has to fulfill demand and prevent shortages. Each day, it cycles through the simulation of deliveries, demand and orders. These orders are gathered and consolidated into an OCEL. At the end the user is on the one hand presented with data about the simulation and on the other hand they can analyze the created OCEL and export it on the analysis page.

2.1. The Simulation

The core of FrOG is the user-configured warehouse simulation. In supply chain management, warehouse simulation is a proven tool to investigate and optimize warehouse policies [10]. Our simulation constructs a virtual warehouse, that manages the inventory for the user-defined items. Each item refers to one type of goods. Prior to the simulation, the user can configure the starting inventory, order cost, holding cost, mean and standard deviation for the daily demand for each created item.

The warehouse follows the classical continuous review policy [11]. This policy determines the economic order quantity (EOQ) and reorder point (ROP) to sustainably manage the operation of the defined goods. The EOQ utilizes the mean daily demand and holding cost of an item to determine the optimal quantity of an order. The ROP uses the mean demand and lead times to determine the optimal stock level to reorder again. The starting EOQ and ROP can be configured by the user and are then adapted by the simulation accordingly.

Each simulated day follows the same structure. First, the warehouse receives the due deliveries, evaluates the corresponding orders and updates its parameters like the reorder point accordingly. Second, it invokes the random demand for each of its managed items and reduces the inventory accordingly. Third, if the stock-level of any item drops below the respective ROP it creates a new order for that item and hands it over to the OCEL-generation module to convert that pre-configured order into one process instance of an OCEL.

2.2. Simulated Order Process

Figure 2 shows BPMN diagram of the fixed order process that the simulation triggers at each ROP. It modifies the order-to-delivery process found in [2] with the corresponding behaviour for partial deliveries. Each order process begins with the “Place Order” event, which initiates an order and creates associated item objects. Each item represents a specific quantity of a material. The order then proceeds through typical administrative steps, including “Send Invoice” and “Receive Payment”.

Once the order is placed, item availability is continuously evaluated through the “Check Availability” activity. Based on material availability at the time of checking, the process branches into three possible paths:

1. **No availability**: If none of the materials for an item are currently available, the item is disregarded and re-evaluated in the next availability check.
2. **Full availability**: If the full quantity of materials is available, the item is directly picked, packed, and included in a shipment.
3. **Partial availability**: If only part of the item’s materials are available, a split occurs (as modeled by the “Split item” activity). The item is divided into two new item objects: one containing the available materials, and another holding the remaining quantity. The available portion continues through the pick, pack, and deliver sequence, while the remaining portion loops back to the availability check.

This fragmentation may repeat multiple times until all materials are eventually picked and delivered. The user defines the average degree to which orders of an item are split, when they configure the item.

2.3. Output

After the simulation is complete, FrOG offers two types of output to the user. First is the supply chain perspective offering data with regards to the simulation itself. This includes the total fulfilled the demand, the amount of orders, unfulfilled demand, and the service level (relative fulfilled demand). These information are computed for each item as well as for the whole warehouse globally.

The second type of output is the OCEL that FrOG constructs using all orders issued during the simulation. It adheres to the OCEL 2.0 standard [12] and thus includes O2O relations. This OCEL is presented to the user on the *analysis* page of the tool and is accompanied by several summarizing stats of the OCEL gathered from the PM4PY [9] function OCEL summary function. Finally, the user may download the OCEL for further work in the OCEL JSON format [12].

3. Maturity

We assess the maturity of FrOG by assessing its scalability. For increasing parameter values that target the tool’s complexity, we evaluate the runtime of the simulation and the following OCEL creation. This benchmark can be recreated on *benchmark* page.

We consider the simulation days, the amount of items, and the mean amount of splits to be the driving dimensions of FrOG’s scalability. They are the most relevant dimensions for the envisaged use case. We diagonalize the benchmark over these dimensions to cover all possible configurations within the following bounds: 1000, 2000, 5000 and 10000 simulation days, 1, 5, 10, and 50 items, as well as 0, 1, 5, 10 mean splits. We repeat each configuration three times.

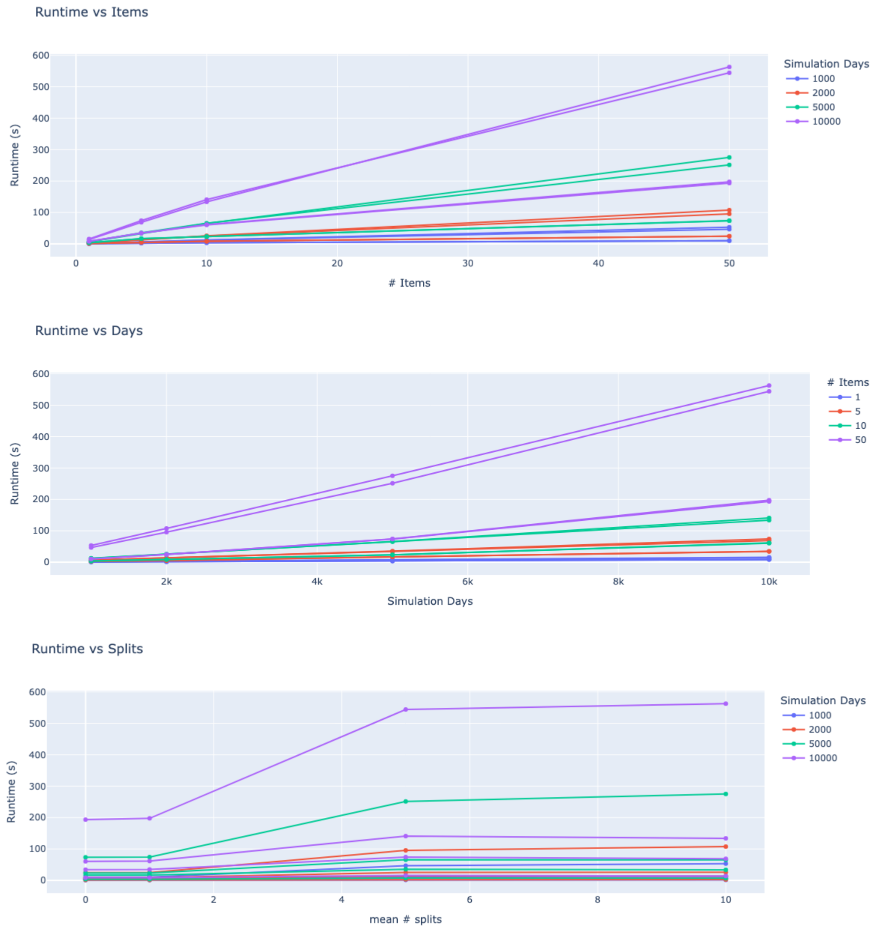

Figure 3: Shows the influence on the runtime over increasing values over the three target dimensions.

The results in Figure 3 indicate different effects on the scalability depending on which dimension is increased. Increasing the splits from zero to one, as well as 5-10 results in similar run times, albeit an increase between one and five severely increases the run time. The effect on the run time of increasing amounts of items decreases with more items. Increasing the amount of days leads to slight super linear growth of the run time.

4. Outlook

In this paper, we proposed FrOG, which generates OCELs from orders created as part of a realistic warehouse simulation. While the tool is a step towards facilitating OCPM techniques for more complex supply chain related processes, there a still improvement opportunities left. Future work should consider adding more functionality, such as a dedicated OCPM page that hosts existing OCPM techniques to apply them on the created OCEL. It could also concern itself with other processes related to the warehouse or adding different behavior to the warehouse, such as configurable stock policies. The current implementation does not consider the availability of resources that could limit warehouse throughput.

Declaration on Generative AI

The author(s) used ChatGPT to: polish the code of the frontend. The author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] F. Oldenburg, K. Hoberg, H. Leopold, Process mining in supply chain management: state-of-the-art, use cases and research outlook, International Journal of Production Research 63 (2025) 2889–2904.

[2] W. M. van der Aalst, Object-centric process mining: dealing with divergence and convergence in event data, in: Software Engineering and Formal Methods: 17th International Conference, SEFM 2019, Oslo, Norway, September 18–20, 2019, Proceedings 17, Springer, 2019, pp. 3–25.

[3] D. Jilg, J. Grüger, T. Geyer, R. Bergmann, DALG: the data aware event log generator, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2023 co-located with 21st International Conference on Business Process Management (BPM 2023), Utrecht, The Netherlands, September 11th to 15th, 2023, volume 3469 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 142–146.

[4] Ändrea Burattin, B. Re, L. Rossi, F. Tiezzi", Purple: a purpose-guided log generator, in: Proceedings of the ICPM Doctoral Consortium and Demo Track 2022, CEUR Workshop Proceedings, CEUR-WS, 2022.

[5] A. Goossens, A. Rebmann, J. D. Smedt, J. Vanthienen, H. v. d. Aa, From OCEL to DOCEL – Datasets and Automated Transformation, 2023. URL: http://arxiv.org/abs/2309.14092. doi:10.48550/arXiv.2309.14092, arXiv:2309.14092 [cs].

[6] A. Wuttke, J. Hunker, M. Rabe, LogFarm: An Open Source Graph-based Simulator for Logistics Networks, SNE Simulation Notes Europe 34 (2024) 43–50. URL: https://www.sne-journal.org/10676. doi:10.11128/sne.34.sw.10676.

[7] P. T. Inc., Dash documentation & user guide, 2025. URL: https://dash.plotly.com/, accessed August 13, 2025.

[8] T. pandas development team, pandas-dev/pandas: Pandas, 2020. URL: https://doi.org/10.5281/zenodo.3509134. doi:10.5281/zenodo.3509134.

[9] A. Berti, S. van Zelst, D. Schuster, Pm4py: A process mining library for python, Software Impacts 17 (2023) 100556. URL: https://www.sciencedirect.com/science/article/pii/S2665963823000933. doi:https://doi.org/10.1016/j.simpa.2023.100556.

[10] J. P. Gagliardi, J. Renaud, A. Ruiz, A simulation model to improve warehouse operations, in: 2007 Winter Simulation Conference, 2007, pp. 2012–2018. doi:10.1109/WSC.2007.4419831.

[11] S. Axsäter, Inventory control, number 90 in International series in operations research & management science, 2nd ed ed., Springer, New York, 2006.

[12] A. Berti, I. Koren, J. N. Adams, G. Park, B. Knopp, N. Graves, M. Rafiei, L. Liß, L. T. G. Unterberg, Y. Zhang, C. Schwanen, M. Pegoraro, W. M. P. van der Aalst, Ocel (object-centric event log) 2.0 specification, 2024. URL: https://arxiv.org/abs/2403.01975. arXiv:2403.01975.