

ocpa: A Python Library for Object-Centric Process Analysis

Jan Niklas Adams¹ (niklas.adams@pads.rwth-aachen.de, Process and Data Science, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany)

Gyunam Park (gnpark@pads.rwth-aachen.de, Process and Data Science, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany)

Wil M.P. van der Aalst (wvdaalst@pads.rwth-aachen.de, Process and Data Science, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany)

Abstract

OCPA is a Python library supporting object-centric process mining. Traditional process mining generate insights for one single process. However, many real-life processes are composed of multiple interacting subprocesses and events may involve multiple objects. Object-centric process mining provides techniques for analyzing multiple interacting processes by generalizing process mining techniques. OCPA contains algorithms for object-centric event log management, process discovery, conformance checking, enhancement, and predictive process monitoring. OCPA is easily integrable with existing solutions as it supports existing object-centric event log standards, provides extensive documentation, and is installable through the Python package installer.

Keywords

Process Mining, Object-Centric Process Mining, Object-Centric Petri Net, Predictive Process Monitoring, Conformance Checking, Performance Analysis

¹Corresponding author

Code metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	1.0.0
C2	Permanent link to code/repository used for this code version	https://github.com/ocpm/ocpa
C3	Permanent link to Reproducible Capsule	https://codeocean.com/capsule/7687700/tree/v1
C4	Legal Code License	GPL-3.0
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python
C7	Compilation requirements, operating environments & dependencies	pandas, numpy, pm4py, networkx, graphviz
C8	If available Link to developer documentation/manual	https://ocpa.readthedocs.io/
C9	Support email for questions	niklas.adams@pads.rwth-aachen.de

1. Introduction

Process mining offers techniques and algorithms to analyze processes based on the event data these processes generated. In the traditional view, each execution of the process generates a sequence of events describing conducted activities and associated data. An event log collects event sequences of different process executions. Starting from the event log, one can uncover different insights typically grouped into four areas: **(1)** Process Discovery: Learning a model that covers the possible sequences of activities in the event log [8]. **(2)** Conformance Checking: Testing whether the event sequences conform to given rules or exhibit deviations [11]. **(3)** Process Enhancement: Equipping a process model with additional information about performance, decisions, or resources [18]. **(4)** Predictive Process Monitoring: Extracting features from the event log and learning predictive models for different targets [20].

One central assumption underlying all of the developed techniques is the following: *One execution of the process is a sequence of events*. However, this assumption does not hold in many real-life situations: For example, when analyzing multiple processes and their interactions or a process composed of multiple subprocesses [15]. One example of such a process encountered in reality is a production process where many individual parts are produced in subprocesses and assembled later. Another example is an ERP system supporting business processes where one process execution consists of multiple documents associated with different actions. In these situations, each subprocess produces one sequence of events. Since events are shared between sequences, the behavior takes the form of a graph rather than a sequence. Currently, the only way to apply process mining techniques in such situations is *flattening*, i.e., enforcing a sequentiality for events. Flattening introduces problems of *duplicated events*, *incorrect dependency constraints*, and *disappearing events* [1, 14, 26].

Object-centric process mining addresses the limitations of traditional process mining techniques by removing the assumption of a sequential event structure and allowing for a more general, graph-based event structure encountered in reality. Specifically, an event can be related to multiple *objects*. Each object represents the instantiation of a subprocess. Since one event can be associated with multiple objects, it can represent an interaction point between subprocesses.

Recently, techniques to solve the four process mining tasks for object-centric processes have been introduced: object-centric process discovery [2], object-centric conformance checking [5], object-centric performance analysis [22], and object-centric predictive process monitoring [6]. So far, no unified platform for object-centric process mining exists. OCPA bundles all the proposed techniques into one single comprehensive Python library (cf. Figure 1). This library has two main implications: First, enabling practitioners to quickly deploy object-centric process mining to analyze federated processes or complex systems composed of multiple subprocesses. Second, equipping researchers with fully implemented state-of-the-art techniques that may be used to propose new and advance existing techniques.

In the remainder of this paper, we will first introduce an overview of OCPA’s functionality and, subsequently, discuss its foundation in research and impact on future research and practical deployment.

2. Software Overview

OCPA is provided as a *GitHub* repository² and through the package installer *pip*.³ We provide the core code framework accompanied by example data and example scripts for the integration of our library. Extensive documentation can be found on *readthedocs*.⁴ The documentation provides a set of examples for each algorithm. The core code framework is split into a data management, algorithm, and visualization submodule.

2.1 Data Management Submodule

The data management submodule contains all functionality to store and access object-centric event data. Functionalities are split into import, access, and export. We allow the import of object-centric event logs in one of three formats: CSV, JSONOCEL, and XMLOCEL⁵. When importing a CSV file, information on the column

²<https://github.com/ocpm/ocpa>

³<https://pypi.org/project/pip/>

⁴<https://ocpa.readthedocs.io>

⁵Both have been defined in the OCEL standard www.ocel-standard.org

mapping must be provided. The data contained in the object-centric event log can be accessed through the OCEL class, forming the central data storage object of this library. A user can access event and object values, as well as trigger more elaborate derivatives of the event log, such as process executions, variants, and various statistics on these objects. To enable interoperability with other tools OCPA supports the export of an OCEL object to JSONOCEL format. Users can use this functionality to convert CSV to JSONOCEL.

2.2 Algorithm Submodule

The algorithm submodule comprises the main algorithms and techniques introduced in object-centric process mining over the last years. Starting from an object-centric event log, different algorithms of object-centric process discovery, conformance checking, process enhancement, and predictive monitoring are available.

Object-Centric Process Discovery OCPA enables the investigation of the control-flow in object-centric processes by two ways: *process discovery* and *variant analysis*. First, process models can be discovered as object-centric Petri nets following the general approach by van der Aalst and Berti [2]. The object-centric Petri net is a class aimed at bundling different functionality to analyze object-centric Petri nets. Second, the user can discover control-flow variants following the algorithm of Adams et al. [7]. In contrast to activity sequences in traditional process mining, each variant is a directed, acyclic graph of activities.

Object-Centric Conformance Checking Two main conformance checking techniques are supported by OCPA: *evaluation metrics* and *constraint monitoring*. Evaluation metrics include precision and fitness of object-centric Petri nets with respect to object-centric event logs [5]. Fitness is the share of replayable events, while precision quantifies the share of possible model behavior contained in the event log. Constraint monitoring evaluates the violation of user-defined business constraints by analyzing object-centric event logs. Such constraints include control-flow constraints (e.g., activity *place order* is directly followed by *send invoice* in the lifecycle of object *order*), object involvement constraints (e.g., the execution of activity *clear invoice* should involve an order along with the corresponding invoice and goods receipt), and performance constraints (e.g., the average synchronization time of *clear invoice*, i.e., the time taken for preparing all orders, invoices, and goods receipts, should be less than three days).

Object-Centric Process Enhancement OCPA supports *object-centric performance analysis* of both temporal and non-temporal performance measures. The temporal performance measures include object-centric performance measures (*flow time*, *synchronization time*, *pooling time*, and *lagging time*) as well as traditional performance measures (*waiting time*, *service time*, and *sojourn time*). To compute such measures for each activity, we take a model-based approach presented in [22]. The approach first discovers an object-centric Petri net from an object-centric event log, 2) then replays the log in the model, and finally computes the aforementioned performance measures. The non-temporal performance measures include *object count*, *object type count*, etc.

Object-Centric Predictive Process Monitoring OCPA provides a foundation for object-centric predictive process monitoring through three main components: feature extraction, preprocessing, and feature encoding. Feature extraction and encoding have been introduced in [6]. Features are extracted based on the graph structure of the object-centric event log, i.e., the feature values accurately depict reality (while feature extraction based on a flat event log provides only one value for a preceding activity, object-centric feature extraction provides one value for each object's preceding activity). Extracted features can be preprocessed by normalizing and splitting the data into training and testing sets. These can be encoded in three ways: As a table, sequences, or graphs. All encodings maintain a different level of the object-centric event data's structure, whereas graph encoding preserves the most structural information. The encodings can be used for different predictive models, e.g., regression (table), LSTM [17] (sequences), or graph neural networks [24] (graphs).

2.3 Visualization Submodule

The visualization submodule offers support for the visualization of different objects in the library. The visualization of object-Centric Petri nets is supported through the GraphViz library.⁶ Next to the layouting through GraphViz, OCPA also offers the export and storage of Petri net visualizations.

Variant graphs can become complex, including activity labels, object types, and objects. Therefore, dedicated variant visualization techniques have been proposed [7]. OCPA includes the proposed layouting algorithm for variants. The algorithm provides a two-dimensional layouting of activity labels and colors that is an extension of traditional variant visualization.

3. Impact Overview

OCPA summarizes several contributions of object-centric process mining, enabling researchers and practitioners to pursue new research questions and applications.

Research Foundations of OCPA The main functionality included in OCPA has been introduced in research over the last two years. Starting from object-centric event logs and their management [1, 16], these contributions cover the discovery of object-centric Petri nets [2] and variants [4, 7], object-centric evaluation metrics [5], object-centric performance analysis [22], and object-centric feature extraction and encoding [6]. While OCPA implements the functionalities of the previously listed research papers, other research contributions are also partly supported in OCPA. The philosophy of graph-structured event data which is present in [14, 15, 26] forms the core philosophy in OCPA. It also easily integrates into proposed solutions for extracting and creating object-centric event logs from information systems or traditional event logs [23, 9].

Impact of OCPA to Current Research OCPA delivers functionalities that consider the graph-based structure of object-centric event data for all process mining tasks. Therefore, it enables the pursuit of new research questions by translating traditional process mining techniques into the object-centric setting by applying a graph setting. With an existing event log management, process execution extraction, process discovery, model quality metrics, performance analysis, and feature extraction and encoding, many new techniques can be explored. These include, but are not limited to: How to cluster object-centric event data? Which models are best suited for object-centric predictive process monitoring? What are alternative techniques for conformance checking? OCPA enables the development of techniques that work without flattening the event data, i.e., transforming an object-centric event log into a traditional event log.

Impact of OCPA to Applied Process Mining Process mining has been applied to analyzing business processes in various industry segments [3]: *Finance* (e.g., banks and insurance companies) [27], *Manufacturing* (e.g., pharmaceuticals and automobile companies) [13], and *Service* (e.g., healthcare, government, and telecommunication) [21]. Process mining application has been accompanied by the tremendous commercial success of process mining vendors like *Celonis*, *Minit* (now Microsoft), *Signavio* (now SAP), or *MyInvenio* (now IBM).

Object-centricity is an omnipresent property in real-life business processes [12]. Thus, various object-centric process mining techniques supported by OCPA can support the application of process mining in different industry segments. One can, e.g., analyze the ensemble of processes interacting throughout the end-to-end order processing within companies [25]. Furthermore, object-centric process mining can be applied to production processes where different production components are represented by objects [10, 19]. First vendors already prepare support for object-centric event data, e.g., Celonis (Multi-Event Log) or *Mehrwerk Process Mining*.

4. Conclusion

This paper introduced OCPA. This Python library offers extensive support for object-centric event data without converting them to traditional event data. Hence, OCPA offers more accurate process analytics for object-centric event data. The comprehensive support of all four process mining tasks and OCPA's easy integrability through python enables the pursuit of many future research questions and process analyses.

⁶<https://graphviz.org/>

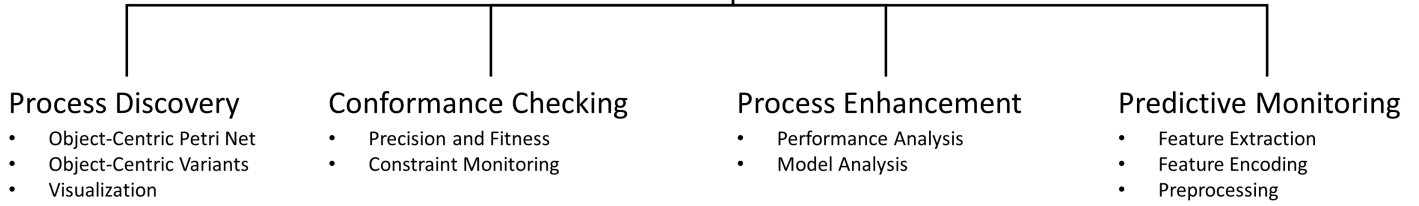


Figure 1: Overview for the functionalities contained in OCPA.

Funding

We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

- [1] Wil M. P. van der Aalst. “Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data”. In: *SEFM*. Springer, 2019, pp. 3–25. DOI: [10.1007/978-3-030-30446-1_1](https://doi.org/10.1007/978-3-030-30446-1_1).
- [2] Wil M. P. van der Aalst and Alessandro Berti. “Discovering Object-centric Petri Nets”. In: *Fundam. Informaticae* 175.1-4 (2020), pp. 1–40. DOI: [10.3233/FI-2020-1946](https://doi.org/10.3233/FI-2020-1946).
- [3] Rafael Accorsi and Julian Leberherz. “A Practitioner’s View on Process Mining Adoption, Event Log Engineering and Data Challenges”. In: *Process Mining Handbook*. Vol. 448. Lecture Notes in Business Information Processing. Springer, 2022, pp. 212–240. DOI: [10.1007/978-3-031-08848-3_7](https://doi.org/10.1007/978-3-031-08848-3_7).
- [4] Jan Niklas Adams and Wil M. P. van der Aalst. “OC π : Object-Centric Process Insights”. In: *PETRI NETS*. Springer, 2022, pp. 139–150. DOI: [10.1007/978-3-031-06653-5_8](https://doi.org/10.1007/978-3-031-06653-5_8).
- [5] Jan Niklas Adams and Wil M. P. van der Aalst. “Precision and Fitness in Object-Centric Process Mining”. In: *ICPM*. IEEE, 2021, pp. 128–135. DOI: [10.1109/ICPM53251.2021.9576886](https://doi.org/10.1109/ICPM53251.2021.9576886).
- [6] Jan Niklas Adams, Gyunam Park, Sergej Levich, Daniel Schuster, and Wil M. P. van der Aalst. “A Framework for Extracting and Encoding Features from Object-Centric Event Data”. In: *(PREPRINT. Accepted for publication at International Conference on Service-Oriented Computing (ICSOC) Conference 2022, Springer)*, CoRR abs/2209.01219 (2022). arXiv: [2209.01219](https://arxiv.org/abs/2209.01219).
- [7] Jan Niklas Adams, Daniel Schuster, Seth Schmitz, Günther Schuh, and Wil M. P. van der Aalst. “Defining Cases and Variants for Object-Centric Event Data”. In: *(PREPRINT. Accepted for publication at International Conference on Process Mining (ICPM) Conference 2022, IEEE)*, CoRR abs/2208.03235 (2022). arXiv: [2208.03235](https://arxiv.org/abs/2208.03235).
- [8] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. “Automated Discovery of Process Models from Event Logs: Review and Benchmark”. In: *IEEE Trans. Knowl. Data Eng.* 31.4 (2019), pp. 686–705. DOI: [10.1109/TKDE.2018.2841877](https://doi.org/10.1109/TKDE.2018.2841877).
- [9] Alessandro Berti, Gyunam Park, Majid Rafiei, and Wil M. P. van der Aalst. “An Event Data Extraction Approach from SAP ERP for Process Mining”. In: *ICPM Workshops*. Vol. 433. Springer, 2021, pp. 255–267. DOI: [10.1007/978-3-030-98581-3_19](https://doi.org/10.1007/978-3-030-98581-3_19).
- [10] Tobias Brockhoff, Merih Seran Uysal, Isabelle Terrier, Heiko Göhner, and Wil M. P. van der Aalst. “Analyzing Multi-level BOM-Structured Event Data”. In: *ICPM Workshops*. Vol. 433. Springer, 2021, pp. 47–59. DOI: [10.1007/978-3-030-98581-3_4](https://doi.org/10.1007/978-3-030-98581-3_4).

- [11] Josep Carmona, Boudewijn F. van Dongen, Andreas Solti, and Matthias Weidlich. *Conformance Checking - Relating Processes and Models*. Springer, 2018. ISBN: 978-3-319-99413-0. DOI: [10.1007/978-3-319-99414-7](https://doi.org/10.1007/978-3-319-99414-7).
- [12] David Cohn and Richard Hull. “Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes”. In: *IEEE Data Eng. Bull.* 32.3 (2009), pp. 3–9.
- [13] Mahendrawathi ER, Noval Arsad, Hanim Maria Astuti, Renny Pradina Kusumawardani, and Rivia Atmajaningtyas Utami. “Analysis of production planning in a global manufacturing company with process mining”. In: *J. Enterp. Inf. Manag.* 31.2 (2018), pp. 317–337. DOI: [10.1108/JEIM-01-2017-0003](https://doi.org/10.1108/JEIM-01-2017-0003).
- [14] Stefan Esser and Dirk Fahland. “Multi-Dimensional Event Data in Graph Databases”. In: *J. Data Semant.* 10.1 (2021), pp. 109–141. DOI: [10.1007/s13740-021-00122-1](https://doi.org/10.1007/s13740-021-00122-1).
- [15] Dirk Fahland. “Process Mining over Multiple Behavioral Dimensions with Event Knowledge Graphs”. In: *Process Mining Handbook*. Vol. 448. Lecture Notes in Business Information Processing. Springer, 2022, pp. 274–319. DOI: [10.1007/978-3-031-08848-3_9](https://doi.org/10.1007/978-3-031-08848-3_9).
- [16] Anahita Farhang Ghahfarokhi, Gyunam Park, Alessandro Berti, and Wil M. P. van der Aalst. “OCEL: A Standard for Object-Centric Event Logs”. In: *ADBIS 2021 Short Papers, Doctoral Consortium and Workshops*. Vol. 1450. Communications in Computer and Information Science. Springer, 2021, pp. 169–175. DOI: [10.1007/978-3-030-85082-1_16](https://doi.org/10.1007/978-3-030-85082-1_16).
- [17] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (1997), pp. 1735–1780. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [18] Massimiliano de Leoni. “Foundations of Process Enhancement”. In: *Process Mining Handbook*. Vol. 448. Lecture Notes in Business Information Processing. Springer, 2022, pp. 243–273. DOI: [10.1007/978-3-031-08848-3_8](https://doi.org/10.1007/978-3-031-08848-3_8).
- [19] Rafael Lorenz, Julian Senoner, Wilfried Sihm, and Torbjørn H. Netland. “Using process mining to improve productivity in make-to-stock manufacturing”. In: *Int. J. Prod. Res.* 59.16 (2021), pp. 4869–4880. DOI: [10.1080/00207543.2021.1906460](https://doi.org/10.1080/00207543.2021.1906460).
- [20] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini. “Predictive Monitoring of Business Processes”. In: *CAiSE*. Vol. 8484. Lecture Notes in Computer Science. Springer, 2014, pp. 457–472. DOI: [10.1007/978-3-319-07881-6_31](https://doi.org/10.1007/978-3-319-07881-6_31).
- [21] Ronny Mans, Wil M. P. van der Aalst, and Rob J. B. Vanwersch. *Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes*. Springer Briefs in Business Process Management. Springer, 2015. ISBN: 978-3-319-16070-2. DOI: [10.1007/978-3-319-16071-9](https://doi.org/10.1007/978-3-319-16071-9).
- [22] Gyunam Park, Jan Niklas Adams, and Wil M. P. van der Aalst. “OPerA: Object-Centric Performance Analysis”. In: *(PREPRINT. Accepted for publication at Conceptual Modeling (ER) Conference 2022, Springer), CoRR* abs/2204.10662 (2022). arXiv: [2204.10662](https://arxiv.org/abs/2204.10662).
- [23] Adrian Rebmann, Jana-Rebecca Rehse, and Han van der Aa. “Uncovering Object-Centric Data in Classical Event Logs for the Automated Transformation from XES to OCEL”. In: *BPM*. Springer, 2022, pp. 379–396. DOI: [10.1007/978-3-031-16103-2_25](https://doi.org/10.1007/978-3-031-16103-2_25).
- [24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The Graph Neural Network Model”. In: *IEEE Trans. Neural Networks* 20.1 (2009), pp. 61–80. DOI: [10.1109/TNN.2008.2005605](https://doi.org/10.1109/TNN.2008.2005605).
- [25] Günther Schuh, Andreas Gützlaß, Seth Schmitz, and Wil M.P. van der Aalst. “Data-based description of process performance in end-to-end order processing”. In: *CIRP Annals* 69.1 (2020), pp. 381–384. DOI: <https://doi.org/10.1016/j.cirp.2020.03.013>.
- [26] Philipp Waibel, Lukas Pfahlsberger, Kate Revoredo, and Jan Mendling. “Causal Process Mining from Relational Databases with Domain Knowledge”. In: *CoRR* abs/2202.08314 (2022). arXiv: [2202.08314](https://arxiv.org/abs/2202.08314).
- [27] Jochen De Weerd, Annelies Schupp, An Vanderloock, and Bart Baesens. “Process Mining for the multifaceted analysis of business processes - A case study in a financial services organization”. In: *Comput. Ind.* 64.1 (2013), pp. 57–67. DOI: [10.1016/j.compind.2012.09.010](https://doi.org/10.1016/j.compind.2012.09.010).