Software Impacts 14 (2022) 100438

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

ocpa: A Python library for object-centric process analysis

Jan Niklas Adams *, Gyunam Park, Wil M.P. van der Aalst

Process and Data Science, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany

ARTICLE INFO

Keywords:
Process mining
Object-centric process mining
Object-centric petri net
Predictive process monitoring
Conformance checking
Performance analysis

ABSTRACT

ocpa is a Python library supporting object-centric process mining. Traditional process mining generates 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.

Code metadata

Current code version	1.0.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-211
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/7687700/tree/v1
Legal Code License	GPL-3.0
Code versioning system used	git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	pandas, numpy, pm4py, networkx, graphviz
If available Link to developer documentation/manual	https://ocpa.readthedocs.io/
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 [1]. **(2)** Conformance Checking: Testing whether the event sequences conform to given rules or exhibit deviations [2]. **(3)** Process Enhancement: Equipping a process model with additional information about performance, decisions, or resources [3]. **(4)** Predictive Process Monitoring: Extracting features from the event log and learning predictive models for different targets [4].

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 [5]. 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* [6–8].

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: niklas.adams@pads.rwth-aachen.de (J.N. Adams), gnpark@pads.rwth-aachen.de (G. Park), wvdaalst@pads.rwth-aachen.de (W.M.P. van der Aalst).

https://doi.org/10.1016/j.simpa.2022.100438
Received 26 September 2022; Received in revised form 15 October 2022; Accepted 26 October 2022

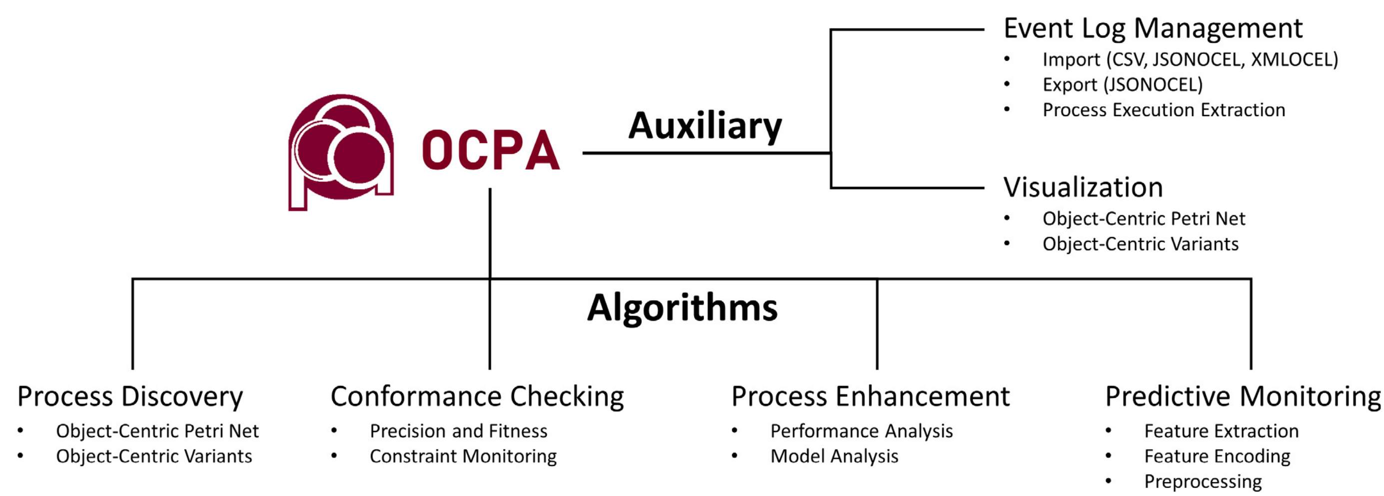

Fig. 1. Overview for the functionalities contained in OCPA.

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 [9], object-centric conformance checking [10], object-centric performance analysis [11], and object-centric predictive process monitoring [12]. So far, no unified platform for object-centric process mining exists. OCPA bundles all the proposed techniques into one single comprehensive Python library (cf. Fig. 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* repository[1] and through the package installer *pip*.[2] 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*.[3] The documentation provides a set of examples for each algorithm. The core code framework is split into an event log management, algorithm, and visualization submodule.

2.1. Event log management submodule

The event log management submodule contains all functionality to store and access object-centric event data. We allow the import of object-centric event logs in one of three formats: CSV, JSONOCEL, and XMLOCEL[4]. When importing a CSV file, information on the column 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 to intercompatability 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 [9]. Second, the user can discover control-flow variants following the algorithm of Adams et al. [13]. 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 [10]. 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 [14]. Such constraints include control-flow constraints (e.g., activity *place order* is directly followed by *send invoice* for an object *order*), object involvement constraints (e.g., the execution of activity *clear invoice* should involve an order along with an invoice and goods receipt), and performance constraints (e.g., the synchronization time of *clear invoice*, i.e., the time 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 the model-based approach presented in [11].

[1] https://github.com/ocpm/ocpa
[2] https://pypi.org/project/pip/
[3] https://ocpa.readthedocs.io
[4] Both have been defined in the OCEL standard www.ocel-standard.org.

The approach discovers an object-centric Petri net from an object-centric event log, replays the log in the model, and 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 [12]. Features are extracted based on the graph structure of the object-centric event log, i.e., the feature values accurately depict reality (for example, 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 [15] (sequences), or graph neural networks [16] (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.[5] Next to the layouting through GraphViz, OPCA 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 [13]. 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 [6,17], these contributions cover the discovery of object-centric Petri nets [9] and variants [13,18], object-centric evaluation metrics [10] and constraint monitoring [14], object-centric performance analysis [11], and object-centric feature extraction and encoding [12]. 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 [5,7,8] 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 [19,20].

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 conformance checking techniques?

OCPA itself has already functioned as a foundation for the development of several tools, e.g., OCπ [18], OPerA [11], and DTween [21]. These tools provide generalized use cases for analyzing any object-centric process, either w.r.t. frequent executions, performance issues, or reacting to violated constraints. Therefore, OCPA's role in object-centric process mining is similar to PM4Py's [22] role in traditional process mining: Both function as a basis for tool developments (e.g., PMTK[6] is developed using PM4Py) and enabling the quick implementation of new algorithms in an existing environment.

Impact of OCPA to applied process mining. Process mining has been applied to analyzing business processes in various industry segments [23]: *Finance* (e.g., banks and insurance companies) [24], *Manufacturing* (e.g., pharmaceuticals and automobile companies) [25], and *Service* (e.g., healthcare, government, and telecommunication) [26]. 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 [27]. 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 [28]. Furthermore, object-centric process mining can be applied to production processes where different production components are represented by objects [29,30]. Leading vendors already prepare support for object-centric event data, e.g., Celonis (with the recently introduced ProcessSphere) 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.

Funding

We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research (grant no. 1191945).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, Allar Soo, Automated discovery of process models from event logs: Review and benchmark, IEEE Trans. Knowl. Data Eng. 31 (4) (2019) 686–705, http://dx.doi.org/10.1109/TKDE.2018.2841877.

[2] Josep Carmona, Boudewijn F. van Dongen, Andreas Solti, Matthias Weidlich, Conformance Checking - Relating Processes and Models, Springer, ISBN: 978-3-319-99413-0, 2018, http://dx.doi.org/10.1007/978-3-319-99414-7.

[3] Massimiliano de Leoni, Foundations of process enhancement, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, vol. 448, Springer, 2022, pp. 243–273, http://dx.doi.org/10.1007/978-3-031-08848-3_8.

[4] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, Chiara Ghidini, Predictive monitoring of business processes, in: CAiSE, in: Lecture Notes in Computer Science, vol. 8484, Springer, 2014, pp. 457–472, http://dx.doi.org/10.1007/978-3-319-07881-6_31.

[5] https://graphviz.org/

[6] https://pmtk.fit.fraunhofer.de/

[5] Dirk Fahland, Process mining over multiple behavioral dimensions with event knowledge graphs, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, vol. 448, Springer, 2022, pp. 274–319, http://dx.doi.org/10.1007/978-3-031-08848-3_9.

[6] 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, http://dx.doi.org/10.1007/978-3-030-30446-1_1.

[7] Stefan Esser, Dirk Fahland, Multi-dimensional event data in graph databases, J. Data Semant. 10 (1) (2021) 109–141, http://dx.doi.org/10.1007/s13740-021-00122-1.

[8] Philipp Waibel, Lukas Pfahlsberger, Kate Revoredo, Jan Mendling, Causal process mining from relational databases with domain knowledge, 2022, CoRR abs/2202.08314, arXiv:2202.08314.

[9] Wil M.P. van der Aalst, Alessandro Berti, Discovering object-centric Petri nets, Fundam. Informaticae 175 (1–4) (2020) 1–40, http://dx.doi.org/10.3233/FI-2020-1946.

[10] Jan Niklas Adams, Wil M.P. van der Aalst, Precision and fitness in object-centric process mining, in: ICPM, IEEE, 2021, pp. 128–135, http://dx.doi.org/10.1109/ICPM53251.2021.9576886.

[11] Gyunam Park, Jan Niklas Adams, Wil M.P. van der Aalst, OPerA: Object-centric performance analysis, in: ER, Springer, 2022, pp. 281–292, http://dx.doi.org/10.1007/978-3-031-17995-2_20.

[12] Jan Niklas Adams, Gyunam Park, Sergej Levich, Daniel Schuster, Wil M.P. van der Aalst, A framework for extracting and encoding features from object-centric event data, CoRR abs/2209.01219 (2022) http://dx.doi.org/10.48550/arXiv.2209.01219.

[13] Jan Niklas Adams, Daniel Schuster, Seth Schmitz, Günther Schuh, Wil M.P. van der Aalst, Defining cases and variants for object-centric event data, CoRR abs/2208.03235 (2022) http://dx.doi.org/10.48550/arXiv.2208.03235.

[14] Gyunam Park, Wil M.P. van der Aalst, Monitoring constraints in business processes using object-centric constraint graphs, CoRR (2022) http://dx.doi.org/10.48550/arXiv.2210.12080.

[15] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[16] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80, http://dx.doi.org/10.1109/TNN.2008.2005605.

[17] Anahita Farhang Ghahfarokhi, Gyunam Park, Alessandro Berti, Wil M.P. van der Aalst, OCEL: A standard for object-centric event logs, in: ADBIS 2021 Short Papers, Doctoral Consortium and Workshop, Vol. 1450, in: Communications in Computer and Information Science, Springer, 2021, pp. 169–175, http://dx.doi.org/10.1007/978-3-030-85082-1_16.

[18] Jan Niklas Adams, Wil M.P. van der Aalst, OCπ: Object-centric process insights, in: PETRI NETS, Springer, 2022, pp. 139–150, http://dx.doi.org/10.1007/978-3-031-06653-5_8.

[19] Adrian Rebmann, Jana-Rebecca Rehse, 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, http://dx.doi.org/10.1007/978-3-031-16103-2_25.

[20] Alessandro Berti, Gyunam Park, Majid Rafiei, 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, http://dx.doi.org/10.1007/978-3-030-98581-3_19.

[21] Gyunam Park, Wil M.P. van der Aalst, Realizing a digital twin of an organization using action-oriented process mining, in: ICPM, IEEE, 2021, pp. 104–111, http://dx.doi.org/10.1109/ICPM53251.2021.9576846.

[22] Alessandro Berti, Sebastiaan J. van Zelst, Wil M.P. van der Aalst, Process mining for python (PM4Py): Bridging the gap between process- and data science, 2019, CoRR abs/1905.06169, arXiv:1905.06169.

[23] Rafael Accorsi, Julian Lebherz, A practitioner's view on process mining adoption, event log engineering and data challenges, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, vol. 448, Springer, 2022, pp. 212–240, http://dx.doi.org/10.1007/978-3-031-08848-3_7.

[24] Jochen De Weerdt, Annelies Schupp, An Vanderloock, Bart Baesens, Process mining for the multi-faceted analysis of business processes - A case study in a financial services organization, Comput. Ind. 64 (1) (2013) 57–67, http://dx.doi.org/10.1016/j.compind.2012.09.010.

[25] Mahendrawathi ER, Noval Arsad, Hanim Maria Astuti, Renny Pradina Kusumawardani, Rivia Atmajaningtyas Utami, Analysis of production planning in a global manufacturing company with process mining, J. Enterp. Inf. Manag. 31 (2) (2018) 317–337, http://dx.doi.org/10.1108/JEIM-01-2017-0003.

[26] Ronny Mans, Wil M.P. van der Aalst, Rob J.B. Vanwersch, Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes, in: Springer Briefs in Business Process Management, Springer, ISBN: 978-3-319-16070-2, 2015, http://dx.doi.org/10.1007/978-3-319-16071-9.

[27] David Cohn, Richard Hull, Business artifacts: A data-centric approach to modeling business operations and processes, IEEE Data Eng. Bull. 32 (3) (2009) 3–9.

[28] Günther Schuh, Andreas Gützlaff, Seth Schmitz, Wil M.P. van der Aalst, Data-based description of process performance in end-to-end order processing, CIRP Ann. 69 (1) (2020) 381–384, http://dx.doi.org/10.1016/j.cirp.2020.03.013.

[29] Tobias Brockhoff, Merih Seran Uysal, Isabelle Terrier, Heiko Göhner, Wil M.P. van der Aalst, Analyzing multi-level BOM-structured event data, in: ICPM Workshops, Vol. 433, Springer, 2021, pp. 47–59, http://dx.doi.org/10.1007/978-3-030-98581-3_4.

[30] Rafael Lorenz, Julian Senoner, Wilfried Sihn, Torbjørn H. Netland, Using process mining to improve productivity in make-to-stock manufacturing, Int. J. Prod. Res. 59 (16) (2021) 4869–4880, http://dx.doi.org/10.1080/00207543.2021.1906460.