PM4Moodle: Enabling Object-Centric Educational Process Mining

Najmeh Miri[1,*,†], Jelena Zdravkovic[1,*,†] and Amin Jalali[1,*,†]

[1]Department of Computer and Systems Sciences, Borgarfjordsgatan 8, Stockholm, 164 25, Stockholm University

Abstract

Process mining is gradually moving from case-centric event logs to object-centric representations that capture multiple interacting objects within a single process, thereby enabling simultaneous analysis from multiple perspectives. Yet, extracting such data from operational information systems remains difficult. This challenge becomes particularly relevant in educational settings, where learning activities are shaped by interactions among learners, teachers, groups, resources, and assessment objects. We introduce PM4Moodle, an open-source tool that extracts and transforms Moodle data into OCEL 2.0 logs, enabling object-centric educational process mining (OC-EPM). This demo paper presents the tool and demonstrates how it supports the object-centric educational event data extraction.

Keywords

Object-centric process mining, Object-centric event data, Object-centric educational event data, Learning analytics

1. Introduction

Learning Management Systems (LMSs) have become central platforms for organizing course material, communication, assessment, collaboration, and feedback. They also record detailed traces of how teachers and students interact with course resources and activities [1]. These traces make LMSs not only delivery platforms, but also potential data sources for understanding how educational processes unfold in practice. Moodle is a commonly used open-source LMS, providing a broad set of educational features through its resource and activity modules [1, 2]. Educational Process Mining (EPM) builds on such LMS-generated event data to analyze learning paths, behavior patterns, and interactions with course activities [3, 4].

Existing studies show that LMS logs can be used to investigate students' behavior and support educators in reflecting on learning processes, adapting teaching and learning activities, and identifying interventions to improve student learning outcomes [3, 5]. However, applying process mining to educational data is not straightforward. LMS data are stored in platform-specific structures, and turning these records into analyzable event logs often requires technical preparation that is difficult for educators [4]. This creates a gap between the operational environment, where Moodle records educational activity, and the analytical environment, where data must be structured for process mining.

This preparation problem is reinforced by a more fundamental limitation in traditional process mining. Most process mining techniques rely on case-centric event logs, where each event is assigned to one case notion. This assumption is often too restrictive for real information systems, where events may involve several related entities. For example, as illustrated in Figure 1, a teacher may want to understand how groups of students work within a course, how they interact with the materials, how members contribute to assignments and how these patterns relate to their performance. Such questions cannot be represented well by selecting only one case notion, such as a student or group. Moreover, flattening such data into a single case view can duplicate events, and produce misleading process models [6].

Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration Resources Forum (BPM 2026), Toronto, Canada, September 27th to October 2nd, 2026.

*Corresponding author.

†These authors contributed equally.

najmeh.miri@dsv.su.se (N. Miri); jelenaz@dsv.su.se (J. Zdravkovic); aj@dsv.su.se (A. Jalali)

0009-0008-8069-4159 (N. Miri); 0000-0002-0870-0330 (J. Zdravkovic); 0000-0002-6633-8587 (A. Jalali)

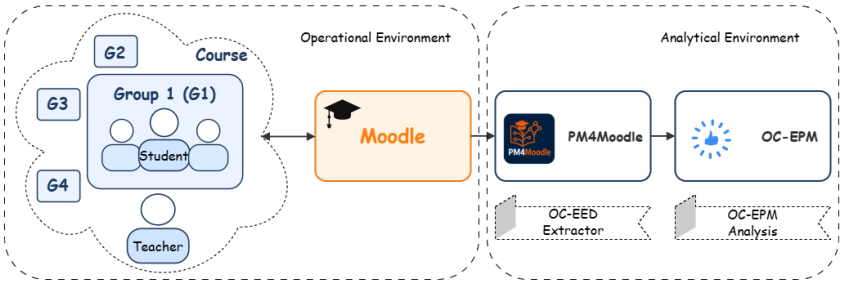


Figure 1: Conceptual Overview of PM4Moodle for Object-Centric Educational Process Mining.

Object-centric process mining (OCPM) addresses this limitation by representing processes through events, objects, and their relationships. This allows multiple case notions to be captured within the same event-data representation [7, 8]. OCEL 2.0 supports this paradigm by providing a standard format for recording and exchanging object-centric event logs across tools and systems [8]. However, the existence of such a standard does not by itself solve the extraction problem. For object-centric EPM (OC-EPM), there is still a need for tools that can manage the extraction complexity and turn educational data into object-centric event logs while preserving the objects, events and relations needed for multi-perspective analysis [2, 5].

We introduce PM4Moodle, an open-source tool for generating OCEL 2.0 logs from Moodle databases. As shown in Figure 1, PM4Moodle connects the operational environment of Moodle with the analytical environment of OC-EPM, enabling analysis of Moodle-based educational processes from multiple perspectives, including students, groups, courses, modules, submissions, and resources. This demo paper presents the design and functionality of the PM4Moodle tool. The demo shows how users can configure log extraction from a Moodle database, generate and inspect an OCEL-compliant event log, and derive an object-centric directly-follows graph (OC-DFG) for analysis. Overall, PM4Moodle is a reusable tool for making object-centric educational event data (OC-EED) available for OC-EPM.

2. Design Choices

PM4Moodle is designed to support four key features: *(i) extensible and systematic extraction*, *(ii) standardized and interoperable output representation*, *(iii) temporal modeling of educational relations*, and *(iv) support for multi-granularity analysis*. Together, these design choices ensure that the tool is adaptable to diverse Moodle settings while enabling reusable and expressive process analysis across different perspectives.

Extensible and systematic extraction. Although the current implementation targets Moodle, its mapping approach follows the system-agnostic OCPM^2 methodology, which structures the identification of object types, activities, their relationships, extraction logic, and verification steps [2]. The same approach can therefore be adapted to other LMSs and application domains. In PM4Moodle, this methodology is operationalized by mapping Moodle entities and modules to object types, and logged activities to event types. A conceptual model and extraction matrices then define the object-to-object and event-to-object relations and their expected cardinalities, as detailed in [2, 5]. This design allows new Moodle modules to be incorporated by specifying their relevant objects, events, and relations without redesigning the extraction pipeline. As a result, the approach supports incremental extension while maintaining consistency and correctness of the extracted data.

Standardized and interoperable output representation. PM4Moodle adopts OCEL 2.0 [8] as its core data representation to ensure compatibility with existing object-centric process mining tools. Using a standardized format enables the extracted data to be directly reused across multiple analysis environments and avoids tool-specific dependencies. In addition, OCEL data can be transformed into other object-centric representations, such as Event Knowledge Graphs (EKGs), where events, objects, and

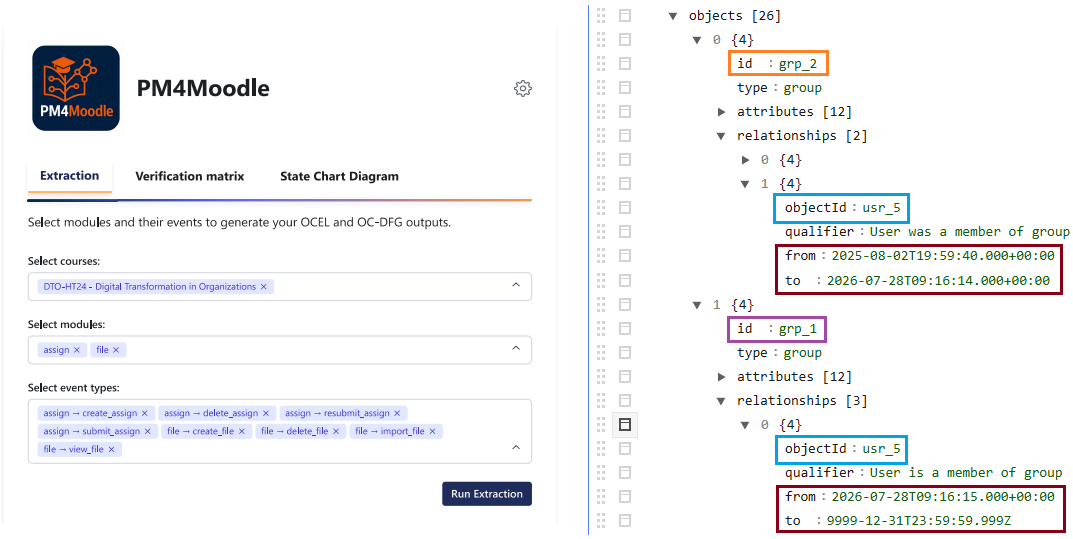
their relations are modeled as interconnected entities [9]. This design promotes interoperability and allows the same extracted data to support different analysis techniques without requiring re-extraction.

Temporal modeling of educational relations. Educational processes inherently involve time-dependent relationships. In Moodle, object-to-object relations may only hold during specific periods, for instance when students change groups or leave a course before completion. To capture such dynamics, PM4Moodle augments the OCEL representation by associating object relations with temporal attributes. Figure 2(b) shows how such temporal relations are represented in the generated OCEL 2.0 through the highlighted validity period timestamps. This design preserves the validity intervals of relationships and enables analyses that consider their evolution over time [10]. Furthermore, it supports transformation into temporal Event Knowledge Graphs (tEKGs) [11], facilitating richer temporal reasoning over educational processes.

Support for multi-granularity analysis. PM4Moodle enables analysis at different levels of abstraction, ranging from high-level views such as courses or modules to fine-grained views including assignments, resources, users, and groups. By producing OCEL 2.0 logs that encode inheritance information for relevant object types, the tool enables OLAP-style operations to be applied on the extracted logs, such as drill-down, roll-up, unfold, and fold [12]. These operations allow analysts to navigate between aggregated and detailed perspectives and to uncover patterns that may remain hidden when considering only a single level of granularity [13, 14].

3. Implementation

PM4Moodle is implemented as a web-based tool with a React/TypeScript frontend and a Python/Flask backend. The frontend supports user interaction for connecting to a Moodle database, configuring the extraction, inspecting verification outputs, and downloading the generated results. The backend provides the extraction and transformation logic. It connects to the Moodle MySQL database, retrieves selected records, maps them to events, objects, and relations, and exports the result as OCEL 2.0. The implementation uses `mysql-connector-python` and SQLAlchemy for database access, PM4Py for process mining operations, and Cytoscape for graph-based visualizations. Figure 2 illustrates the configurable extraction interface and an excerpt of the generated output showing the temporal representation of an object-to-object relation between user 5 (`usr_5`) and groups 2 (`grp_2`) and 1 (`grp_1`).



(a) (b)

Figure 2: PM4Moodle extraction interface (a) and an excerpt of the generated OCEL 2.0 showing the temporal relations of user 5 with groups 2 and 1 preserved by validity period timestamps (b).

3.1. Features

PM4Moodle supports data extraction and validation through several features, including *database connection*, *OCEL 2.0 extraction*, *downloadable outputs*, *verification matrices*, and *state chart diagrams*.

Database connection: Users can connect to a Moodle database by providing their Moodle credentials.

Extraction scope: After establishing a connection, users define the extraction scope by selecting courses, modules, and event types, or perform a full extraction of all supported data.

Downloadable outputs: The tool produces an OCEL 2.0 event log which can be downloaded for further analysis. Also, it generates an object-centric directly-follows graph (OC-DFG) as early feedback on the overall process.

Verification matrices: PM4Moodle provides a *frequency matrix*, showing counts of event–object relations, and a *cardinality matrix*, showing their minimum and maximum occurrences. Both support filtering by event and object types.

State chart diagrams: State charts are generated from observed event transitions, enabling comparison of expected and actual behavior (e.g., *create*, *update*, *view*, *delete*).

3.2. Resources

We implemented PM4Moodle as an open-source tool together with documentation, setup instructions, and usage examples which are publicly available on https://github.com/MiriNajme/PM4Moodle. A short screencast demonstrating the main features and extraction workflow of the tool is available at https://youtu.be/AHblrPF_owU. To support testing and reproducibility, the repository includes `backup.sql`, a Moodle database backup containing two simulated courses, together with a usage guide at https://github.com/MiriNajme/PM4Moodle/blob/main/TEST_DATASET.md. The guide explains how to restore the backup locally, connect PM4Moodle to the restored database, run the extraction, and validate the generated outputs. To reduce setup effort, we also provide a lightweight Docker demo: a single command launches PM4Moodle and a database preloaded with the test dataset, requiring only Docker. An optional Moodle instance can be launched against the same database, allowing course changes to be reflected in subsequent extractions.

3.3. Maturity

PM4Moodle has evolved from an initial code-driven implementation for demonstrating the applicability of OCPM^2 methodology [2, 5] . The current version operationalizes the extraction logic through an interactive interface that supports database connection, configurable extraction, output generation, and verification. The tool has also matured in terms of extraction coverage. While the initial version supported a limited set of Moodle modules, the current version covers nine common activity and resource modules: *assignment*, *forum*, *quiz*, *choice*, *file*, *folder*, *page*, *URL*, and *text and media area*. Since PM4Moodle is open source, it can also serve as a demonstrable tool that other researchers can run, inspect, and extend. The tool was evaluated with real educational data in two case studies. In [5], PM4Moodle extracted a log from a real university course; the resulting analysis linked repeated responsibility for group submissions with stronger academic outcomes, while simple counts of learning-material access did not explain exam performance. In [13], PM4Moodle extracted four years of data from 401 students in 91 groups; The resulting log supported OLAP operations that distinguished user roles and assignments, improved model fitness and precision in most of 71 before-and-after comparisons.

4. Conclusion & Future Work

This demo paper presented PM4Moodle, an open-source tool for extracting object-centric event data from Moodle. Following the OCPM^2 methodology, the tool transforms Moodle database records into OCEL 2.0 logs and operationalizes object-centric educational process mining by capturing relations between multiple educational entities, such as users, courses, groups, and modules. The current version

is limited to a selected set of modules. Future work will extend the coverage to more Moodle modules and explore how PM4Moodle can contribute to the development of digital twins of educational processes by providing live object-centric event data that can serve as the basis for monitoring, analysis, and simulation of learning processes.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI GPT-5 and Grammarly to improve readability and check grammar. These tools were used only to refine the authors' own text. The authors reviewed and edited the content and take full responsibility for the publication.

References

[1] R. El-Bahsh, M. Daoud, Evaluating the use of moodle to achieve effective and interactive learning: A case study at the german jordanian university, in: Proceedings of the 35th annual IEEE international conference on computer communications, 2016, pp. 1–5.

[2] N. Miri, S. Khayatbashi, J. Zdravkovic, A. Jalali, Ocpm^2: Extending the process mining methodology for object-centric event data extraction, in: International Conference on Business Process Modeling, Development and Support, Springer, 2025, pp. 123–140.

[3] Y. Li, Q. Jiang, W. Xiong, W. Zhao, Investigating behavior patterns of students during online self-directed learning through process mining, Education and Information Technologies 28 (2023) 15765–15787.

[4] E. Real, E. Pimentel, An educational process mining model on students' paths data from virtual learning environments, Technology, Knowledge and Learning (2025). URL: https://doi.org/10.1007/s10758-025-09824-y. doi:10.1007/s10758-025-09824-y.

[5] N. Miri, S. Khayatbashi, J. Zdravkovic, A. Jalali, An object-centric approach to multi-entity educational process mining, Decision Analytics Journal 19 (2026) 100724. doi:10.1016/j.dajour.2026.100724.

[6] W. M. P. van der Aalst, Object-centric process mining: Dealing with divergence and convergence in event data, in: Software Engineering and Formal Methods, Springer, 2019, pp. 3–25.

[7] A. Seidel, et al., Object-centric process management: A research manifesto, Information Systems 141 (2026) 102728. URL: https://www.sciencedirect.com/science/article/pii/S0306437926000426. doi:https://doi.org/10.1016/j.is.2026.102728.

[8] 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, 2023.

[9] S. Khayatbashi, O. Hartig, A. Jalali, Transforming event knowledge graph to object-centric event logs: A comparative study for multi-dimensional process analysis, in: Conceptual Modeling (ER 2023), Springer, 2023, pp. 220–238.

[10] S. Khayatbashi, V. Sjölind, A. Granåker, A. Jalali, Ai-enhanced business process automation: a case study in the insurance domain using object-centric process mining, in: International Conference on Business Process Modeling, Development and Support, Springer, 2025, pp. 3–18.

[11] S. Khayatbashi, O. Hartig, A. Jalali, Transforming object-centric event logs to temporal event knowledge graphs, in: BPM Workshops, Springer, 2025, pp. 300–313.

[12] S. Khayatbashi, N. Miri, A. Jalali, Olap operations for object-centric process mining, in: International Conference on Advanced Information Systems Engineering, Springer, 2025, pp. 111–118.

[13] S. Khayatbashi, N. Miri, A. Jalali, Advancing object-centric process mining with multi-dimensional data operations, arXiv preprint arXiv:2412.00393 (2024).

[14] N. Miri, A. Jalali, Uncovering patterns in object-centric process mining: An approach using drill-down and roll-up techniques, in: Proc. iiWAS, Springer, 2025, pp. 49–54.