Process Mining by Design for Low-Code Application Platforms

Cassio Alexandre Pereira de Castro

University of Ottawa, Ottawa, Canada

Abstract

Process mining initiatives frequently face costly and complex post-hoc extraction of event logs, particularly when processes are implemented outside traditional process-aware information systems. Meanwhile, Low-Code Application Platforms (LCAPs) are increasingly used to extend enterprise systems, develop internal applications, and host process automation, including AI-driven execution. Nevertheless, LCAPs are typically not designed to generate process-mining-ready event data by default. This doctoral research introduces *Process Mining by Design* (PMbD) for LCAP environments, a design-time approach that embeds case identification, activity semantics, and timestamp recording directly into application and UX/UI design decisions. By moving log readiness from post-hoc reconstruction to a development-time concern, PMbD aims to reduce extraction effort and ease log generation in LCAP-based solutions. The research follows a Design Science Research methodology and will deliver the UI to Log Mapping (UI2LogM) method, reusable logging modules for Mendix and OutSystems, and UX/UI design guidelines supported by an LLM-based assistant, evaluated through real-world LCAP projects to assess feasibility, usability, and adoption.

Keywords

Process mining, Low-code application platforms, Process mining by design, UI to Log Mapping (UI2LogM), Event logs, User experience, Design science research,

1. Introduction and Background

Companies are constantly looking for ways to reduce operational costs, and applications such as Enterprise Resource Planning (ERP) systems are often used to achieve this goal. However, other technologies, such as Low-Code Application Platforms (LCAPs), are being used to extend ERP features, particularly those that are not standard and require customization. An LCAP is a software environment based on visual programming that enables both professional and citizen developers to build and maintain applications faster than with traditional development approaches, often with built-in process automation engines, and is widely adopted by small and medium enterprises [1, 2, 3].

Process mining techniques enable organizations to analyze and understand their processes by exploiting event logs generated by information systems. These techniques support process discovery, conformance checking, and process improvement, providing an essential bridge between process modelling, analysis, and data mining [4]. Data collection is one of the most expensive phases of process mining projects [5]. While Process-Aware Information Systems (PAIS) such as SAP, Microsoft, Oracle, and Salesforce provide ready-to-use connectors, substantial effort is required when processes are implemented in other applications, including those developed using LCAP, where event logs are unavailable or not readily usable.

Specifically, in LCAP environments, non-workflow applications provide no explicit activity semantics, and even workflow engine logs do not separate execution time from waiting time, which might lead to inconsistent analyses. Also, LCAP data models rarely offer a single case notion, and capturing better event data may require interface changes whose effect on the user experience has not been studied.

Additionally, LCAP platforms are becoming workbenches for Artificial Intelligence (AI) agents. With the rise of Agentic AI, platforms such as Mendix and OutSystems are integrating agent execution into their environments [6, 7], giving AI agents access to applications and workflows previously operated

Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum (BPM 2026) , Toronto, Canada, September 27th to October 2nd, 2026.
cpere103@uottawa.ca (C. A. P. d. Castro)

only by humans. Since agents may perform tasks that people typically do, these actions should include proper logging to record non-human execution and ensure that agent activity is captured in the same event format as human interactions.

The absence of ready-to-mine event logs has been addressed by several techniques, all operating after the application exists: extraction pipelines reconstruct logs from application data [5], task mining derives events from recorded user interactions [8], and reverse engineering generates logs from non-process-aware systems [9]. To the best of the author's knowledge, no approach prepares applications to be mined before development, embedding log existence and quality concerns into application design. This research addresses this gap, whatever log representation is adopted, such as object-centric [10] or translucent event logs [11]. However, preparing applications this way is not free: it demands additional design and development effort from teams that did not plan for it. Whether the resulting gains justify this effort is an open question, which this research evaluates in real development projects.

In summary, this work proposes a design-time approach that enables LCAP applications to be developed from a *process mining by design* (PMbD) perspective, regardless of whether tasks are executed by humans or by agents. The concept is analogous to the privacy-by-design approach [12], where best practices are integrated throughout development rather than added as an afterthought. PMbD does not aim to change the platforms themselves: it acts on the development process built on top of them, through design practices and modules published in the platform repositories.

2. Research Questions

- **RQ1:** *What resources are available on LCAP platforms to support process mining log generation and process mining by design?* LCAPs primarily provide audit-oriented logging when their workflow engines are used; non-workflow applications and the emerging agent-based software integrated into LCAP workflows require investigation of their logging features.
- **RQ2:** *What UX design practices should be applied to prevent a poor user experience when process mining by design is used?* Generating event logs by design may require adding or adjusting interface components, and these changes should not negatively affect the user.
- **RQ3:** *What resources should be offered to support LCAP developers and designers when using the process mining by design approach?* This research will investigate which resources could support LCAP developers and designers in enabling event log generation during development.

3. Contributions

This research is expected to deliver three main contributions, aligned with the research questions above.

3.1. The UI to Log Mapping (UI2LogM) method (RQ2 and RQ3)

Each "by design" paradigm relies on a structured method to translate principles into implementation decisions, as threat modelling does for Security by Design [13]. PMbD requires an analogous method. UI2LogM is a structured, pre-implementation activity in which the design team traverses each screen (or wireframe), identifies which user actions correspond to process mining events, classifies the required instrumentation as *silent* (server-side logging of an existing action, with no interface change), *opportunistic* (a UI element that serves both the user's workflow and PMbD's data needs), or *additive* (a new or modified UI element needed primarily for PMbD, creating potential UX tension), and documents the result as an event mapping table. Since the interface exposes, at each moment, the next actions enabled to the user, the mapping might also record the alternatives that were presented but not chosen, which would produce translucent event logs [11] by design. This possibility also illustrates why UI2LogM operates at the UI level rather than at a lower layer: the alternatives not chosen are only observable where they are presented to the user, while purely back-end recording is covered by the silent category. The event mapping table bridges the other contributions: it drives the logging requirements that the

PMbD modules must support and identifies the UI touchpoints where UX guidelines and LLM-based assistance are needed.

3.2. Assessment of event log generation and PMbD modules for LCAP and Agentic AI frameworks (RQ1 and RQ3)

Lichtenthäler et al. [14] report that most developers using LCAP platforms create process automation and business-centric solutions, often extending ERP functions. Additionally, with the rise of AI agents, LCAP platforms are establishing themselves as agent workbenches. This contribution includes an assessment of the current state of event log generation in LCAP environments, covering both workflow-based and non-workflow applications and how agent-driven actions can be logged alongside human-driven ones. Building on this assessment, the research will deliver reusable PMbD modules for Mendix and OutSystems that support the generation of process-mining-ready event logs during application implementation. These modules will be published in the respective platform repositories and will provide a web interface for defining the attributes to be logged, log export, and an API for on-line consumption.

3.3. UX/UI design guidelines and LLM-based support (RQ2)

While UI2LogM defines *what* to log and *where*, this contribution addresses *how* to design the affected interface points without degrading the user experience. The two are complementary: the additive touchpoints identified by UI2LogM are the input for the guidelines and for the LLM-based assistant. Zhou et al. [15] conducted an empirical case study of ChatGPT applied to UX design, identifying nine functions of the tool across UX activities, none of which addressed UX/UI adjustments to support the generation of process mining event logs. Since this work also aims to leverage LLMs while ensuring proper log collection, it will deliver design guidelines and an LLM-based assistant to support designing applications that log events without compromising user experience. Designers should assess whether the processes to be implemented with LCAP are suitable candidates for PMbD. Technical criteria follow van der Aalst [16]: an identifiable case notion, clear activity names, recorded timestamps, multiple events per case, repeatable behaviour, and complete cases. Business criteria follow the process selection recommendations of Dumas et al. [17], including strategic importance and feasibility of improvement. Initially, a good candidate should meet all technical criteria and at least three of the five business criteria.

4. Design Science Research Methodology

This research adopts the Design Science Research (DSR) methodology [18], which aims to address organizational problems by designing, developing, and evaluating artifacts. DSR is appropriate for this study, as the research objective is to introduce and operationalize a new paradigm, PMbD, and to evaluate its applicability and value through concrete artifacts applied in real-world LCAP projects. The research follows the five-phase DSR approach: (1) awareness of the problem, (2) suggestion, (3) development, (4) validation, and (5) conclusion.

4.1. Awareness of the Problem

The problem awareness phase investigates the limitations of current LCAP practices regarding process mining readiness, where the required event logs are incomplete, semantically weak, or absent. This phase also examines the increasing adoption of Agentic AI in task execution and its implications for process observability.

4.2. Suggestion

Based on the identified problem, this research proposes PMbD as a new paradigm to address limitations in process observability within LCAP environments. By embedding event semantics, case identification, and timing information into the design of LCAP applications, event logs suitable for process mining can be generated by default, reducing the need for later reconstruction and enabling more accurate analysis of processing and waiting times, regardless of whether humans or AI agents execute tasks.

4.3. Development

The development phase focuses on the design and implementation of PMbD-related artifacts and on defining how these artifacts will be introduced and used by different role groups (development roles, design/analysis/governance roles, and operational users) within low-code process automation projects. The primary artifacts to be developed are: (a) the **UI2LogM method**, a structured preparatory activity that connects the paradigm's principles to the artifacts that implement them; (b) **PMbD modules** for Mendix and OutSystems, supporting the generation of process-mining-ready event logs during application implementation and published in the platform module repositories; and (c) **UX/UI design guidelines and LLM-based support**, aimed at assisting design, analysis, and governance roles in applying PMbD without negatively impacting the user experience.

4.4. Validation

Validation will be conducted in real LCAP development projects organised as two iterative DSR cycles. Each iteration runs one project on OutSystems and one on Mendix, in parallel or in sequence; both must be completed to produce a consolidated list of improvements for the PMbD modules, UX/UI guidelines, and the UI2LogM method.

Projects are selected by duration (at most 12 weeks: four sprints plus two weeks of discovery and two of release and stabilization), the mandatory presence of AI-agent tasks, and preferably a different business process each. Improvements are implemented before the second iteration; after both cycles, the analysis retains the must-have improvements raised in both projects.

4.4.1. Role-Based Perspective on Evaluation.

Feedback will be collected from the three role groups involved during development. Development roles will provide feedback primarily through surveys focusing on implementation effort, technical feasibility, and the perceived impact of PMbD modules. Design, analysis, and governance roles will provide feedback through non-role-aware surveys and contextual inputs from sprint retrospectives. Operational users will provide feedback through in-context immediate micro-feedback and delayed, aleatory feedback mechanisms, focusing on usability,clarity, trust, and perceived system behaviour during application use.

4.4.2. Evaluation and Data Collection.

Validation will examine PMbD adoption in real projects, avoiding simulated scenarios, with data collection embedded into existing development practices. *Lightweight and anonymous surveys* will be applied repeatedly throughout the lifecycle to capture how PMbD perceptions evolve. To preserve anonymity, surveys will not be role-aware; respondents will rate each answer by perceived criticality on a 1-to-5 scale (1 most critical, 5 least). *Sprint retrospectives* will add qualitative input on adoption, friction, and perceived value; given small teams and role overlap, they will be reported as anonymized, aggregated thematic summaries without speaker attribution. Recordings will be transcribed and, if needed, translated to English with LLM services, while all coding is performed manually by two independent researchers, who code a subset (20–30%), assess inter-rater reliability, and resolve discrepancies by consensus. Must-have enhancements raised by more than two respondents will trigger their implementation. Generated event logs quality will be assessed relative to their intended use (process discovery) by checking case, activity, and timestamp completeness against the criteria Section 3.3.

4.4.3. End-User Feedback and Operational Perception.

End-user evaluation targets the operational impact of PMbD-enabled applications, not the concept itself. As end users are unaware of the underlying logging, feedback is grounded in the UTAUT model [19], focusing on effort expectancy (perceived usability, clarity, and cognitive load) and trust. Two complementary, anonymous, non-intrusive mechanisms will be used: in-context feedback triggered after selected interactions, and delayed, aleatory feedback after random usage instances. Responses use a 1-to-10 scale and, following the Net Promoter Score (NPS) methodology, questions averaging 6 or below enter the improvement list.

4.4.4. Researcher Positioning and Anonymity.

Given the organizational context in which the researcher operates, full anonymity of qualitative discussions cannot be guaranteed. To address this, the study will avoid individual attribution, rely on role-based aggregation, and separate data collection from project facilitation. The researcher will not participate in sprint retrospectives and will not have access to raw discussion records (information will be aggregated and provided by the process owner).

4.5. Conclusion Phase and Iteration

The conclusion phase will consolidate findings from all validation cycles to assess the overall value of PMbD and to identify opportunities for improvement. Insights synthesized across the two iterative DSR cycles will be used to reflect on the effectiveness, limitations, and applicability of the proposed artifacts, ensuring both practical relevance and methodological rigour.

4.6. Research Threats

LCAP applications are evolving rapidly, not only toward becoming AI-agent platforms but also toward offering AI-assisted development, possibly changing how developers work and impact the outcomes proposed in this research, such as the PMbD modules.

Beyond platform evolution, the study faces the four standard categories of validity threats. Generalizability is constrained by the external scope (only Mendix and OutSystems, web-based, and structured case-centric processes) and a small number of projects per iteration. Internal-validity risks arising from the researcher's professional ties to the host organizations are mitigated by keeping the researcher out of sprint retrospectives and raw records. For construct and reliability, event-log quality will be checked with at least two process mining tools, perception ratings will be supplemented with objective metrics where feasible, and the LLM versions, prompts, and inter-rater coding agreement will be documented.

5. Conclusion

This thesis introduces process mining by design, which advocates that applications should be developed with event log generation as a core feature, similar to the privacy-by-design approach. As LCAPs gain popularity among enterprises, their flexibility and broad range of applications make pre-built connectors infeasible, and PMbD offers an alternative to costly post-hoc extraction. It also paves the way for further research, such as extending the approach to other development platforms as generative AI advances. The research is at the proposal stage, with the defense scheduled for August 2026 and 19 months of remaining work covering artifact development, two validation cycles, and writing.

Declaration on Generative AI

During the preparation of this work, the author used Claude Opus 4.8 in order to: Grammar and spelling check, paraphrase and reword. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] M. Overeem, S. Jansen, M. Mathijssen, API management maturity of low-code development platforms, in: Web Engineering (ICWE 2021), volume 12706 of *Lecture Notes in Computer Science*, Springer, Cham, 2021, pp. 380–387. doi:10.1007/978-3-030-79186-5_25.

[2] A. Sahay, A. Indamutsa, D. Di Ruscio, A. Pierantonio, Supporting the understanding and comparison of low-code development platforms, in: Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2020), IEEE, 2020, pp. 171–178. doi:10.1109/SEAA51224.2020.00036.

[3] L. Bies, M. Weber, T. Greff, D. Werth, A mixed-methods study of low-code development platforms: Drivers of digital innovation in SMEs, in: Proceedings of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2022), IEEE, 2022, pp. 1–6. doi:10.1109/ICECCME55909.2022.9987920.

[4] W. M. P. van der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, et al., Process mining manifesto, in: Business Process Management Workshops (BPM 2011), volume 99 of *Lecture Notes in Business Information Processing*, Springer, Berlin, Heidelberg, 2012, pp. 169–194. doi:10.1007/978-3-642-28108-2_19.

[5] A. Berti, G. Park, M. Rafiei, W. M. P. van der Aalst, An event data extraction approach from SAP ERP for process mining, in: Process Mining Workshops (ICPM 2021), volume 433 of *Lecture Notes in Business Information Processing*, Springer, 2022, pp. 255–267. doi:10.1007/978-3-030-98581-3_19.

[6] Mendix, Mendix 11: Agents kit and agent builder for agentic AI workflows, Mendix Press Release, 2025. URL: https://www.mendix.com/press/mendix-launches-new-release-to-meet-surging-demand-for-agentic-ai/.

[7] OutSystems, Agent workbench: End-to-end low-code environment for building enterprise AI agents, OutSystems Press Release, 2025. URL: https://www.outsystems.com/news/outsystems-agent-workbench-ai/.

[8] V. Leno, A. Polyvyanyy, M. Dumas, M. La Rosa, F. M. Maggi, Robotic process mining: Vision and challenges, Business & Information Systems Engineering 63 (2021) 301–314. doi:10.1007/s12599-020-00641-4.

[9] R. Pérez-Castillo, B. Weber, J. Pinggera, S. Zugal, I. García-Rodríguez de Guzmán, M. Piattini, Generating event logs from non-process-aware systems enabling business process mining, Enterprise Information Systems 5 (2011) 301–335. doi:10.1080/17517575.2011.587545.

[10] A. Berti, I. Koren, J. N. Adams, G. Park, B. Knopp, N. Graves, M. Rafiei, L. Liß, L. Tacke Genannt Unterberg, Y. Zhang, C. Schwanen, M. Pegoraro, W. M. P. van der Aalst, OCEL (Object-Centric Event Log) 2.0 specification, arXiv:2403.01975, 2024. URL: https://arxiv.org/abs/2403.01975.

[11] W. M. P. van der Aalst, Lucent process models and translucent event logs, Fundamenta Informaticae 169 (2019) 151–177. doi:10.3233/FI-2019-1842.

[12] A. Cavoukian, Privacy by Design: The 7 Foundational Principles. Implementation and Mapping of Fair Information Practices, Technical Report, Information and Privacy Commissioner of Ontario, Toronto, Canada, 2009. URL: https://www.ipc.on.ca/en/media/1826/download?attachment, revised January 2011.

[13] C. Del-Real, E. De Busser, B. van den Berg, A systematic literature review on security by design and privacy by design: Identifying commonalities and differences, International Review of Law, Computers & Technology 39 (2025). doi:10.1080/13600869.2025.2457227.

[14] R. Lichtenthäler, S. Böhm, J. Manner, S. Winzinger, A use case-based investigation of low-code development platforms, in: Proceedings of the 14th Central European Workshop on Services and their Composition (ZEUS 2022), Bamberg, Germany, 2022, pp. 76–83. URL: https://ceur-ws.org/Vol-3113/paper12.pdf.

[15] Z. Zhou, Y. Li, J. Yu, Exploring the application of LLM-based AI in UX design: An empirical case study of ChatGPT, Human-Computer Interaction (2024). doi:10.1080/07370024.2024.2420991.

[16] W. M. P. van der Aalst, Process Mining: Data Science in Action, 2nd ed., Springer, Berlin, Heidelberg,

2016. doi:10.1007/978-3-662-49851-4.

[17] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, 2nd ed., Springer, Berlin, Heidelberg, 2018. doi:10.1007/978-3-662-56509-4.

[18] V. K. Vaishnavi, W. Kuechler, Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 2nd ed., CRC Press, Boca Raton, FL, 2015. doi:10.1201/b18448.

[19] V. Venkatesh, M. G. Morris, G. B. Davis, F. D. Davis, User acceptance of information technology: Toward a unified view, MIS Quarterly 27 (2003) 425–478. doi:10.2307/30036540.