A Conceptual Model to Enhance Process Mining Analysis with Provenance

Álvaro Bernal-Caunedo(✉)[1], Marco Franceschetti[1], Hagen Völzer[1], Manuel Resinas[2], and Barbara Weber[1]

[1] University of St. Gallen, St. Gallen, Switzerland
{alvaro.bernalcaunedo, marco.franceschetti, hagen.voelzer, barbara.weber}@unisg.ch
[2] Universidad de Sevilla, Seville, Spain
resinas@us.es

Abstract. Process mining relies on event data to analyze and improve business processes. The exploratory and non-linear nature of these analyses makes it difficult to understand, validate, compare, and communicate how findings are produced, limiting their business value. Existing provenance techniques focus on data lineage but lack support for systematic comparison of analyses, modular reuse of analysis steps, and comprehensive execution tracking. To address these limitations, we systematically derive a set of requirements and acceptance criteria based on established analytic provenance purposes. We further contribute a structured comparison of state-of-the-art provenance models, revealing critical gaps in comparative capabilities, reusability, and execution tracking. Based on these insights, we propose a conceptual provenance model tailored to exploratory process mining that supports branching analysis paths, comparison of analysis histories and states, reusability of analysis steps, and transparent representation of data evolution.

Keywords: Exploratory Process Mining · Analytic Provenance · Conceptual Model

1 Introduction

Process mining is widely used to analyze and improve processes based on event data [10]. In practice, analyses are rarely linear, requiring analysts to iteratively filter event logs, adjust parameters, apply different techniques, and compare alternative results [14]. This exploratory nature makes it difficult to understand, validate, and communicate how analysis outcomes (e.g., business rule violations, temporal constraint deviations) are produced.

Provenance techniques address these challenges by capturing analysis history and supporting analytical needs such as recalling prior steps, replicating results, exploring alternative paths, and communicating findings [8]. Building on these established provenance purposes, this work derives requirements tailored to exploratory process mining. Acceptance criteria operationalize these requirements, ensuring they are conceptually grounded and practically verifiable.

Applying these criteria in a literature review shows that existing approaches focus primarily on basic data lineage and linear workflows. They exhibit critical gaps in four areas: comprehensive execution tracking (i.e., tracking the agent and execution environment of each operation), which limits reliable reproducibility, systematic comparison of alternatives, inspection of how intermediate data states evolve over time, and seamless reuse of analytical steps. Furthermore, branching support relies on informal mechanisms [2,7,4,11].

Based on these insights, we propose a conceptual provenance model tailored to exploratory process mining. The model explicitly supports branching analysis paths, systematic comparison of analysis histories and states, reuse of analysis steps, and transparent representation of data evolution. In doing so, it addresses the limitations of existing approaches and enables more effective exploration, validation, and communication of analysis processes.

The paper is structured as follows. Sect. 2 introduces core concepts like analytic provenance and its purposes, and formalizes the requirements, mapping them to provenance purposes and establishing acceptance criteria to compare provenance models. Sect. 3 reviews literature to identify gaps based on these requirements. Sect. 4 introduces our provenance model and its creation methodology. Sect. 5 discusses limitations and concludes with future work.

2 Requirements and Acceptance Criteria

To systematically derive a provenance model that meets the needs identified above, we first ground our understanding in analytic provenance and its objectives. Analytic provenance adapts general provenance principles to explicitly capture the analysis history, recording the sequence of analysis steps and their corresponding findings during exploratory data analysis [8]. In such settings, analysts iteratively refine their approach, compare alternative paths, and examine how data evolves over time. These activities motivate the need to understand the objectives of analytic provenance. We therefore ground our approach in an established classification of six analytic provenance purposes [8] that apply broadly to data analysis rather than specifically to exploratory process mining: *recall* (recovering awareness of current and previous analysis states), *presentation* (communicating insights to external stakeholders), *meta-analysis* (reviewing the analytic process to improve it), *replication* (reproducing a previous analysis), *action recovery* (undo/redo and branching of actions), and *collaborative communication* (sharing the analysis with others). Their full definitions are provided in the online appendix [1]. Within these purposes, our approach is scoped to objective operational data, excluding the subjective facets such as free-text annotations detailing the analyst's reasoning, hypotheses, or intents.

Based on these purposes, we derive a set of requirements for a provenance model tailored to exploratory process mining. To ground them, we first define the core structural components of an analytical workflow. An *analysis state* captures a complete snapshot at a specific point (data objects, intermediate results, parameters, and context). Workflows transition between states via an *operation*

(the abstract definition of a task), whose concrete execution by an agent or system is recorded as an *analysis step*. The set of alternating states and steps forms an *analysis history* that may diverge into independent *branches* that share a common origin but maintain distinct subsequent states and dependencies.

To derive the requirements, we operationalize the mechanics necessary to fulfill each purpose in an exploratory environment. For example, action recovery (the user's need to undo actions and explore alternatives) is operationalized into *Traceability of Analysis Steps* and *Branching Capability*. Mapping these purposes to the workflow components yields the following core requirements:

R1: Traceability of Analysis Steps. The provenance model shall trace every step and state back to its originating data through the complete set of dependencies and the ordering of analysis steps, including all intermediate states, operations, parameter configurations and the agents or systems involved.

R2: Reproducibility. The provenance model shall capture all required components (input data versions, parameter configurations, and the execution environment) to enable the reliable re-execution of an analysis in identical conditions to obtain equivalent results.

R3: Branching Capability. The provenance model shall represent branching analysis histories by capturing independent analysis paths, their divergence points, and their ancestry relationships, ensuring that subsequent states and dependencies remain distinct across branches while preserving their common origin.

R4: Reusability of Analysis Steps. The provenance model shall represent a set of analysis steps as parameterizable pipelines and capture their structure (including operations, their ordering, dependencies, and configurations) to enable their identification, referencing for reuse and adaptation to different contexts.

R5: Support for Comparison of Analysis Histories. The provenance model shall represent analysis histories in a structured and transparent manner, including abstraction levels that group or categorize detailed analysis steps. This representation shall enable the systematic identification of similarities and differences in data selection, operations, and parameter configurations between multiple analysis histories, supporting meta-analysis through structured, side-by-side examination across histories that are not directly identical.

R6: Support for Comparison of Analysis States. The provenance model shall represent analysis states and defined abstractions of these states to enable the comparison of similarities and differences in their associated data. These abstractions shall capture relevant aspects or summaries (e.g., aggregated metrics, distributions, or feature-based representations) to support comparison when direct comparison is impractical due to complexity or dimensionality.

R7: Data Evolution Transparency. The provenance model shall represent the transformation lifecycle of data objects, such that each state transition can be examined in terms of the applied operations and the resulting changes in the data (e.g., event logs, models, KPIs).

As detailed in Table 1, each provenance purpose is supported by a specific subset of our requirements: foundational goals such as recall and presentation

Table 1. Correlation between provenance purposes and model requirements.

Purpose	Requirements
Recall	Traceability, Data Evolution Transparency
Presentation	Traceability, Comparison of Analysis States, Data Evolution Transparency
Meta-analysis	Traceability, Reusability of Analysis Steps, Comparison of Analysis Histories, Comparison of Analysis States
Replication	Reproducibility, Branching Capability, Reusability of Analysis Steps
Action Recovery	Traceability, Branching Capability
Collaborative Communication	Traceability, Comparison of Analysis Histories

rely primarily on traceability and data evolution transparency, whereas operational functions like action recovery and collaborative communication additionally depend on branching and history comparison. More complex objectives such as meta-analysis and replication require reproducibility, reusability, and the comparison of states or histories.

To perform an analytical comparison of the provenance models, each requirement is operationalized through a set of acceptance criteria defined as verifiable conditions. For example, one criterion for *Traceability of Analysis Steps* requires that, *given an analysis step, the provenance model must capture and link the exact operations performed and their precise parameter configurations to the corresponding step.* The complete set of 20 acceptance criteria for all remaining requirements is provided in [1]. These criteria are used to verify whether existing approaches meet the established requirements.

3 Literature Review and Comparative Analysis

To systematically identify and analyze relevant state-of-the-art approaches, our literature review was guided by the established methodology proposed by Kitchenham [5]. It was driven by the research question: to what extent do existing provenance models satisfy the requirements for exploratory process mining analysis? Using Scopus with the keywords "Provenance Model" and "Analytic Provenance", the search yielded 57 occurrences; after a screening phase and a manual snowballing search [12], we obtained a final corpus of 48 papers. To ensure reproducibility and transparency, the detailed source selection, exclusion criteria, and snowballing procedure are provided alongside a BPMN model of the entire review procedure in [1].

Following the initial retrieval, a rigorous screening phase was conducted to isolate the studies most pertinent to our research objectives. We applied strict exclusion criteria to (i) discard papers that focused exclusively on domain-specific applications of provenance, (ii) surveys or semantic integration studies lacking concrete provenance models, and (iii) any articles where the full text was not available. After applying these filters, the remaining 32 papers advanced to the detailed data extraction stage. To ensure the comprehensiveness of our review

Table 2. Partial comparison of approaches based on provenance requirements (●= fully supported; ◐= partially supported; ○= not supported)

Requirement	**Our model**	[13]	[9]	[2]	[7,6]	[4]	[11]
Traceability of A. Steps	●	●	●	●	●	●	●
Reproducibility	●	◐	◐	●	◐	●	○
Branching Capability	●	●	◐	◐	●	◐	●
Reusability of A. Steps	●	◐	●	◐	◐	●	○
Support for Comparison of A. Histories	●	●	●	●	◐	○	○
Support for Comparison of A. States	●	●	◐	◐	◐	○	●
Data Evolution Transparency	●	◐	◐	◐	◐	◐	●

and to capture foundational or highly relevant works potentially missed by the primary query, we complemented our methodology with a manual snowballing search [12]. This iterative process of examining references and citations identified an additional 16 papers, which were re-evaluated using the same screening and verification procedures to ensure full compliance with our requirements before being synthesized into the final analytical corpus.

With the final corpus of 48 papers established, we systematically assessed the state-of-the-art against our derived requirements. Because two of these papers were direct extensions of other works within the corpus, our evaluation ultimately focused on 46 distinct models. Each proposed approach was evaluated and classified for every requirement as either "Fully Supported", "Partially Supported" or "Not Supported". Specifically, if a model satisfies all acceptance criteria of a requirement, it is considered "Fully Supported". If it meets at least one acceptance criterion but fails others (e.g., when checking the traceability of analysis steps, the model tracks the sequence but lacks agent tracking or distinct parameter configurations), it is classified as "Partially Supported". If it meets none of the acceptance criteria, it is considered "Not Supported". To mitigate subjectivity, all classifications were independently reviewed by a second author, and disagreements were resolved through discussion. Detailed justifications and the underlying evidence for these classifications are provided in [1]. To visually synthesize these findings, Table 2 presents a comparison of our proposed model alongside the six approaches that most comprehensively fulfill the requirements, ordered by their overall degree of fulfillment. This table utilizes Harvey Balls to represent the classification levels (a filled circle indicates full support, a half-filled circle indicates partial support, and an empty circle signifies no support). The extended table detailing the evaluation of all 46 approaches is provided in [1].

The review reveals that Traceability of Analysis Steps is the most consistently supported feature (20 of 46 fully, 26 partially), whereas the model's headline capabilities remain weakly supported across three fronts. First, comparison is the most critical gap: comparing analysis states is supported only by 16 of 46 models (only 7 full), as models represent states without the schemas or structural constructs needed to compare high-dimensional data, and comparison of full histories is likewise largely unsupported (18 none, 7 full), since models lack the structural abstractions to represent multiple paths for systematic, side-by-side

comparison. Second, reusability is a major gap (21 none, 8 full): many models function strictly as logging tools for past executions, with no mechanisms to represent uninstantiated, parameterizable pipelines that analysts could extract and reuse. Third, comprehensive execution tracking is poorly supported (only 7 of 46 fully support reproducibility and a mere 3 provide full data evolution transparency) as models log operations and parameters but lack mechanisms to capture the complete runtime environment for reliable replay or to record the incremental, step-by-step changes between data states. Branching is only partially addressed, typically through annotations to track divergent decisions rather than an inherent ability to manage independent branches. Overall, because existing models act primarily as passive logging mechanisms, they misalign with the iterative, non-linear nature of exploratory process mining, leaving analysts unable to branch alternative hypotheses, compare divergent states, or handle the high-dimensional data that characterizes process mining analyses. To address these gaps, the next section presents our conceptual model, purposefully designed to satisfy our derived requirements and enable complex exploratory workflows.

4 A Conceptual Model to Enhance Process Mining Analysis with Provenance

This section presents the conceptual model for capturing analytic provenance in exploratory process mining. Sect. 4.1 derives the model's components from the requirements, Sect. 4.2 defines the core analytical entities, and Sect. 4.3 presents the formalized model with its structural mechanics and relationships.

4.1 Methodology

We employed a grammar-based derivation to transition from textual requirements to a conceptual model. Prior research indicates that applying such grammatical analysis to requirements engineering artifacts can significantly improve the completeness and accuracy of derived conceptual models [3]. Accordingly, nouns and noun phrases identified candidate entities and attributes, while verbs and relational expressions informed interactions and dependencies; this initial derivation was then refined by domain-specific modeling decisions.

This enabled traceable extraction of domain objects; for example, *R3* nouns ("analysis histories", "branches", "states"), informed entities like *AnalysisHistory*, *AnalysisBranch*, *AnalysisState* and *ArtifactState*, while verbs and relational phrasing yielded the model's structural relationships. This parsing ensured the extraction of the model's core objects (defined in Sect. 4.3). Furthermore, we iteratively refined this derivation process to achieve a comprehensive model.

4.2 Core Entities

The core of our conceptual provenance model is anchored by the *AnalysisHistory*, which encapsulates the entire lifecycle of an analytical process. To address *R3*,

this history is structurally decomposed into independent *AnalysisBranch* paths. The progression within each branch is recorded as a sequence of immutable *AnalysisState* instances, which are bridged by active *AnalysisStep* executions. Each step represents the concrete application of a specific *Operation* (e.g., filters, attribute derivations), which is further classified by an *OperationType* and a broader *StepCategory* (e.g., log enrichment, process discovery). These operations are performed by an *Agent* within a precisely captured *RuntimeEnvironment*. To satisfy *R1* and *R2*, every step documents the exact configuration applied, linking concrete *ParameterValue* instances to their overarching *Parameter* definitions.

At the granular data level, an *AnalysisState* represents a holistic snapshot of the overall analysis at a specific point in time. It acts as a container that can encompass one or more *ArtifactState* instances, which are themselves immutable snapshots of specific artifacts (e.g., event logs, process models) utilized or generated during the analysis. To support *R7*, the model employs a *Delta* construct, designed as an abstract class to accommodate various subclasses (e.g., *DataFrameDelta*). These subclasses explicitly capture the specific, incremental structural modifications and transformations occurring between states by linking a root *ArtifactState* to an updated *ArtifactState*. By establishing this direct link, the model facilitates the generation of artifact-specific deltas, allowing for precise tracking of changes tailored to the type and structure of each artifact. Additionally, the diagram visually abstracts *Delta* as a single conceptual entity, omitting its underlying artifact-specific subclasses (e.g., *DataFrameDelta*). Direct comparison of raw analysis states is frequently impractical due to their size, complexity, and heterogeneity. To address this challenge and support *R5* and *R6*, the model introduces *StateAbstraction*, which provides comparable summaries or aggregated representations (e.g., trace variant distributions, performance metrics, or feature-based representations) of distinct states and overarching histories. Finally, to fulfill *R4*, sets of operations can be combined into modular *PipelineFragment* entities. Rather than treating an entire reusable workflow as a single, rigid sequence, utilizing fragments allows analysts to capture and reuse interchangeable sets of independent analysis steps, which can even be drawn from different branches. Furthermore, fragments extracted from several distinct analysis histories can be flexibly aggregated to compose a single pipeline. Within a pipeline of reusable steps, the specific operations executed are managed by explicit *Version* entities, ensuring that the lineage, structural integrity, and adaptability of reusable components are maintained across diverse execution contexts.

4.3 Conceptual Model

The formalized conceptual model is depicted as a UML class diagram in Fig. 1. This model is designed to capture the dynamic nature of exploratory analysis, enforcing an immutable and tree-based execution flow. We focus on a tree structure because it accurately reflects the chronological and step-by-step branching history of the analysis, providing a natural representation of how an analyst investigates alternative hypotheses over time without overwriting previous steps. In essence, an *AnalysisHistory* acts as the root container, aggregating one or more

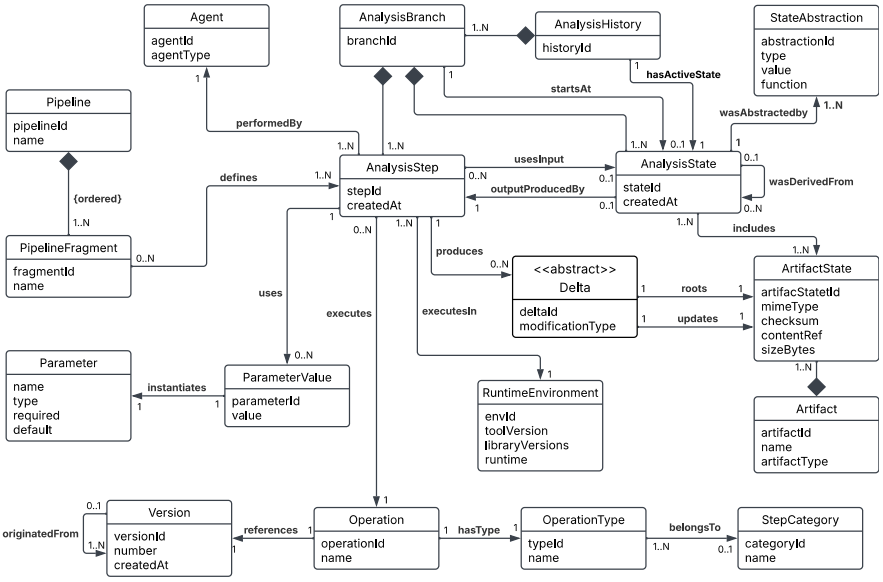


Fig. 1. Provenance Conceptual Model for Exploratory Process Mining

AnalysisBranch entities. This composition allows the model to capture parallel analysis paths without overwriting previous work (*R3*), with each branch explicitly tracking its structural origin via a *startsAt* relationship to its originating analysis state. Within a given branch, the analytical progression is modeled as a composition of static *AnalysisState* nodes and active *AnalysisStep* transitions.

The model differentiates between input and output (made possible by explicit relationships between *AnalysisStep* and *AnalysisState*) to maintain a traceable exploratory process mining workflow (*R1*). An *AnalysisStep* consumes exactly one source *AnalysisState* and produces exactly one target *AnalysisState*. To document the evolution of the data (*R7*), the step produces instances of *Delta* entities. To capture the exact transformation, this abstract entity is extended into artifact-specific subclasses (e.g., *DataFrameDelta*). Each *AnalysisState* encapsulates data through *ArtifactState* objects, acting as snapshots of overarching conceptual *Artifact* entities and utilizing specific metadata (*checksum*, *sizeBytes*, *contentRef*, and *mimeType*) to enable precise tracking and retrieval of data. Additionally, each state defines *StateAbstraction* entities, which use designated *functions* to facilitate comparison (*R5* and *R6*).

To support reproducibility (*R2*), every *AnalysisStep* is linked to the *Agent* that performed it and the *RuntimeEnvironment* (capturing specific *toolVersion* and *libraryVersions*). The underlying functional logic is encapsulated by an *Operation*, which is structurally classified by an *OperationType* (e.g., attribute derivation) and a broader *StepCategory* (e.g., enrichment). Operations are tightly coupled with their configurations: an *Operation* aggregates an ordered set of *Parameter* definitions (specifying data *type*, *required* flags, and *default* values),

which are concretely instantiated by the precise *ParameterValue* entities logged during a specific *AnalysisStep*. For readability, the specific parameters used as inputs for an *AnalysisStep*, such as an *ArtifactState*, are not shown.

Reusability (*R4*) is structurally enforced through a composite workflow design. Rather than representing workflows as rigid monoliths, a *Pipeline* orderedly aggregates *PipelineFragment* entities. These fragments define specific sets of *AnalysisStep*s that function as reusable templates within a *Pipeline*. While *AnalysisStep*s are typically associated with *AnalysisState*s through input and output relationships, these associations are optional within fragments. This allows independent steps from different branches to be grouped without enforcing a rigid internal sequence. It is only within the overarching *Pipeline* that these fragments are strictly ordered. This design allows granular analytical sub-routines to be independently reused and adapted across different pipelines. Governance over both individual operations and pipelines is maintained by referencing *Version* entities, which capture creation timestamps and hierarchical lineage (*originatedFrom*).

5 Conclusion and Future Work

The proposed conceptual model provides a new understanding of analytic provenance by formalizing the highly iterative and non-linear realities of exploratory process mining. Designed to fulfill all derived requirements, it addresses gaps identified in the literature regarding the comparison of analysis histories and states, the modular reuse of analysis steps, and comprehensive execution tracking. While our approach shares similarities with models such as Flowris [9], FORSETI [11], Chimera [4], and Urbanite [6], and directly adapts mechanisms from related branching research [13], it extends that branching mechanism with dedicated structural constructs for the modular reuse of pipeline fragments and the systematic comparison of analysis histories and states (capabilities absent from these prior models). Future interactive tools built on this model will elevate provenance from a static log into an active structural enabler.

Despite these advancements, several limitations guide our future research. First, we will implement this conceptual foundation into a tool and carry out a thorough validation, including a qualitative evaluation with process mining experts to assess the usefulness and completeness of the captured provenance, together with comparative studies against existing tools. We will further extend it to process-specific operations such as discovery and alignment, and to Object-Centric Event Logs (OCEL). Second, we will enhance the model's practical utility by integrating human-centric annotations so analysts can record alternative hypotheses, and by embedding the model within an assisted approach that lowers its adoption complexity through explainability and reuse-based guidance. Additionally, we will develop tree pruning algorithms, which are necessary to simplify complex exploratory paths and communicate findings clearly to stakeholders. By satisfying requirements grounded in fundamental analytic provenance purposes, this work provides the necessary structural foundation to support transparency and rigor in complex exploratory workflows.

Acknowledgments

The work of M. Resinas is part of the project PID2024-156482NB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU.

References

1. Bernal-Caunedo, Á., Franceschetti, M., Völzer, H., Resinas, M., Weber, B.: A conceptual model to enhance process mining analysis with provenance (2026), https://doi.org/10.5281/zenodo.21373047
2. Chapman, A.P., Jagadish, H.V., Ramanan, P., et al.: Efficient provenance storage. In: Proc. of the 2008 ACM SIGMOD Int. Conf. on Management of data. pp. 993–1006 (2008)
3. Dalpiaz, F., Gieske, P., Sturm, A., .: On deriving conceptual models from user requirements: An empirical study. Inf. Softw. Technol. **131**, 106484 (2021)
4. Foster, I., Vockler, J., Wilde, M., Zhao, Y.: Chimera: A virtual data system for representing, querying, and automating data derivation. In: Proc. 14th Int. Conf. on Scientific and Statistical Database Management. pp. 37–46. IEEE (2002)
5. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE-2007-01, Keele University and University of Durham (2007)
6. Moreira, G., Ferreira, L., Veiga, C., Hosseini, M., Miranda, F.: Urbanite: A Dataflow-Based Framework for Human-AI Interactive Alignment in Urban Visual Analytics. IEEE Trans. Vis. Comput. Graph. **32**(1), 1065–1075 (2026)
7. Moreira, G., Hosseini, M., Veiga, C., Alexandre, L., Colaninno, N., de Oliveira, D., Ferreira, N., Lage, M., Miranda, F.: Curio: A Dataflow-Based Framework for Collaborative Urban Visual Analytics. IEEE Trans. Vis. Comput. Graph. **31**(1), 1224–1234 (2025)
8. Ragan, E.D., Endert, A., Sanyal, J., Chen, J.: Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. IEEE Trans. Vis. Comput. Graph. **22**(1), 31–40 (2015)
9. Sun, J., Wang, J., Chen, Y., Qin, X.: Flowris: Managing Data Analysis Workflows for Conversational Agent. In: Lecture Notes in Computer Science. vol. 13946 LNCS, pp. 724–728 (2023)
10. Van Der Aalst, W.: Process mining: Overview and opportunities. ACM Trans. Manag. Inf. Syst. **3**(2), 1–17 (2012)
11. Wang, B., et al.: Forseti: A visual analysis environment enabling provenance awareness for the accountability of e-autopsy reports. Visual Informatics **6**(3), 69–80 (2022)
12. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proc. of the 18th Int. Conf. on Evaluation and Assessment in Software Engineering (2014)
13. Zerbato, F., Burattin, A., Völzer, H., Becker, P.N., Boscaini, E., Weber, B.: Supporting provenance and data awareness in exploratory process mining. In: Int. Conf. on Advanced Information Systems Engineering. pp. 454–470. Springer (2023)
14. Zerbato, F., et al.: Initial insights into exploratory process mining practices. In: Int. Conf. on business process management. pp. 145–161. Springer (2021)