https://doi.org/10.31449/inf.v50i14.14899 Informatica **50** (2026) 343–356 **343**

Configurable Process Mining for Context-Aware Customer Journey Analysis: A Simulation-Based Comparative Study

Imane El Alama, Hanae Sbai
Mathematics, Computer Science and Application Laboratory, Faculty of Sciences and Techniques of Mohammedia, Hassan II University of Casablanca, Mohammedia 28806, Morocco
imane.elalama-etu@etu.univh2c.ma
hanae.sbai@fstm.ac.ma

Keywords: Customer journey maps, process mining, configurable process mining, variability modeling, context-aware analytics

Received: April 23, 2026

Customer journeys frequently differ depending on the context. Customer experiences with a digital service can be influenced by factors such as location, delivery limitations, or payment methods. Process mining can reconstruct customer journeys from event logs. However, classical process discovery usually integrates all observed behavior into a single model, which may obscure context-specific variations. This study examines whether such variations can be represented more explicitly within a shared process structure using configurable process mining. A synthetic e-commerce event log was generated for two regional contexts with distinct shipping and payment policies, comprising 1,600 customer sessions, approximately 7,500 events, and 13–14 activity types. Using ProM, the Inductive Miner algorithm was applied to discover a global process model and region-specific process trees, which were then merged and configured into regional variants. Fitness and precision were used as the main evaluation metrics. In this controlled setting, the classical model achieved higher fitness (0.9438) but lower precision (0.7129), while the configured regional models achieved higher precision (0.996) with lower fitness (0.902). These findings suggest that configurable process mining may be more informative when the objective is to analyze contextual differences, whereas classical discovery remains useful for a broad understanding of the process. Real-world validation remains necessary to assess the robustness of the approach on operational customer journey logs.

Povzetek: Študija kaže, da konfigurabilno procesno rudarjenje bolje prikaže kontekstualne razlike v uporabniških poteh, medtem ko klasični pristop ponuja širši pregled procesa.

1 Introduction

Customer Journey Maps (CJMs) are widely used to analyze how customers interact with digital services across multiple touchpoints [1]. However, many CJMs are still built from workshops, expert assumptions, or aggregated views of customer behavior, which may obscure the diversity of real customer journeys [2]. Process mining addresses this limitation by reconstructing process models directly from event logs, allowing observed customer behavior to be analyzed from data rather than assumptions [3].

Classical process discovery is useful for obtaining a global view of customer behavior, but it usually produces a single aggregated model when logs from multiple contexts are combined. In customer journey analysis, this can hide important context-specific differences, such as regional delivery constraints, payment preferences, or service configurations. As a result, behaviors that are valid only in one context may appear together in the same model, reducing interpretability for context-specific analysis.

Configurable process mining (CPM) offers a way to represent shared and variable behavior within a common process structure [4]. Instead of maintaining separate models for each context or merging all behavior into a single unrestricted model, CPM allows context-specific alternatives to be enabled, hidden, or disabled. Although CPM has been studied in general process variability settings, its operationalization for customer journey analysis remains limited.

This study examines whether configurable process mining can improve the interpretation of customer journey models when contextual variability is present. A controlled simulation-based experiment is conducted using a synthetic e-commerce event log with two regional contexts and distinct delivery and payment rules. The objective is to compare classical and configurable process discovery in terms of fitness, precision, and interpretability.

This study does not introduce a new configurable process mining algorithm. Its contribution is methodological and application-oriented. More specifically, it examines how existing CPM techniques can be operationalized for customer journey analysis when contextual variability is present. The paper should therefore be understood as a proof-of-concept study focused on the problem–solution fit between configurable process mining and customer journey variability.

The study is guided by the following research questions:

RQ1. Does configurable process mining improve precision compared with classical process discovery in context-specific customer journey analysis?

RQ2. What fitness trade-off is introduced when context-specific configured variants are derived from a shared configurable process model?

RQ3. Does configurable process mining improve the interpretability of customer journey models by explicitly separating shared behavior from context-specific behavior?

The contributions of this study are fourfold. First, it frames customer journey variability as an explicit configuration problem. Second, it operationalizes existing CPM techniques through a structured workflow including synthetic event log generation, classical/configurable discovery, and comparative evaluation. Third, it provides a theoretical rationale for the expected precision improvement of configured variants under behavioral restriction. Fourth, it provides a proof-of-concept comparison with classical process discovery in a controlled e-commerce setting.

The remainder of the paper is organized as follows. Section 2 discusses related work, Section 3 introduces the necessary background, Section 4 presents the methodological setup, Section 5 reports the results, and Sections 6 and 7 discuss the implications and conclude the study.

2 Related work

2.1 Comparative analysis summary

The following table presents a comparative analysis of 11 representative papers, highlighting how each addresses process mining, variability dimensions, and treatment approaches. This analysis systematically demonstrates the research gap that our work addresses.

Table 1: Comparative synthesis of related work on process mining, customer journey mapping, and variability handling

Work	PM Method	Variability Dimension	Treatment Approach	Dataset / Domain	Key Results
[5]	Literature review	Customer journey variability	Review of PM applications for CJM	Scopus literature review	Highlights the potential of PM for CJM and the need for integrated approaches
[6]	Conceptual / analytical study	Touchpoints, service contexts, and end-to-end journey complexity	Analysis of PM–CJM challenges and opportunities	Customer journey / service systems	Identifies challenges such as missing touchpoints, cross-organizational journeys, and subjective experience capture
[7]	Process discovery	E-commerce visitor journey variability and cart-state transitions	Cart-state-aware process discovery	E-commerce visitor journeys	Proposes a structured methodology to extract and understand e-commerce visitor journeys using PM
[8]	Process discovery + recommendation	Web customer journey variability and user activity sequences	Discovery-to-recommendation pipeline	Web / clickstream customer journeys	Uses PM to analyze customer journeys and generate recommendations
[9]	Process mining + sequence-aware recommendation	Web journey behavior and sequence variability	Process discovery, clustering, and recommendation	Web log customer journey data	Combines PM and recommender systems for customer journey optimization

[10]	Discovery + conformance checking	Rehabilitation variability by clinical and demographic factors	Stratified journey comparison	Healthcare stroke rehabilitation	Demonstrates the value of PM for patient-centered journey analysis and subgroup comparison
[11]	Process discovery	Touchpoint, channel, stage, and experience variability	PM-based CJM construction	Customer journey mapping	Bridges process mining and CJM through event-log-based journey modeling
[12]	Process model merging	Variability between related process models	Model consolidation / merging	Business process models	Provides a method for merging related process models
[13]	Configurable workflow modeling	Workflow variant variability	Configurable model design	Workflow / process families	Represents multiple workflow variants in one configurable model
[14]	Survey	Business process variability	Review of variability modeling approaches	BPM literature	Synthesizes business process variability modeling methods
[15]	Configurable process fragment mining	Process fragment variability	Mining configurable fragments	Business process design	Supports configurable process design from reusable fragments

Table 1 summarizes the main methodological orientations of prior work and serves as the basis for the analytical positioning developed in Section 2.2.

2.2 Research gap & positioning

The studies reviewed in Table 1 can be grouped into three research streams.

The first stream applies process mining to customer journey or service journey analysis. This includes works such as [5], [6], [7], [8], [10], and [11]. These studies show that process mining can support data-driven journey reconstruction, customer journey mapping, healthcare journey analysis, and model-based interpretation of customer behavior. Recent works, particularly [5], [6], [7] and [10], confirm the growing interest in connecting process mining with journey analysis. However, these studies mainly focus on discovery, conformance analysis, visualization, recommendation, or journey interpretation. Contextual variability is generally identified, compared, or used analytically, rather than explicitly encoded within a unified configurable process structure.

The second stream addresses behavioral heterogeneity through segmentation, clustering, recommendation, or variant-oriented analysis. Journey-oriented studies such as [8], [9], and [10] consider differences between user sequences, customer paths, or behavioral groups. These approaches are useful for identifying distinct journey variants, predicting next touchpoints, or supporting recommendations. However, variability is usually treated as an analytical output. Trace clustering separates traces into groups, while recommendation-oriented approaches use observed sequences to guide future actions. In both cases, the relationship between shared behavior and context-specific alternatives is not represented as part of a common configurable model.

The third-stream concerns variability-aware process mining, process model merging, configurable workflow models, and configurable process fragments. This includes works such as [12], [13], [14], and [15]. These contributions provide the conceptual and technical foundations for representing process variability, consolidating process models, and designing configurable process structures. However, their primary focus is general business process variability, workflow variants, or process family management. They are not specifically oriented toward customer journey mapping as an analytical artifact, nor toward evaluating geographic or context-specific customer journey variability against classical process discovery.

The research gap should therefore be understood in a narrow sense. The issue is not the absence of variability-aware process mining, clustering, recommendation, or

variant analysis. Rather, the gap concerns the limited operationalization of configurable process mining for customer journey analysis when contextual variability must be represented within a common reference model.
This study addresses this gap by applying existing CPM techniques to customer journey mapping and evaluating whether configured variants can make region-specific behavior explicit while preserving comparability across contexts. This differentiates the study from trace clustering, which separates traces into groups; from recommendation-oriented approaches, which exploit observed sequences for future action guidance; and from general variability-aware process mining, which is not specifically oriented toward customer journey interpretation.

3 Background

3.1 Customer journey maps: formal foundations and process-oriented perspective

Customer Journey Maps (CJMs) constitute visual and analytical representation of customer interactions with a service system across multiple touchpoints and temporal stages. From a process perspective, a CJM can be formalized as a sequence of activities (touchpoints) ordered by time and context, where each activity represents a moment of customer-brand interaction. This process-oriented view aligns naturally with event log representation in process mining: each customer journey becomes a trace, and collections of journeys form an event log recording the observed behavior of the service system [16].

Key Insight: CJMs exhibit substantial variability across customer segments, geographic contexts, or service conditions. This variability representing different journey paths followed by different customer groups parallels the concept of process variants in business process management. Traditional approaches either merge all variants into a single aggregated model or maintain separate models for each variant, both approaches having significant drawbacks in terms of model comprehension and analytical utility [1], [17].

3.2 Process mining foundations: scalability and role for macro-level analysis

Process mining comprises three interrelated tasks: process discovery (automatically extracting process models from event logs), conformance checking (assessing model quality against observed behavior), and process enhancement (refining models based on performance insights) [18]. In this work, we focus primarily on discovery and conformance checking.

MacroLevel Effectiveness: Process mining has demonstrated strong effectiveness for analyzing processes at the macro (global or organizational) level. Classical process discovery techniques, when applied to aggregated event logs encompassing all behavioral variants, provide valuable overviews of overall system behavior and are particularly effective for identifying global workflow patterns, system-wide bottlenecks, and enterprise level process optimization opportunities [2]. Process mining remains a solid, validated methodology for understanding aggregate business operations and has been successfully applied across diverse domains including healthcare, education, finance, and e-commerce [5], [19], [20].

Complementary Role of Configurable Process Mining: The role of configurable process mining (CPM), as presented in this work, is not to replace classical process mining but to complement it by addressing a specific analytical need: understanding process behavior at the context-specific level. While process mining excels at revealing global patterns, it can mask important context-dependent variations that are critical for customer experience optimization, service personalization, and context-aware decision-making. CPM provides granular, context-specific models while maintaining the advantages of classical process mining for macro-level analysis.

3.2.1 Event logs and trace representation

An event log is a collection of execution traces recording the observed behavior of a process. Formally, let Σ be a finite set of activities. An event log is denoted $L \in \mathbb{B}(\Sigma^*)$, where each trace $\sigma \in \Sigma^*$ is a finite sequence of activities. In the context of CJMs, each trace represents a single customer journey, and the event log aggregates all observed journeys within a given time period [18].

3.2.2 Conformance checking: fitness, precision, and model quality metrics

Process discovery algorithms automatically construct process models from event logs, aiming to extract the underlying behavioral patterns [18]. Given an event log L, a discovery function δ maps the log to a process model M:
$\delta: L \rightarrow M$
The quality of discovered models is assessed using conformance checking, which compares expected behavior described by a reference model to actual recorded behavior in an event log [21]. Four principal dimensions assess model quality [21]:

Fitness (or Recall): Measures the degree to which the model can replay the traces in the event log. High fitness indicates that the model recognizes and accommodates the ob-served behavior. Formally, fitness quantifies what proportion of the observed traces are accepted by the model without deviations [21]. A fitness value of 1.0 indicates perfect alignment between model and log, while lower values indicate missing or incorrectly modeled activities [22].

Precision: Measures the extent to which behavior allowed by the model corresponds to behavior actually observed in the event log. Low precision indicates that the model permits "too much" behavior beyond what was recorded, potentially making it an overgeneralized representation unsuitable for understanding actual customer patterns [21]. This metric is particularly important for context-specific analysis: a precise model for a specific customer segment accurately captures only the behaviors relevant to that segment, excluding irrelevant paths from other segments that a global model might permit [18].

Generalization: Measures the model's ability to represent correct but unseen behavior of the underlying real process. High generalization indicates that the model is robust and not overfitted to the training log [21].

Simplicity: Measures the structural complexity of the process model (number of nodes, edges, branches). More simple models are preferred as they are easier to understand and maintain [23].

These metrics are central to model quality assessment and directly motivate our approach to context-specific configurability, as they enable quantitative comparison between classical (global) and configurable (context-specific) models.

3.3 Configurable process mining: mathematical framework and precision improvement

Configurable process models represent a family of related process variants within a single coherent structure. Rather than creating separate models for each customer segment or context, or merging all variants into one large model, configurable process mining dis-covers structural patterns common to all variants while explicitly capturing con-text-specific deviations [15].

3.3.1 Process tree as formal representation

A process tree is a hierarchical, block-structured formal representation of process control-flow behavior. It provides a comprehensive alternative to flat, graphical represen-tations by organizing process logic into a tree structure where each node represents either a control-flow operator or an activity.

Formal Definition: A process tree PT is defined recursively as follows: (1) An activity node $a \in \Sigma$ is a process tree; (2) If $PT_1, PT_2, ..., PT_n$ are process trees and $op \in \{\rightarrow, \times, \wedge, \vee\}$ is a control-flow operator, then $op(PT_1, PT_2, ..., PT_n)$ is a process tree.

Control-Flow Operators:

- Sequence Operator (→): Activities are executed in a fixed, deterministic order from left to right.
- Exclusive Choice (XOR or ×): Exactly one of the child branches is selected at runtime.
- Parallel (AND or ∧): All child branches execute concurrently without a fixed order.
- Loop (∨): Allows a branch to repeat multiple times before moving to the next activity.

Process trees are suitable for this study because they provide a sound, block-structured representation of process behavior. Their hierarchical structure makes it possible to identify common fragments and context-specific branches, which is necessary for merging regional models and deriving configured variants.

3.3.2 Theoretical rationale for precision improvement

Configured process models are expected to improve precision because they restrict the behavior allowed by the global model to the behavior relevant to a given context. In a global model discovered from an aggregated event log, alternatives from different contexts may be combined. As a result, the model may allow paths that are valid globally but not valid for a specific region or customer segment.

In contrast, a configured model removes or hides context-irrelevant branches. If this restriction does not remove behavior that is actually observed in the corresponding context-specific log, the model preserves coverage while reducing the space of allowed behavior. Under this condition, precision is expected to be non-decreasing.

This principle can be summarized as follows:

Proposition 1. Non-decreasing precision under behavioral restriction.

For a given context-specific event log, if the configured model restricts the behavior allowed by the global model while still covering the observed traces of that context, then the precision of the configured model is greater than or equal to the precision of the global model for that context.

This proposition provides the theoretical rationale for the precision improvement observed in the experiments. It is not specific to the synthetic dataset and applies to synthetic or real event logs, provided that the stated assumptions are satisfied. The detailed formalization and proof are provided in Appendix A.

4 Materials and methods

This section presents the data, tools, and methodological choices adopted in this study. The research follows a simulation-based experimental design, which enables controlled comparison between classical and configurable process mining under known business rules. The design is aligned with the research questions stated in Section 1: RQ1 is evaluated through precision, RQ2 through fitness, and RQ3 through qualitative analysis of model structure, context discrimination, and interpretability.

4.1 Research design

The study adopts a controlled simulation-based methodology focused on one contextual dimension: geographic region. This choice allows the underlying process, regional rules, and expected variants to be known in advance. It also ensures that both classical and configurable process mining are evaluated on the same event log under identical conditions.

The methodology consists of four steps. First, a customer journey process is defined with common activities and region-specific business rules. Second, a synthetic event log is generated through stochastic simulation, including temporal dynamics, regional variability, and controlled noise. Third, classical process discovery and configurable process mining are applied to the same log using the Inductive Miner algorithm. Fourth, the resulting models are compared in terms of fitness, precision, interpretability, context discrimination, and structural simplicity.

4.2 Synthetic event log generation

A synthetic e-commerce clickstream event log was generated to support controlled feasibility evaluation. The dataset contains 1,600 customer sessions, with 800 sessions per region, collected over 31 days. It includes approximately 7,500 events and 13–14 activity types.

The event log was generated according to the simplified simulation model shown in Figure 1. The model captures the main stages of the customer journey, including arrival, exploration, cart and checkout, shipping, payment, coupon use, and final outcome. It also summarizes the main simulation parameters, including Poisson-based arrivals, bounded loops during exploration and coupon use, region-specific shipping and payment rules, confirmation probabilities, abandonment points, and controlled noise.

For readability, Figure 1 reports only the main parameters. The complete Python script, including all branching probabilities, temporal parameters, region-specific rules, and noise settings, is provided as supplementary material to support replication. The generated CSV event logs are also provided, and the process mining experiments can be reproduced in ProM by applying the Inductive Miner with the same settings and by following the classical and configurable discovery pipelines described in Section 4.3.

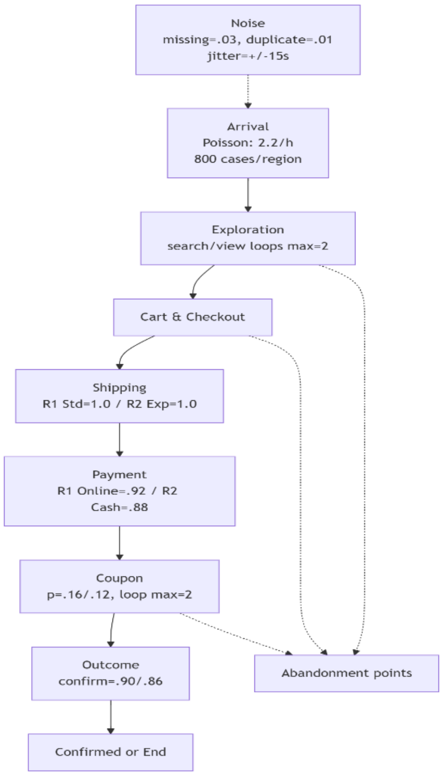

Figure 1: Simplified simulation model for synthetic event log generation.

The simulation parameters were selected to approximate common properties of digital interaction logs while preserving experimental control. Poisson-based arrivals were used because Poisson models are commonly applied to web browsing session arrivals [24]. Lognormal interevent times were adopted because human interaction delays are often skewed and heavy-tailed, and have been studied using lognormal assumptions [25]. The injected noise reflects well-known event-log quality issues in process mining, including missing events, duplicated events, and timestamp imperfections [26], [27]. These parameters are therefore intended to define a realistic controlled scenario rather than to reproduce a specific operational platform.

4.3 Experimental workflow and tool support

Figure 2 summarizes the workflow used in this study. The workflow starts from a synthetic event log and applies two analysis pipelines (classical and configurable). The results are compared at the end.

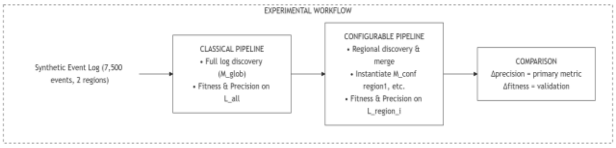

Figure 2: Workflow of classical and configurable process mining discovery.

Classical Pipeline. The Inductive Miner algorithm is applied to the complete event log without regional stratification, producing a single global process tree M_{glob}. Fitness and precision are then computed on the full log L_{all}, as defined in Section 3.2.2.

Configurable Pipeline. The event log is partitioned by region into $L_1$and L_2. Inductive Miner is applied separately to each regional log to obtain process trees $T_1$and T_2. These trees are merged in ProM into a configurable process tree M_{conf}, from which region-specific variants are instantiated. Fitness and precision are computed for each configured variant.

Configuration logic. The configuration procedure was applied as follows:

Step 1 — Merge regional process trees: The process trees discovered for Region_1 and Region_2 were merged into a single configurable process tree.

Step 2 — Preserve shared behavior: Activities and control-flow structures common to both regional trees were retained as non-configurable elements.

Step 3 — Identify configuration points: Structural divergences between the regional trees were marked as configuration points, especially when an activity or branch appeared in one region but was absent or invalid in the other.

Step 4 — Resolve regional alternatives: For each regional variant, valid branches were enabled, while invalid region-specific branches were hidden or disabled.

Step 5 — Instantiate configured variants: The final regional variants preserved the shared customer journey structure while restricting behavior not valid for the selected region.

The Inductive Miner was applied with a noise threshold of 20%. The same threshold was used for both the classical and configurable discovery pipelines to ensure a fair comparison between the two approaches. This threshold was selected to filter infrequent behavior and controlled noise while preserving the main customer journey structure and region-specific behavioral differences. This choice is consistent with process mining literature, where infrequent behavior filtering is used to reduce the influence of noise and improve the readability of discovered models [4], [28]. Region A, split into two parts The objective was not to maximize the inclusion of all rare traces, but to obtain sound and interpretable block-structured models suitable for comparing classical and configurable discovery.

4.4 Evaluation: fitness and precision focus

The two approaches were evaluated using quantitative and qualitative criteria. Fitness and precision were used as the main quantitative metrics. Fitness evaluates the extent to which the model can reproduce the traces observed in the event log, while precision evaluates whether the model avoids allowing behavior not observed in the log.

Interpretability, context discrimination, generalization, and simplicity were assessed qualitatively. These criteria were used to compare the ability of the classical and configurable approaches to represent shared behavior, context-specific alternatives, and region-dependent customer journey patterns.

5 Results

This section reports the results obtained from the application of classical and configurable process mining to the synthetic event log. The results follow the experimental workflow presented in Figure 2, starting with the global model discovered through classical process mining and continuing with the configurable models and their associated evaluation metrics.

5.1 Classical discovery: global model

Figure 3 shows the global process model obtained through classical process mining and visualized using an Inductive Visual Miner representation. This representation emphasizes customer journey states and transitions, providing a compact view of the overall control-flow structure.

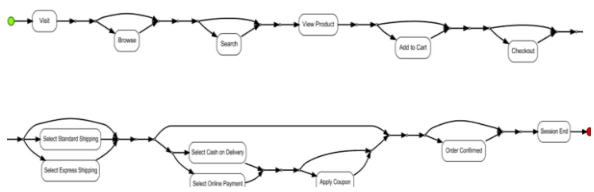

Figure 3. Global process model discovered from the combined event log using the Inductive Visual Miner. The model provides an aggregated view of customer behavior across both regions; context-specific variability is embedded in the same structure rather than shown separately.

In this visualization, the green circle represents the start of the process, while the red circle marks the end. The model shows a coherent sequence of customer interactions, from the initial access to the completion of an order. By aggregating all observed executions into a single process model, it captures the dominant behavioral structure derived from the event log. From an analytical perspective, the Inductive Visual Miner provides a global view of the discovered process. Behavioral variability is therefore embedded within a unified structure and is not explicitly distinguished according to context.

5.2 Process tree for each event log (Region A and Region B)

In this step, the global synthetic event log is split into event logs per region, and process discovery using the Inductive Miner is applied independently to each log to obtain one process tree per region. At this stage, no comparison between regions is made, as the objective is solely to generate intermediate process tree models. Figures 4 and 5 present the process trees obtained for each region, which will be used as inputs for the subsequent process tree merging step.

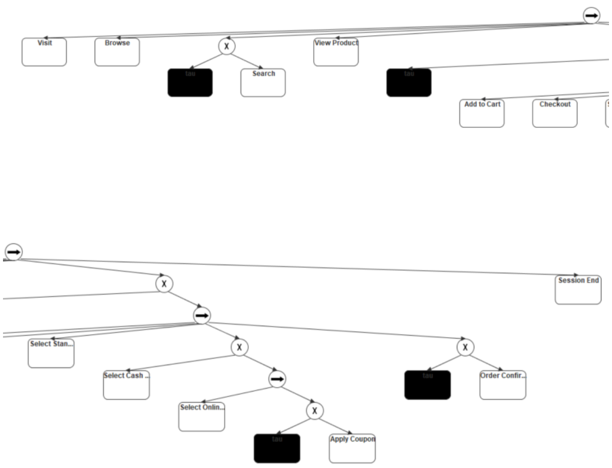

Figure 4. Process tree discovered for Region A. split into two parts for readability. The model represents the control-flow behavior observed in the Region A event log and is used as input for the subsequent merging step.

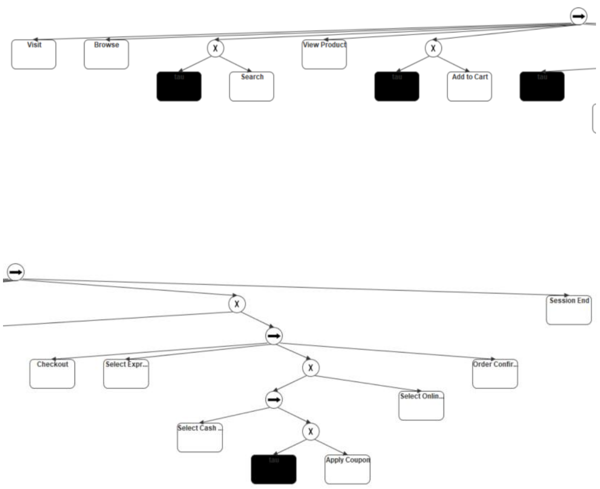

Figure 5. Process tree discovered for Region B. split into two parts for readability. The model represents the control-flow behavior observed in the Region B event log and is used as input for the subsequent merging step.

Figures 4 and 5 illustrate the process trees obtained for Region A and Region B. In these trees, leaf nodes correspond to activities, while internal nodes represent control-flow operators that structure the execution logic, such as sequence and branching. The black arrow denotes the sequence operator, indicating that the connected activities are executed in a fixed order from left to right. In addition, the X node denotes an exclusive choice (XOR): at runtime, exactly one of its children branches is selected, meaning that the alternative behaviors connected to this operator are mutually exclusive (only one path can occur per case). This hierarchical, block-structured representation ensures that each subtree forms a well-defined fragment of the process. At this stage, no comparison between regional variants is performed; the process trees are used solely as formal representations of region-specific control-flow behavior and as inputs for the subsequent process tree merging step.

5.3 Merge of process

In this step, the process trees obtained for each region in the previous section are merged using a process tree merging plugin in ProM. The purpose of this operation is to construct a single configurable process tree that integrates the regional models into a unified structure prior to configuration, as illustrated in Figure 6.

To improve readability, the merged tree should be interpreted by focusing on the configurable nodes. These nodes indicate the points where the regional variants differ, mainly in delivery and payment choices.

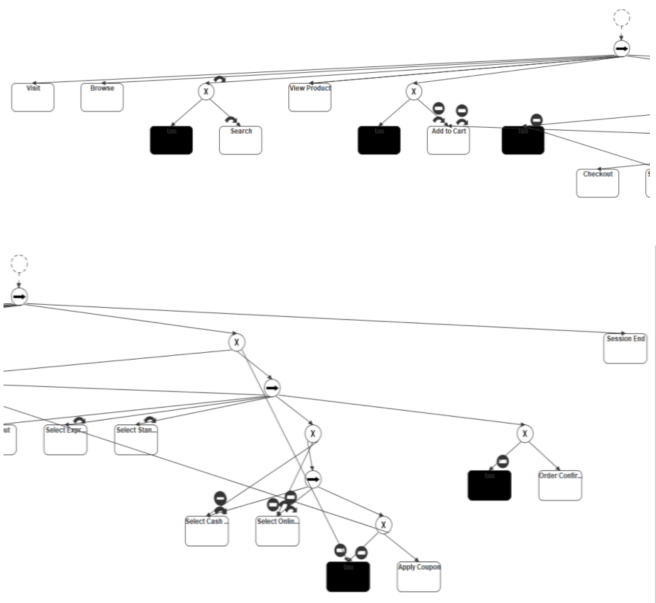

Figure 6. Merged configurable process tree obtained by integrating the regional process trees before configuration. Configurable nodes indicate variation points where region-specific behavior can be enabled, hidden, or blocked. These variation points mainly concern delivery and payment alternatives.

Figure 6 presents the merged configurable process tree obtained by integrating the regional process trees into a unified structure. Leaf nodes correspond to observable activities, while internal nodes represent control-flow operators such as sequence and XOR. The configurable nodes indicate the main variation points between regions, especially delivery and payment alternatives. At this stage, no regional configuration is applied; the merged tree defines the configuration space from which region-specific variants are later derived.

5.4 Configuration results

The configurable process model is instantiated for each region using the Exhaustive Configurable Process Tree Configuration plugin in ProM. In the remainder of the paper, the term "configured variant" refers to the context-specific process model derived from the configurable process tree for a given region. This plugin evaluates the configurable process tree against a context-specific event log and determines, for each configurable node, whether it is enabled (E) or hidden (H) in the corresponding configured variant. The configuration outputs are reported in Figures 7 and 8 for Region A and Region B, respectively**.**

In Figures 7 and 8, the most important information is the configuration state of each node. Enabled nodes remain visible in the configured variant, whereas hidden nodes are replaced by silent transitions.

Conformance		0,912	
quality metrics	fitness	0,902	90 %
	precision	0,996	10 %
	generalization	0,788	0 %
	simplicity	0,935	0 %
Number of configurations		15	
Configured nodes	Select Online Payment	E	
	Select Express Shipping	H	
	Add to Cart	E	
	Select Cash on Delivery	H	
	Search	E	
	tau	B	
	Select Standard Shipping	E	
	Xor(tau, Search)	E	

Figure 7: Configuration output for Region A. The table reports quality metrics and the configuration state of each configurable node, where E indicates an enabled activity and H indicates a hidden activity.

Conformance		0,912	
quality metrics	fitness	0,902	90 %
	precision	0,998	10 %
	generalization	0,816	0 %
	simplicity	0,967	0 %
Number of configurations		15	
Configured nodes	Select Online Payment	H	
	Select Express Shipping	E	
	Add to Cart	E	
	Select Cash on Delivery	E	
	Search	E	
	tau	B	
	Select Standard Shipping	H	
	Xor(tau, Search)	E	

s

Figure 8: Configuration output for Region B. The table reports quality metrics and the configuration state of each configurable node, where E indicates an enabled activity and H indicates a hidden activity.

The configurable process model is instantiated for each region using the Exhaustive Configurable Process Tree Configuration plugin in ProM. This algorithm applies configuration semantics to a configurable process tree by resolving each configurable element through selection or hiding of branches [15]. Hiding replaces the corresponding activity/branch with a silent (τ) node, meaning that the behavior can be traversed without producing an observable event in the log. Each configured process tree represents a restricted behavioral subset of the original configurable model, obtained according to the selected context.

Figures 7 and 8 report the configuration outputs for Region A and Region B. Each table shows the quality metrics and the state assigned to each configurable node. Enabled nodes remain visible in the configured variant, while hidden nodes are replaced by silent τ-transitions. The results reflect the expected regional constraints: Region A retains Standard Shipping and hides Express Shipping, whereas Region B retains Express Shipping and hides Standard Shipping. Payment-related nodes may appear more than once because the same activity label can belong to different XOR contexts in the merged tree; this does not imply repeated execution within the same trace.

In addition to these configuration tables, the algorithm produces context-specific derived process variants, which represent the concrete configured behavior for each region. These variants are presented in Figures 9 and 10.

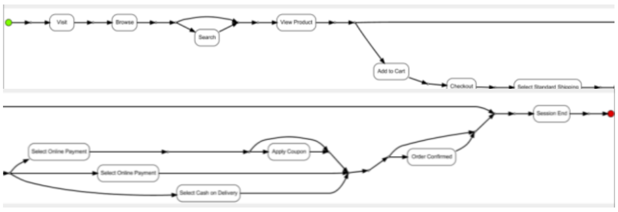

Figure 9: Derived process variant for Region A visualized using the Inductive Visual Miner.

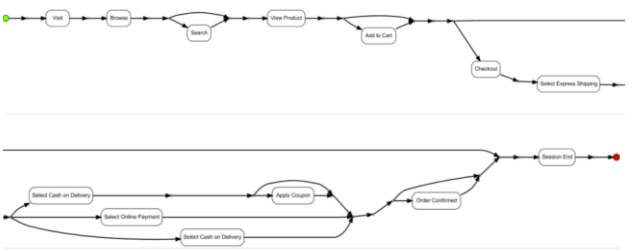

Figure 10: Derived process variant for Region B visualized using the Inductive Visual Miner.

Figures 9 and 10 present the derived process variants obtained after configuration for Region A and Region B, respectively, using an Inductive Visual Miner visualization. This representation highlights the sequence of activities that constitute the customer journey after applying the region-specific configuration.

The derived model for Region A retains Select Standard Delivery, while Select Express Delivery is removed; conversely, Region B retains Select Express Delivery and removes Select Standard Delivery. Hence, the two variants mainly differ at the level of the configurable activities, while the remaining structure stays unchanged, illustrating how configuration activates/deactivates branches to obtain context-specific variants.

In both variants, some payment activities may appear duplicated (e.g., Select Online Payment or Select Cash on Delivery). This duplication is a merge/configuration artifact caused by the fact that the same payment label can occur in different alternative subtrees (XOR scenarios) of the merged configurable process tree. It does not mean that payment is executed twice; it reflects two possible contexts. For readability, the duplicate can be removed/merged in the visualization without affecting correctness.

5.5 Quantitative comparison: fitness and precision

Table 2 presents the quantitative metrics (fitness and precision) computed for both approaches evaluated on region-specific logs. These metrics are defined formally in Section 3.2.2 and provide primary evidence for comparing the modeling efficacy of the two approaches.

Table 2: Quantitative comparison of fitness and precision metrics.

Criterion	**Classical Approach**	**Configurable Approach**
Fitness	0.9438	0.902
Precision	0.7129	0.996
Variability representation	Variability is implicit and embedded in a single global model.	Variability is explicitly modeled and resolved through configuration.
Evaluation Basis	Full combined log (L_all): 1,600 sessions, 7,500 events	Region-specific logs: L_regionA & L_regionB (800 sessions each)

Interpretation of the fitness–precision trade-off.
The results indicate a trade-off between behavioral coverage and model specificity. The classical approach achieves higher fitness (0.9438), which indicates that it can replay most traces in the combined event log. This is expected because the model is discovered from the aggregated log and is therefore designed to cover behavior observed across both regions. However, this broader coverage reduces precision (0.7129), since the model also allows behavioral combinations that are not valid within a specific regional context.

In contrast, the configurable approach achieves higher precision (0.996) with a moderate decrease in fitness (0.902). This improvement in precision is explained by the configuration step, which hides or disables region-irrelevant branches. As a result, each configured variant permits a narrower set of behaviors that better matches the corresponding regional log. The slight loss in fitness reflects the more restrictive nature of the configured models, which may no longer replay some noisy, infrequent, or borderline traces accepted by the global model.

This trade-off supports the use of classical discovery for global process understanding and configurable process mining for context-sensitive analysis, where interpretability and specificity are more important.

5.6 Model structure and interpretability observations

Beyond the quantitative metrics in Table 2, the discovered models show clear structural differences. The classical model integrates all activities from both regions into a single control-flow structure. As a result, region-specific alternatives such as Standard Shipping, Express Shipping, Online Payment, and Cash-on-Delivery appear together in the same model, which may allow behavioral combinations that are not valid within a specific region.

By contrast, the configured variants reduce this ambiguity by removing or hiding region-irrelevant branches. The Region A variant excludes Express Shipping, while the Region B'(((variant excludes Standard Shipping. This makes the configured models structurally clearer and easier to interpret as region-specific customer journey representations.

5.7 Comparative evaluation of analytical criteria

In addition to fitness and precision, Table 3 compares both approaches using qualitative criteria related to variability representation, interpretability, context discrimination, and analytical use case.

Analytical Criterion	Classical Approach	Configurable Approach
Variability Representation	Implicit in one global model	Explicit through configuration points
Model Interpretability	Medium; multiple alternatives mixed	Higher; one configured variant per context
Context Discrimination Capability	Limited; requires post-hoc analysis	High; direct context-specific analysis.
Optimal Use Case	Macro-level overview and audit.	Context-specific optimization and regional analysis

Table 3: Qualitative comparison of analytical criteria for classical and configurable process mining approaches.

The evaluation presented in Tables 2 and 3 confirms that the two approaches address different analytical objectives. The classical approach prioritizes behavioral coverage and is suitable for obtaining a global view of the process. However, its broader structure may reduce precision and make context-specific behavior harder to interpret. The configurable approach prioritizes context-specific clarity by removing or hiding irrelevant branches, which improves precision and interpretability.

Generalization and simplicity were also considered qualitatively. Generalization was not measured quantitatively in this proof-of-concept study because it requires validation on independent event logs. At the qualitative level, the classical model may provide broader macro-level generalization because it includes behavior from both regions. However, this broader scope may also lead to overgeneralization by allowing activity combinations that are not valid in a specific context.

The configured variants are more specialized. They are designed to represent behavior within a specific context rather than to generalize across all contexts. Their generalization should therefore be assessed in future work using unseen logs from the same region or customer segment. Regarding simplicity, the configured variants are structurally clearer because configuration removes context-irrelevant branches. This reduces the number of alternatives to inspect and improves readability for analysts and non-expert stakeholders.

6 Discussion

6.1 Added value of configurable process mining compared to classical discovery

Classical process discovery produces a single global model when event logs from multiple contexts are combined. Although this correctly captures the overall control-flow structure, it implicitly merges behaviors from different regions, causing context-dependent alternatives (e.g., delivery methods, payment options) to appear together even when they never cooccur in reality. This limits the granularity needed for analyzing context-specific process behavior.

Configurable process mining addresses this by explicitly separating shared behavior from context-dependent behavior. Configuration points are inserted at divergence points, allowing region-specific variants that reflect only the behavior present in each context. This prevents the artificial combination of mutually exclusive behavioral paths and yields clearer, more interpretable models.

6.2 Comparative analysis of classical and configurable approaches

The quantitative comparison in Table 2 shows a clear fitness–precision trade-off. The classical model achieves higher fitness (0.9438) because it is discovered from the aggregated log and covers a broader range of regional behaviors. However, this broader coverage reduces precision (0.7129), since the model may allow combinations that are not valid in a specific regional context.

The configurable approach shows the opposite pattern, with higher precision (0.996) and lower fitness (0.902). This is consistent with process mining evaluation principles: more restrictive models may improve precision by limiting allowed behavior, but they can lose some fitness when fewer traces can be replayed. In this study, the configuration step removes context-irrelevant branches, such as region-specific delivery or payment alternatives.

Compared with the works summarized in Table 1, the added value of the proposed approach is the explicit modeling of context-specific variability within a shared process structure. Table 3 further shows that this improves interpretability and context discrimination. The classical approach remains suitable for macro-level process understanding, while the configurable approach is more suitable for context-aware analysis.

6.3 Implications for customer analytics and context-aware recommendation systems

Configurable process mining opens a promising research direction for context-aware recommendation systems. A configured customer journey variant can represent valid behavior for a given context, such as a region, customer segment, or service configuration. In a future recommendation pipeline, the customer context could first be used to select the corresponding configured variant. The active customer trace could then be compared with valid paths in that variant to identify suitable next actions or next touchpoints.

For example, if a Region A customer reaches checkout, recommendations could be generated using behavior consistent with Region A constraints, such as standard shipping and online payment. This avoids recommending actions that are inconsistent with the customer's current context and supports more context-aware journey guidance. The implementation and evaluation of such a recommendation pipeline is planned as part of future work.

6.4 Implications and limitations

The exclusive use of synthetic event logs is a major limitation of this study. Synthetic data enabled controlled experimentation because the underlying process, contextual rules, and expected variants were known in advance. However, this restricts external validity, since real customer journey logs may contain weaker contextual separation, overlapping behaviors, incomplete sessions, rare variants, and unexpected deviations.

Real event logs may also include bot-generated sessions, web crawlers, repeated page refreshes, network-related loops, duplicated events, timestamp inconsistencies, and technical events that do not correspond to meaningful customer behavior. Future validation on operational logs will therefore require preprocessing to remove technical noise, artificial loops, bot activity, incomplete traces, and duplicated events before process discovery.

Consequently, the results should be interpreted as proof-of-concept evidence rather than definitive operational validation. Future work will test the approach on real customer journey logs, additional contextual dimensions, alternative discovery algorithms, different configuration strategies, and larger-scale datasets.

7 Conclusions

This study examined the contribution of configurable process mining to customer journey analysis in the presence of contextual variability. The results indicate that configurable process mining can distinguish shared journey behavior from context-specific behavior while preserving a common process structure. In the simulated e-commerce setting, regional differences related to delivery and payment options were explicitly represented through configured variants.

The comparison with classical process discovery highlights a clear fitness–precision trade-off. The classical model achieved higher fitness (0.9438), confirming its suitability for macro-level process understanding. However, the configured variants achieved substantially higher precision (0.996), showing their ability to reduce overgeneralization and avoid artificial combinations of mutually exclusive regional paths. These results suggest that configurable process mining is particularly useful when the objective is to analyze context-specific behavior and improve interpretability.

The study is limited by the exclusive use of synthetic event logs. Although this choice enabled controlled experimentation and access to known process rules, it restricts the generalizability of the findings. Real customer journey logs may contain additional sources of complexity, including web crawlers, bot-generated sessions, repeated page refreshes, network-related loops, duplicated events, incomplete traces, and timestamp inconsistencies.

Future work will validate the proposed approach using real customer journey event logs from e-commerce or digital service platforms. This validation will require a preprocessing phase to remove technical noise, artificial loops, bot activity, incomplete traces, and duplicated events before applying process discovery.

Further experiments will assess whether the precision advantage observed in the synthetic experiment is preserved on real logs and whether the assumptions of Proposition 1 hold under operational conditions. Future work will also include statistical validation, noise sensitivity analysis, quantitative simplicity measures, and robustness tests using alternative discovery algorithms, different configuration strategies, and larger-scale datasets. The analysis will also be extended beyond geographic region by considering additional contextual dimensions, such as customer segment, device type, product category, seasonality, marketing campaign exposure, and service availability.

7.1 Appendix

Appendix A. Formal statement and proof of Proposition 1

Let Σ denote the finite set of activities. Let $L_i$be a context-specific event log and $L_m = \uplus_i L_i$the aggregated log. Classical discovery produces a global model $M_{glob} = \delta(L_m)$. Configurable discovery produces a configurable model M_{conf}, from which a context-specific model $M_{conf,i} = C(M_{conf}, c_i)$ is derived.

For a given context i, assume that:
H1. M_{glob}covers the context-specific log L_i.
H2. $M_{conf,i}$restricts the behavior allowed by M_{glob}, i.e.,

$\mathcal{L}(M_{conf,i}) \subseteq \mathcal{L}(M_{glob})$.
H3. $M_{conf,i}$covers the context-specific log L_i.

Under these assumptions, the configured model allows a smaller behavioral space while preserving the observed behavior of L_i. Since precision measures the degree to which the behavior allowed by the model corresponds to the behavior observed in the log, reducing the allowed behavior while preserving coverage leads to non-decreasing precision. Therefore:

$$precision(M_{conf,i}, L_i) \geq precision(M_{glob}, L_i)$$

References

[1] K. N. Lemon and P. C. Verhoef, "Understanding Customer Experience Throughout the Customer Journey," *Journal of Marketing*, vol. 80, no. 6, pp. 69–96, Nov. 2016, doi: 10.1509/jm.15.0420.

[2] M. S. Rosenbaum, M. L. Otalora, and G. C. Ramírez, "How to create a realistic customer journey map," *Business Horizons*, vol. 60, no. 1, pp. 143–150, Jan. 2017, doi: 10.1016/j.bushor.2016.09.010.

[3] W. Van Der Aalst, "Process mining," *Commun. ACM*, vol. 55, no. 8, pp. 76–83, Aug. 2012, doi: 10.1145/2240236.2240257.

[4] S. J. J. Leemans, D. Fahland, and W. M. P. Van Der Aalst, "Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour," in *Business Process Management Workshops*, vol. 171, N. Lohmann, M. Song, and P. Wohed, Eds., in Lecture Notes in Business Information Processing, vol. 171. , Cham: Springer International Publishing, 2014, pp. 66–78. doi: 10.1007/978-3-319-06257-0_6.

[5] I. Mandic, "ENHANCING CUSTOMER JOURNEY THROUGH PROCESS MINING: A LITERATURE REVIEW AND FUTURE RESEARCH DIRECTIONS," in *Proceedings–The First International Conference FUTURE-BME 2024 (Forging the Future: Pioneering Approaches in Business, Management and*, p. 987. Accessed: Dec. 02, 2025. [Online]. Available: https://www.researchgate.net/profile/Tamara-Gajic-2/publication/388626197_Permanent_Involvement_of_Digitization_and_Modern_Technology_in_the_Direction_of_Sustainable_Growth_in_the_Tourism_and_Hotel_Industry_of_Serbia/links/679f4e408311ce680c4b7f7f/Permanent-Involvement-of-Digitization-and-Modern-Technology-in-the-Direction-of-Sustainable-Growth-in-the-Tourism-and-Hotel-Industry-of-Serbia.pdf#page=1014

[6] R. Halvorsrud, F. Mannhardt, O. Prillard, and C. Boletsis, "Customer journeys and process mining–challenges and opportunities," in *ITM Web of Conferences*, EDP Sciences, 2024, p. 05002. Accessed: Dec. 02, 2025. [Online]. Available: https://research.tue.nl/en/publications/customer-journeys-and-process-mining-challenges-and-opportunities

[7] B. Topaloglu, B. Oztaysi, and O. Dogan, "Cart-state-aware discovery of e-commerce visitor journeys with process mining," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 19, no. 4, pp. 2851–2879, 2024.

[8] A. Terragni and M. Hassani, "Analyzing Customer Journey with Process Mining: From Discovery to Recommendations," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug. 2018, pp. 224–229. doi: 10.1109/FiCloud.2018.00040.

[9] A. Terragni and M. Hassani, "Optimizing customer journey using process mining and sequence-aware recommendation," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, Limassol Cyprus: ACM, Apr. 2019, pp. 57–65. doi: 10.1145/3297280.3297288.

[10] I. Shafei, J. Karnon, and M. Crotty, "Process mining and customer journey mapping in healthcare: Enhancing patient-centred care in stroke rehabilitation," *DIGITAL HEALTH*, vol. 10, p. 20552076241249264, Jan. 2024, doi: 10.1177/20552076241249264.

[11] G. Bernard and P. Andritsos, "A process mining based model for customer journey mapping," in *Forum and doctoral consortium papers presented at the 29th international conference on advanced information systems engineering (CAiSE 2017)*, CEUR Workshop Proceedings, 2017, pp. 49–56. Accessed: Dec. 02, 2025. [Online]. Available: https://serval.unil.ch/resource/serval:BIB_6FD456C4AB58.P001/REF.pdf

[12] M. La Rosa, M. Dumas, R. Uba, and R. Dijkman, "Business Process Model Merging: An Approach to Business Process Consolidation," *ACM Trans. Softw. Eng. Methodol.*, vol. 22, no. 2, pp. 1–42, Mar. 2013, doi: 10.1145/2430545.2430547.

[13] F. Gottschalk, W. M. P. Van Der Aalst, M. H. Jansen-Vullers, and M. La Rosa, "CONFIGURABLE WORKFLOW MODELS," *Int. J. Coop. Info. Syst.*, vol. 17, no. 02, pp. 177–221, Jun. 2008, doi: 10.1142/S0218843008001798.

[14] M. L. Rosa, W. M. P. V. D. Aalst, M. Dumas, and F. P. Milani, "Business Process Variability Modeling: A Survey," *ACM Comput. Surv.*, vol. 50, no. 1, p. 2:1-2:45, Mar. 2017, doi: 10.1145/3041957.

[15] N. Assy, W. Gaaloul, and B. Defude, "Mining Configurable Process Fragments for Business Process Design," in *Advancing the Impact of Design Science: Moving from Theory to Practice*, vol. 8463, M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta, and V. Yoon, Eds., in Lecture Notes in Computer Science, vol. 8463. , Cham: Springer International Publishing, 2014, pp. 209–224. doi: 10.1007/978-3-319-06701-8_14.

[16] E. B. Varga and A. Baksa, "Evaluating Process Discovery From Loop Structures," *Production Systems and Information Engineering*, vol. 12, no. 2, pp. 23–39, Aug. 2024, doi: 10.32968/psaie.2024.2.2.

[17] A. Bogarín, R. Cerezo, and C. Romero, "Discovering learning processes using Inductive Miner: A case study with Learning Management Systems (LMSs)," *Psicothema*, vol. 3, no. 30, pp. 322–329, Aug. 2018, doi: 10.7334/psicothema2018.116.

[18] A. Augusto *et al.*, "Automated Discovery of Process Models from Event Logs: Review and Benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 686–705, Apr. 2019, doi: 10.1109/TKDE.2018.2841877.

[19] A. J. Weijters, W. M. van Der Aalst, and A. A. De Medeiros, "Process mining with the HeuristicsMiner algorithm," 2006, Accessed: Dec. 02, 2025. [Online]. Available: https://research.tue.nl/files/2388011/615595.pdf

[20] J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. Van Der Aalst, "On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery," in *On the Move to Meaningful Internet Systems: OTM 2012*, vol. 7565, R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz, Eds., in Lecture Notes in Computer Science, vol. 7565. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 305–322. doi: 10.1007/978-3-642-33606-5_19.

[21] A. Augusto, A. Armas-Cervantes, R. Conforti, M. Dumas, and M. La Rosa, "Measuring fitness and precision of automatically discovered process models: A principled and scalable approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1870–1888, 2020.

[22] J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. Van Der Aalst, "Quality Dimensions in Process Discovery: The Importance of Fitness, Precision, Generalization and Simplicity," *Int. J. Coop. Info. Syst.*, vol. 23, no. 01, p. 1440001, Mar. 2014, doi: 10.1142/S0218843014400012.

[23] A. Augusto, R. Conforti, M. Dumas, and M. La Rosa, "Split Miner: Discovering Accurate and Simple Business Process Models from Event Logs," in *2017 IEEE International Conference on Data Mining (ICDM),* New Orleans, LA: IEEE, Nov. 2017, pp. 1–10. doi: 10.1109/ICDM.2017.9.

[24] E. Chlebus and J. Brazier, "Nonstationary poisson modeling of web browsing session arrivals," *Information Processing Letters*, vol. 102, no. 5, pp. 187–190, 2007.

[25] N. Blenn and P. V. Mieghem, "Are human interactivity times lognormal?," Jul. 11, 2016, *arXiv*: arXiv:1607.02952. doi: 10.48550/arXiv.1607.02952.

[26] S. Suriadi, R. Andrews, A. H. ter Hofstede, and M. T. Wynn, "Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs," *Information systems*, vol. 64, pp. 132–150, 2017.

[27] H. M. Marin-Castro and E. Tello-Leal, "Event log preprocessing for process mining: a review," *Applied Sciences*, vol. 11, no. 22, p. 10556, 2021.

[28] R. Conforti, M. La Rosa, and A. H. Ter Hofstede, "Filtering out infrequent behavior from business process event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 300–314, 2016.