Segmentation for Optimizing Long-Lifecycle Processes in Object-Centric Process Mining

Alessandro Berti[1], Dina Kretzschmann[1], and Wil M.P. van der Aalst[1,2]

[1] Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany
[2] Celonis, Munich, Germany
{a.berti, dina.kretzschmann, wvdaalst}@pads.rwth-aachen.de

Abstract. Object-centric process mining enables the analysis of complex business processes involving multiple interacting objects, such as in inventory management or e-business workflows. However, long object lifecycles often result in convoluted event logs, obscuring actionable insights and challenging process mining techniques. This paper introduces a segmentation framework that decomposes object-centric event logs into focused, analytically tractable units called segments. We propose segmentation strategies based on object relationships, event types, and temporal intervals, and integrate them with machine learning techniques for anomaly detection and correlation analysis. Implemented in the OC-PM tool, our approach empowers users to uncover localized patterns and optimize process execution. A case study on inventory management demonstrates how segmentation reveals insights into understock and overstock issues, enhancing decision-making in data-intensive domains.

Keywords: Object-Centric Event Logs · Segmentation Strategies · Business Process Optimization · Inventory Management

1 Introduction

Business processes in domains like finance, e-business, and healthcare increasingly involve multiple interacting entities (orders, items, customers, or patients) generating complex event logs that defy traditional process mining's single-case paradigm [14]. Object-centric process mining (OCPM) addresses this by modeling events linked to multiple objects, offering a richer view of real-world workflows such as Purchase-to-Pay (P2P) and Order-to-Cash (O2C). Yet, when objects have long lifecycles spanning numerous events, the resulting data complexity overwhelms standard techniques, obscuring insights critical for optimization [5]. For example, in inventory management, a material might interact with thousands of purchase orders over years, involving events like "Create Purchase Order Item" and "Goods Receipt". A single material's lifecycle could span hundreds of such events, with interleaved stock adjustments (e.g., "Goods Issue" for sales), making it nearly impossible to isolate patterns like order delays or stock depletion risks using traditional methods.

To tackle this challenge, we propose a segmentation framework for OCPM, decomposing event logs into manageable units called segments. Unlike traditional flattening approaches, our method defines segments based on object relationships, event types, or temporal intervals. Implemented in the OC-PM tool, this approach enables focused control-flow analysis, anomaly detection, and correlation discovery, transforming raw data into actionable business insights. In the inventory example, segmentation can isolate a material's interactions with a specific purchase order, revealing how order delays contribute to understock situations, thus optimizing resource allocation.

This paper addresses three questions: (1) How can segmentation strategies effectively partition object-centric event logs? (2) How do these segments enhance process optimization through AI analytics? (3) What practical benefits emerge in real-world settings like inventory management? Our contributions include a conceptual framework, an AI-integrated OC-PM extension, and a case study showcasing process improvements.

2 Related Work

This section situates our segmentation framework within the broader landscape of process mining, focusing on object-centric approaches, UI log mining, and computational intelligence applications. We highlight how our work complements and extends existing methodologies, particularly in optimizing complex business processes.

Object-Centric Process Mining: OCPM emerged to address limitations of traditional case-centric mining by modeling events linked to multiple objects [14]. In [15], the discovery of object-centric Petri nets from event logs is proposed, capturing interactions across object types like orders and items. However, such models often become complex when objects exhibit long lifecycles, prompting methods to simplify analysis. In [16], silent objects enhance Petri net precision by accounting for unobserved influences, while [11] introduces a temporal object type model (TOTeM) to capture object-type-level relations (e.g., order-to-invoice precedence). Similarly, [13] proposes object-centric directly-follows graphs (DFGs) for understanding object-centric behavior, and [10] offers granularity adjustments (e.g., drill-down) to focus analysis. Unlike these holistic approaches, our segmentation framework partition logs into focused units, enabling localized insights without requiring new modeling formalisms. By integrating machine learning, we extend OCPM's analytical flexibility, aligning with business optimization goals.

UI Log Mining and Task Segmentation: Process mining on user interface (UI) logs, or task mining, tackles unstructured event streams from user interactions, often for robotic process automation (RPA). In [8], routines are discovered from UI sequences, while [3] employs trace alignment to segment logs into task traces. Similarly, [2] develops an interactive segmentation tool to handle interleaved actions and noise. These methods share our goal of partitioning complex data but focus on sequential user tasks rather than concurrent object interactions. Our approach adapts segmentation to OCPM's multi-object context, managing concurrency, revealing process dynamics, and offering a novel lens for business process analysis beyond RPA.

Computational Intelligence in Process Mining: Computational intelligence enhances process mining by extracting actionable insights from event data. In [9], the IODDA algorithm mines decision rules from object-centric logs, linking object attributes to control-flow decisions. In [1], multivariate anomaly detection identifies deviations in object-centric data, while [4] extracts numerical features for machine learning. Our framework builds on these by embedding AI within segmentation, using techniques like anomaly detection and correlation analysis on segmented logs. This synergy amplifies OCPM's optimization potential, distinguishing our work from global modeling or decision-focused methods.
Positioning and Contribution: Prior OCPM techniques prioritize comprehensive models [11, 15], risking complexity, or adjust granularity post-discovery [10]. UI log mining segments sequential data for automation [2], not multi-object processes. Computational intelligence methods enhance analysis but rarely partition event logs [9]. Our segmentation bridges these gaps, offering a preprocessing lens that simplifies OCPM while leveraging machine learning for optimization. Applied to inventory management, it uncovers localized patterns (e.g., stock correlations), advancing smart business modeling for domains like e-business and finance.

3 Background

This section presents the minimal notations for our segmentation framework in OCPM, emphasizing temporal event data for analyzing complex business processes. An *object-centric event log* records events tied to multiple objects, moving beyond traditional single-case mining [14]. Formally, it is a tuple $L = (E, O, T_E, T_O, \mathit{evtype}, \mathit{objtype}, \mathit{rel}, \mathit{time})$, where:

- E is a set of events (e.g., "Create Purchase Order Item");
- O is a set of objects (e.g., a material or purchase order);
- T_E is a set of event types;
- T_O is a set of object types (e.g., "Material", "Purchase Order");
- $\mathit{evtype} : E \to T_E$ maps each event to its event type;
- $\mathit{objtype} : O \to T_O$ assigns each object to its object type;
- $\mathit{rel} : E \to \mathcal{P}(O)$ relates each event to a set of objects;
- $\mathit{time} : E \to \mathbb{R}^+$ assigns each event a timestamp, a positive real number representing its occurrence time.

For instance, in an inventory log, an event $e \in E$, such as event `e4` ("Goods Receipt") in Table 1, relates to a material $M1$ ($\mathit{objtype}(M1)$ = "Material") and a purchase order item $PO1$ ($\mathit{objtype}(PO1)$ = "Purchase Order (Item)"), with $\mathit{rel}(e4) = \{M1, PO1\}$. This aligns with the OCEL standard [7], supporting object-centric, time-ordered analysis. Table 1 illustrates such an event log, showing events like "Create Purchase Order Item" and "Goods Issue" tied to a material and purchase orders. Additionally, Table 2 presents numerical features extracted from this log, such as event counts and object interactions, enabling machine learning applications.

Objects in O have lifecycles over events in $\{e \in E \mid o \in \mathit{rel}(e)\}$, ordered by *time*, often spanning long periods in practice (e.g., materials in inventory). We

Ev.ID	Event Activity	Material	PO (Item)
e1	Create Purchase Order Item	M1	PO1
e2	Goods Issue	M1	
e3	Goods Issue	M1	
e4	Goods Receipt	M1	PO1
e5	Goods Issue	M1	
e6	Goods Issue	M1	
e7	Create Purchase Order Item	M1	PO2
e8	Goods Issue	M1	
e9	Goods Issue	M1	
e10	Goods Issue	M1	
e11	Create Purchase Order Item	M1	PO3
e12	Goods Issue	M1	
e13	Goods Receipt	M1	PO3
e14	Delete Purchase Order Item	M1	PO2
e15	Goods Issue	M1	

Table 1: Example object-centric event log, having *Material* as the lead object type.

Object Type	Object	# of CPO Item	# of GI	# of GR	# of DPOI	# of Mat. Interactions	# of PO Interactions
Material	M1	3	9	2	1	-	3
Purchase Order	PO1	1	-	1	-	1	-
Purchase Order	PO2	1	-	-	1	1	-
Purchase Order	PO3	1	-	1	-	1	-

Table 2: Numeric features for the object-centric event log, including *Material* and *Purchase Order (Item)* objects.

introduce *segmentation* to partition L into manageable units. A *segment* is a subset $S \subseteq E \times O$ where $(e, o) \in S \implies o \in \text{rel}(e)$, capturing a focused log portion (e.g., events for one purchase order) as a bipartite representation of a sub-log. We store segments as pairs (e, o), so an event e may appear multiple times, once per object it involves. A segmentation strategy $Seg : L \to \mathcal{P}(\mathcal{P}(E \times O))$ partitions L into a collection $C = \{S_1, S_2, \ldots, S_n\}$, where each segment $S_i \subseteq E \times O$ satisfies $(e, o) \in S_i \Rightarrow o \in rel(e)$.

Key to our approach is the *lead object type*, a type $t_L \in T_O$ (e.g., "Material") with a one-to-many relationship to other types. Formally, each object o_c of a *child object type* $t_C \in T_O$ (e.g., "Purchase Order") relates to exactly one o_L with $objtype(o_L) = t_L$, while o_L may link to multiple o_c. This guides temporal segmentation (e.g., segments per purchase order tied to a material). Events and objects may have *attributes* (e.g., quantities), supporting AI-driven feature extraction [4].

Object-Based Feature Extraction: To enable AI-driven analysis, we transform an object-centric event log $L = (E, O, T_E, T_O, evtype, objtype, rel, time)$ into an *object-based feature table*, where each row corresponds to an object $o \in O$. Columns represent numerical features derived from the log, capturing object behavior for machine learning tasks like anomaly detection [4].

For an object o, features are computed from its lifecycle $\{e \in E \mid o \in rel(e)\}$. Simple features include:

- *Event counts*: Number of events of a type $t_E \in T_E$ involving o (e.g., count of "Goods Receipt" events for a material).
- *Temporal spans*: Time difference between the earliest and latest events, $\max(\{time(e) \mid o \in rel(e)\}) - \min(\{time(e) \mid o \in rel(e)\})$.
- *Interactions*: Count of distinct objects $o' \in O$ (where $o' \neq o$) sharing events with o, i.e., $|\{o' \mid \exists e : \{o, o'\} \subseteq rel(e)\}|$.

For example, a material $M1$ has 2 "Goods Receipt" events, a lifecycle span from its earliest to latest events, and interactions with 3 purchase order items

($PO1$, $PO2$, $PO3$), as shown in Table 2. Attributes (e.g., quantities) can also be aggregated (e.g., total quantity received). This table, easily constructed via queries on L, supports advanced analytics by converting event sequences into numerical vectors.

4 Approach

Complex object lifecycles in OCPM create significant challenges when trying to extract actionable insights from event logs. To overcome this complexity, we propose segmenting event logs into smaller, more focused subsets. These segments target specific relationships between objects, key event types, or critical time intervals, enabling a more intuitive and effective analysis of processes.

4.1 Segmentation Strategies and Visualization

We propose three main segmentation strategies, each designed to simplify the analysis by focusing on distinct aspects of the event logs:

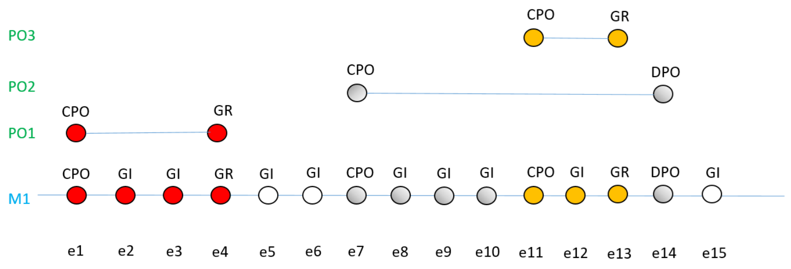
- **Lead-Child Segmentation:** This strategy focuses on one main object type (the *lead object*) and segments the log according to its relationships with secondary object types (*child objects*). For example, in an inventory scenario, segments can isolate activities related to a specific material and individual purchase orders linked to it. This approach helps clearly identify events correlated to specific orders, making it easier to pinpoint delays or anomalies.
- **Event-Type-Based Segmentation:** Here, segments are created based on specific key events. For instance:
  - *Segments starting from a specific event type:* Events occurring after each instance of a key event (like a "Goods Receipt") until the next occurrence are grouped. This is particularly useful for tracking what happens immediately following critical activities.
  - *Segments ending at a specific event type:* Events leading up to and including a key event are grouped. For example, analyzing activities before each "Goods Receipt" can highlight what contributes to delays.
- **Temporal Interval-Based Segmentation:** Segments are defined by intervals between two specific event types, capturing complete process phases. For example, considering intervals between "Create Purchase Order Item" and "Goods Receipt" makes it easier to analyze the efficiency and timeliness of each procurement cycle.

Figure 1 visually demonstrates these strategies using simplified event names (e.g., "CPO" for "Create Purchase Order Item", "GR" for "Goods Receipt"). Each subfigure clarifies how different strategies segment the log into clearly defined subsets, facilitating easier and more meaningful analyses.

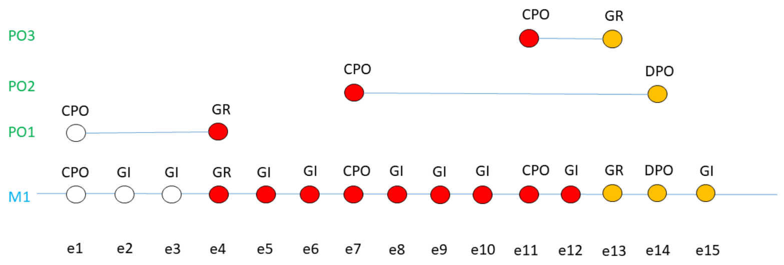
4.2 Improved Activity Sequence Analysis

Segmentation significantly simplifies the analysis of activity sequences by clearly separating relevant activities:

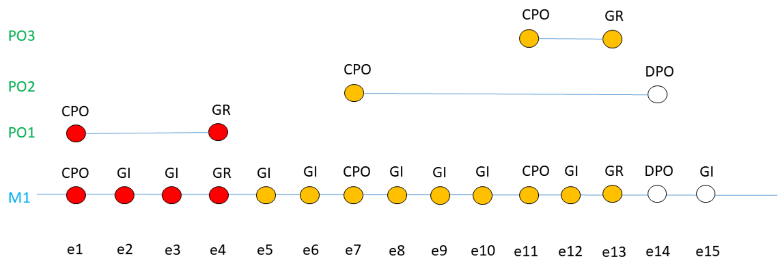
- Without segmentation, event sequences are complex, mixing multiple purchase orders and unrelated events, leading to confusing process flows.


PO3
PO2
PO1
M1
CPO
GR
CPO
DPO
CPO
GR
CPO GI GI GR GI GI CPO GI GI GI CPO GI GR DPO GI
e1 e2 e3 e4 e5 e6 e7 e8 e9 e10 e11 e12 e13 e14 e15

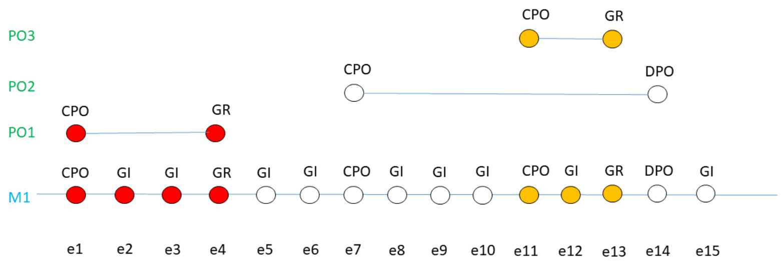
(a) Segments corresponding to the child objects of type purchase order.


PO3
PO2
PO1
M1
CPO
GR
CPO
DPO
CPO
GR
CPO GI GI GR GI GI CPO GI GI GI CPO GI GR DPO GI
e1 e2 e3 e4 e5 e6 e7 e8 e9 e10 e11 e12 e13 e14 e15

(b) Segments from the event type “Goods Receipt”.


PO3
PO2
PO1
M1
CPO
GR
CPO
DPO
CPO
GR
CPO GI GI GR GI GI CPO GI GI GI CPO GI GR DPO GI
e1 e2 e3 e4 e5 e6 e7 e8 e9 e10 e11 e12 e13 e14 e15

(c) Segments to the event type “Goods Receipt”.


PO3
PO2
PO1
M1
CPO
GR
CPO
DPO
CPO
GR
CPO GI GI GR GI GI CPO GI GI GI CPO GI GR DPO GI
e1 e2 e3 e4 e5 e6 e7 e8 e9 e10 e11 e12 e13 e14 e15

(d) Segments between the event types “Create Purchase Order Item” and “Goods Receipt”.

Fig. 1: Proposed segmentation strategies applied in an inventory management context.

– Applying segmentation, such as the lead-child approach, isolates event sequences tied specifically to individual orders. This reveals simpler, clearer flows—such as a straightforward progression from order creation, through

N.	Type	Lead Object Type	Child Object Type	Source Event Type	Target Event Type	Motivation
1)	Between	Material		Create Purchase Order Item	Goods Receipt	Understanding when the orders are placed too late by discovering understock in the given segments.
2)	Between	Material		Create Purchase Order Suggestion Item	Create Purchase Order Item	Understanding user delays in the order placement phase and alternative reasons for the placement of the purchase orders.
3)	Lead+Child	Material	Purchase Order			Understanding unexpected splits of the receipt of the goods for a given purchase order, resulting potentially in understock situations.
4)	Lead+Child	Material	Sales Order			Identifying orders connected with several goods issues (periodic contracts). Understanding unexpected splits of the goods issue with regards to the initial requested quantity.
5)	To	Material			Create Purchase Order Item	Understanding the reasons leading to triggering a purchase order (such as suggestion from the predictive system, response to sales orders).
6)	From	Material		Goods Receipt		Identifying goods receipts resulting in an initial overstock after the receipt. Observing patterns leading to understock.
7)	To	Material			Goods Receipt	Understanding the correlation between stock status and the number of goods issues.
8)	Between	Supplier		Create Purchase Order Item	Goods Receipt	Assessing the predictability of single suppliers in the provision of the goods.
9)	Between	Customer		Create Sales Order Item	Goods Issue	Assessing customer satisfaction based on prompt responses to sales orders.

Table 3: Different segmentation strategies applicable to inventory management. Each segmentation is motivated by different analytical needs.

goods issuance, to receipt—making patterns like stock depletion immediately evident.

Such clear sequences enable businesses to identify and directly address specific issues, like reorder delays or inventory shortages, enhancing operational efficiency.

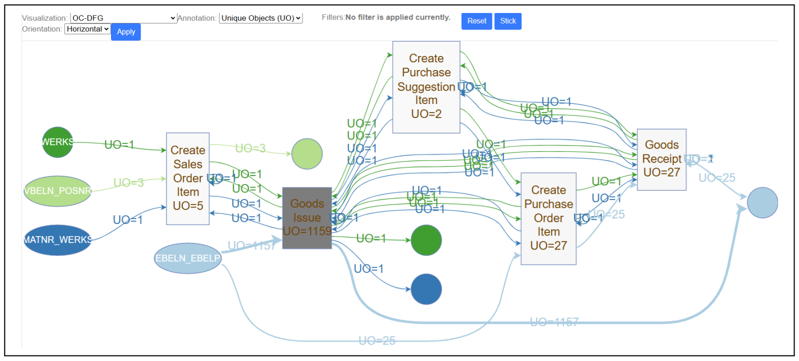
Figures 2a and 2b illustrate this transformation. Figure 2a shows the object-centric directly-follows graph (DFG) of the complete event log before segmentation, where numerous overlapping relationships and activities obscure meaningful patterns. In contrast, Figure 2b shows the DFG after applying segmentation based on object interactions. The resulting structure is significantly more interpretable, emphasizing localized behaviors and process fragments relevant to specific object instances, which enhances the clarity and accuracy of downstream analyses.

4.3 Enhanced Machine Learning Insights

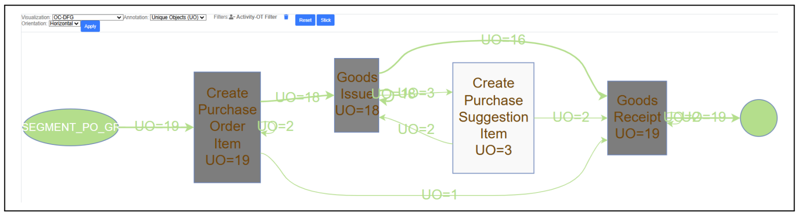
Segmentation also improves the clarity and effectiveness of machine learning analyses:

- Without segmentation, features used for machine learning - like event counts or lifecycle durations - combine unrelated events, resulting in noisy and less insightful analyses.
- With segmentation, each segment provides more meaningful and context-specific features. For example, features computed within a procurement cycle between order creation and receipt clearly indicate stock usage patterns and inventory risks.

Segmented data leads to stronger correlations and more accurate anomaly detection. Businesses can thus precisely target optimization strategies, such as adjusting reorder points or expediting deliveries during identified risk periods.


Visualization: OC-DFG
Annotation: Unique Objects (UO)
Orientation: Horizontal
Apply
Filters: No filter is applied currently.
Reset
Stick
Create Purchase Suggestion Item UO=2
Create Sales Order Item UO=5
Goods Issue UO=1159
Create Purchase Order Item UO=27
Goods Receipt UO=27
WERKS
VBELN_POSNR
MATNR_WERKS
EBELN_EBELP
UO=1
UO=3
UO=1157
UO=25

(a) Overall object-centric DFG (as shown by the OC-PM tool) computed before segmentation.



(b) Object-centric DFG (as shown by the OC-PM tool) filtered on the segments object type.

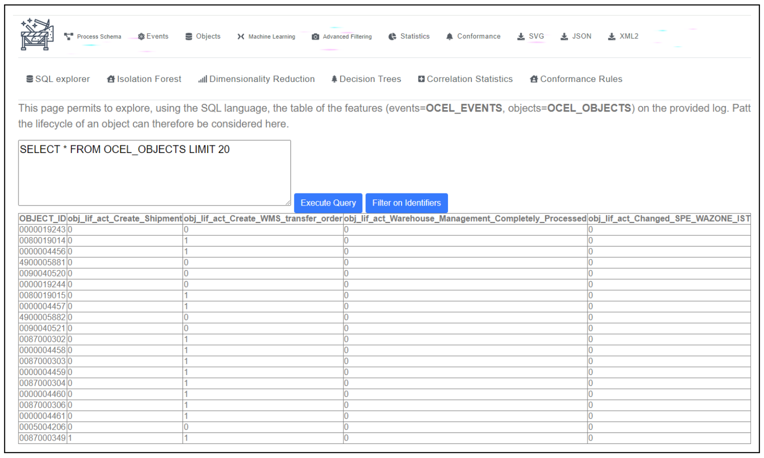
Fig. 2: Combined view of object-centric DFGs before and after segmentation.

5 Implementation

To facilitate the practical application of our segmentation framework, we integrated the proposed strategies into the OC-PM tool, a web-based platform designed specifically for object-centric process mining (available at `https://www.ocpm.info/`). The OC-PM tool supports object-centric event logs in the OCEL 2.0 format and provides functionalities for filtering, analyzing, and visualizing process data.

We enhanced OC-PM by adding a segmentation module accessible through the existing interface, allowing users to easily apply the segmentation strategies outlined earlier. The module provides three straightforward segmentation options:

- **Lead-Child Segmentation:** Users choose a primary object type (e.g., “Material”) and a related secondary type (e.g., “Purchase Order”). The tool automatically segments the log into subsets corresponding to each unique pair, such as individual materials and an associated purchase order.
- **Event-Type-Based Segmentation:** Users select a particular event type (e.g., “Goods Receipt”) and decide whether to segment events occurring before or after each occurrence of this event type. This helps users focus on activities immediately surrounding key business events.

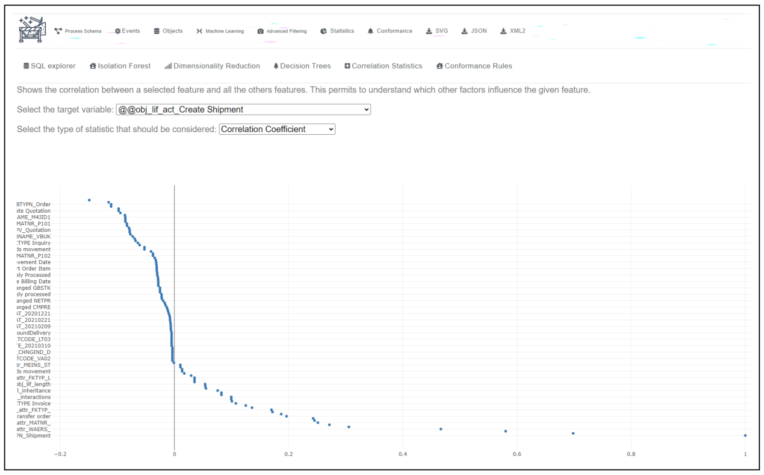


This page permits to explore, using the SQL language, the table of the features (events=**OCEL_EVENTS**, objects=**OCEL_OBJECTS**) on the provided log. Patt
the lifecycle of an object can therefore be considered here.

```
SELECT * FROM OCEL_OBJECTS LIMIT 20
```

OBJECT_ID	obj_lif_act_Create_Shipment	obj_lif_act_Create_WMS_transfer_order	obj_lif_act_Warehouse_Management_Completely_Processed	obj_lif_act_Changed_SPE_WAZONE_IST
0000019243	0	0	0	0
0080019014	0	1	0	0
0000004456	0	1	0	0
4900005881	0	0	0	0
0090040520	0	0	0	0
0000019244	0	0	0	0
0080019015	0	1	0	0
0000004457	0	1	0	0
4900005882	0	0	0	0
0090040521	0	0	0	0
0087000302	0	1	0	0
0000004458	0	1	0	0
0087000303	0	1	0	0
0000004459	0	1	0	0
0087000304	0	1	0	0
0000004460	0	1	0	0
0087000306	0	1	0	0
0000004461	0	1	0	0
0005004206	0	0	0	0
0087000349	1	1	0	0

(a) SQL explorer of the numerical features (as shown by the OC-PM tool), allowing for filtering/aggregation.



Shows the correlation between a selected feature and all the others features. This permits to understand which other factors influence the given feature.

Select the target variable: @@obj_lif_act_Create Shipment

Select the type of statistic that should be considered: Correlation Coefficient

(b) Correlation statistics between the different features.

Fig. 3: Some ML features offered in the OC-PM tool.

– **Temporal Interval-Based Segmentation:** Users define segments based on the intervals between two specific event types (e.g., from “Create Purchase Order Item” to “Goods Receipt”). This clearly isolates complete phases of the process, such as procurement cycles.

Once segments are generated, OC-PM adds them as new “Segment” objects within the log, allowing users to apply further analyses or filtering directly within the tool.

The segmentation module significantly enhances OC-PM’s analysis capabilities. Users can visualize simplified and focused process flows for each segment, greatly improving the readability of complex event logs. Additionally, users can query segmented logs using an intuitive interface to explore specific subsets of interest, aggregate data, or investigate specific issues such as procurement delays or stock depletion.

Moreover, OC-PM provides integrated support for machine learning analyses [4, 12]. It automatically computes relevant features from each segment, such as event counts or durations, enabling correlation analyses and anomaly detection directly within segmented logs. This integration allows users to easily identify and act upon patterns or anomalies that might be hidden in a non-segmented view.

6 Case Study

We evaluated the effectiveness of our segmentation approach through a case study involving inventory management at a retail company. The company faced challenges with maintaining optimal stock levels, frequently experiencing understock and overstock issues. Their goal was to improve inventory efficiency by gaining clearer insights into the underlying causes of these issues.

The case study utilized a real-world object-centric event log that combined processes from purchasing (Purchase-to-Pay) and sales (Order-to-Cash). The event log includes various types of objects, such as materials, suppliers, customers, and purchase orders, along with event attributes like quantities and stock levels. A simplified example of the data is provided in Table 4, where specific thresholds for safety stock (SS) and overstock (OS) are defined.

Initially, analyzing the complete, unsegmented event log proved challenging due to its complexity and the overlapping lifecycles of multiple purchase orders and sales events. The resulting visualizations and analyses lacked clarity, making it difficult to pinpoint precise issues affecting inventory levels.

We applied several segmentation strategies from the OC-PM tool to address this complexity:

– **Lead-Child Segmentation (Material-Purchase Order):** By segmenting events related to each specific material-purchase order pair, we isolated the activities correlated to individual orders. This segmentation clarified the flow of each order cycle, clearly identifying events such as goods issues and receipts for each order.

Transformed Event Type	**Material**	**Supplier**	**Customer**	**PO Item**	**Quantity**	**Stock Before**	**Stock After**
Goods Issue	M001		C001		2	100	98
Goods Issue	M001		C002		3	98	95
Create Purchase Suggestion Item	M001	S001			100	95	95
Create Purchase Order Item	M001	S001		PO1	100	95	95
Goods Issue	M001		C003		5	95	90
Goods Receipt	M001	S001		PO1	50	90	140
Goods Receipt	M001	S001		PO1	50	140	190
Goods Issue	M001		C001		10	190	180

Table 4: Simplified extract of the object-centric event log considered in the case study. We assume that for the material $M001$, $SS = 91$ and $OS = 150$. The event log is available at `https://zenodo.org/records/13347782`.

- **Temporal Interval-Based Segmentation (from order creation to goods receipt):** This approach focused specifically on procurement phases. It clearly highlighted procurement timelines and helped identify periods where delays caused stock to fall below safety thresholds or exceed overstock levels.

Applying these segmentations yielded significant clarity improvements, revealing key insights for inventory management. Segmentation clearly identified instances where goods were ordered prematurely or in excessive quantities, leading to significant overstock situations after receipt, while also isolating procurement cycles to detect specific cases where delayed orders caused stock to fall below safe thresholds, thus highlighting critical reordering issues. Moreover, segmented data enhanced machine learning analyses, clarifying correlations between goods issues and stock levels with strong negative correlations in specific procurement cycles that were previously obscured in the complete dataset, and enabled precise identification of anomalous stock movements, such as unusually large goods issues or unexpectedly high stock levels after receipts, which are crucial for accurate inventory management.

Overall, segmentation significantly improved the company's ability to identify and understand inventory management issues. These insights led directly to actionable recommendations, such as adjusting reorder points, optimizing order quantities, and managing stock more proactively, ultimately enhancing operational efficiency and reducing inventory-related costs.

7 Discussion and Conclusions

This paper introduced a segmentation framework for object-centric process mining (OCPM) to tackle the complexity of long and intricate object lifecycles in event logs. By partitioning logs based on object relationships, event types, and temporal intervals, our approach enhances the clarity and interpretability of process data, addressing challenges that traditional methods struggle to overcome. The integration of this framework into the OC-PM tool, as demonstrated in our inventory management case study, revealed actionable insights—such as overstock and understock patterns and anomalous demand behaviors—that were obscured in unsegmented analyses. These findings directly informed operational improvements, including optimized reorder points and better purchase order timing, driving efficiency and cost reduction in inventory processes.

Despite these advancements, our approach has limitations. First, the effectiveness of segmentation relies on user expertise in selecting appropriate strategies (e.g., choosing relevant object types or event intervals), as outlined in Table 3. Inappropriate choices may lead to incomplete or misleading insights. Second, the framework assumes well-structured, high-quality event logs; noisy or incomplete data, common in real-world settings, can degrade segment quality and subsequent analyses. Third, computational costs may increase for very large logs, as segmenting and analyzing numerous subsets demands significant resources. These constraints highlight the need for careful application and potential preprocessing to ensure data quality.

Our segmentation framework lays a robust foundation for future work in OCPM and related fields. By breaking complex logs into focused, analyzable units, it enables researchers and practitioners to uncover localized patterns and correlations, as seen in our case study's improved anomaly detection and stock correlation insights. Future studies can exploit this approach by integrating automated segmentation strategy selection, leveraging AI techniques like large language models to recommend optimal partitions based on log characteristics [6]. Additionally, combining segmentation with predictive analytics could forecast process bottlenecks or resource needs, enhancing proactive decision-making.

References

1. Abb, L., Rehse, J.: Multivariate anomaly detection in object-centric event data. In: BPM Conf. pp. 20–36. Springer (2024)
2. Agostinelli, S., Leotta, F., Marrella, A.: Interactive segmentation of user interface logs. In: ICSOC Conf. pp. 65–80. Springer (2021)
3. Agostinelli, S., Marrella, A., Mecella, M.: Automated segmentation of user interface logs. In: Robotic Process Autom., pp. 201–222. De Gruyter Oldenbourg (2021)
4. Berti, A., Herforth, J., Qafari, M., van der Aalst, W.: Graph-based feature extraction on object-centric event logs. Int. J. Data Sci. Anal. **18**(2), 139–155 (2024)
5. Berti, A., Jessen, U., Park, G., Rafiei, M., van der Aalst, W.: Analyzing interconnected processes: using object-centric process mining to analyze procurement processes. Int. J. Data Sci. Anal. pp. 1–23 (2023)
6. Berti, A., Jessen, U., van der Aalst, W., Dirk Fahland: Explainable Object-Centric Anomaly Detection: the Role of Domain Knowledge. In: BPM-D Forum. pp. 162–168. CEUR-WS.org (2024)
7. Berti, A., Koren, I., Adams, J., Park, G., Knopp, B., Graves, N., Rafiei, M., Liß, L., Tacke Genannt Unterberg, L., Zhang, Y., et al.: Ocel (object-centric event log) 2.0 specification. arXiv:2403.01975 (2024)
8. Bosco, A., Augusto, A., Dumas, M., La Rosa, M., Fortino, G.: Discovering automatable routines from user interaction logs. In: BPM Forum. pp. 144–162. Springer (2019)
9. Goossens, A., De Smedt, J., Vanthienen, J.: Extracting process-aware decision models from object-centric process data. Inf. Sci. **682**, 121263 (2024)
10. Khayatbashi, S., Miri, N., Jalali, A.: Advancing object-centric process mining with multi-dimensional data operations. arXiv:2412.00393 (2024)
11. Liss, L., Adams, J., van der Aalst, W.: Totem: Temporal object type model for object-centric process mining. In: BPM Conf. pp. 107–123. Springer (2024)
12. Liu, F., Ting, K., Zhou, Z.: Isolation forest. In: ICDM Conf. pp. 413–422. IEEE (2008)
13. Park, G., Adams, J., van der Aalst, W.: Conformance checking and performance analysis using object-centric directly-follows graphs. In: BPM Conf. pp. 179–196. Springer (2024)
14. van der Aalst, W.: Object-centric process mining: unraveling the fabric of real processes. Math. **11**(12), 2691 (2023)
15. van der Aalst, W., Alessandro Berti: Discovering object-centric petri nets. Fundam. Inform. **175**(1-4), 1–40 (2020)
16. van Detten, J., Schumacher, P., Leemans, S.J.: Object Synchronizations and Specializations with Silent Objects in Object-Centric Petri Nets. In: BPM Conf. pp. 57–74. Springer (2024)