Explainable Object-Centric Anomaly Detection: the Role of Domain Knowledge

Alessandro Berti[1,2,*], Urszula Jessen[3,4], Wil M.P. van der Aalst[1,2] and Dirk Fahland[4]

[1]Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
[2]Fraunhofer FIT, Sankt Augustin, Germany
[3]Process Insights, ECE Group Services, Hamburg, Germany
[4]Eindhoven University of Technology, The Netherlands

Abstract

Anomaly detection is used in process mining to identify behavior differing significantly from the other instances. However, providing actionable insights out of the raw scores is challenging. In this paper, we propose three methodologies for explainable anomaly detection. In particular, we focus on object-centric event data as it increases the dimensions for anomaly detection, including the lifecycle of different objects and the interactions between them. Two of the proposed methodologies rely on the provision of domain knowledge, which can also be provided by Large Language Models (LLMs). We test the proposed techniques in a real-life case study on an (object-centric) ERP process.

Keywords

Object-Centric Anomaly Detection, Object-Centric Feature Extraction, Procurement Processes, Large Language Models

1. Introduction

Object-centric process mining [1] is a novel discipline that exploits *object-centric event data*, i.e., event data having each event correlated with several objects of different object types. Object-centric event data contains information related to the *lifecycle* of the different object types and the *interactions* between them. Several types of *object-centric process models* have been proposed, which can be discovered from object-centric event data using *object-centric process discovery* [2] algorithms. *Object-centric conformance checking* aims to compare the behavior contained in the object-centric event data against object-centric process models representing the normative behavior (*de-jure models*) to identify deviations. However, defining de-jure models in the object-centric setting is complicated due to the potentially large number of object types and their possible interactions.

Object-centric anomaly detection aims to identify anomalous behavior in the object-centric event data without requiring the definition of object-centric process models. They work by encoding object-centric event data into numerical *situation tables* to which anomaly detection algorithms are applied. For instance, if each row of the situation table represents a different object, anomaly detection algorithms assign an anomaly score to each object, which can be used to rank the objects based on their anomalousness.

A limitation of traditional approaches is the lack of interpretability of such scores, i.e., we are able to identify anomalous objects having a relevant anomaly score, but we are not able to provide any insights on why such objects were classified as anomalous. In this paper, we discuss three methodologies (**AF1**, **AF2**, and **AF3**) to provide actionable insights starting from either the situation tables or the anomaly

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 22nd International Conference on Business Process Management (BPM 2024), Krakow, Poland, September 1st to 6th, 2024.
*Corresponding author.
a.berti@pads.rwth-aachen.de (A. Berti); u.a.jessen@tue.nl (U. Jessen); wvdaalst@pads.rwth-aachen.de (W. M.P. v. d. Aalst); d.fahland@tue.nl (D. Fahland)
0000-0002-3279-4795 (A. Berti); 0000-0002-7282-8451 (U. Jessen); 0000-0002-0955-6940 (W. M.P. v. d. Aalst); 0000-0002-1993-9363 (D. Fahland)

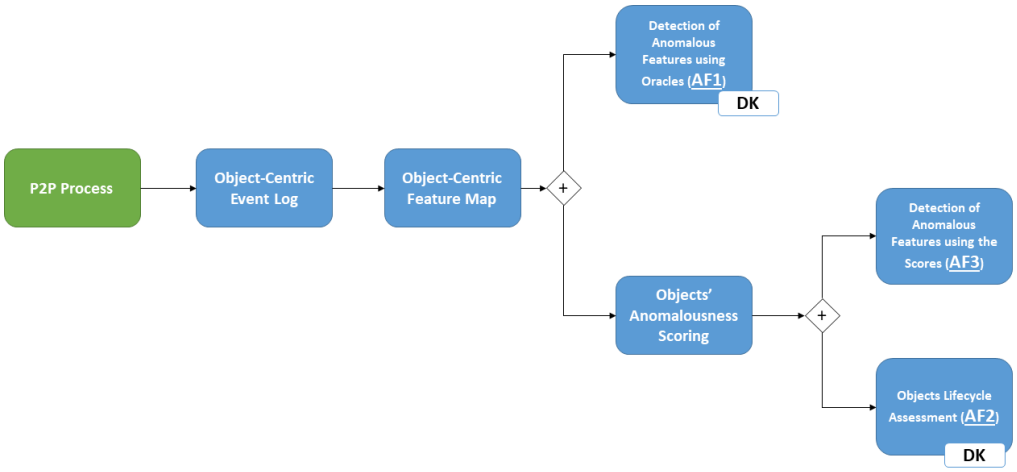

Figure 1: Outline of the contributions proposed in the paper. The approaches highlighted with "DK" require domain knowledge.

scores and apply them to a real-life (object-centric) P2P process. Fig. 1 summarizes the contributions of the paper.

The rest of the paper is organized as follows. In Section 2, we present the related work. In Section 3, we present three methodologies for explainable anomaly detection in the object-centric setting. In Section 4, we present the results of a case study using the techniques proposed in the paper. Finally, Section 5 concludes the paper.

2. Related Work

Object-centric conformance checking approaches divide between model-based [3, 4] and rule-based [5]. However, both categories suffer from the curse of dimensionality, as different object types and their interactions need to be modeled.

In the context of process mining, several anomaly detection approaches exist. Focusing on the object-centric setting, the usage of graph neural networks for anomaly detection is proposed in [6]. However, the approach focuses on detecting anomalous events, while in this paper we focus on objects and their relationships.

In [7], LLMs are proposed for semantic anomaly detection tasks. However, the approach focuses on traditional process mining instances, while we focus on object-centric process mining.

3. Approach

In this section, we propose three methodologies for anomaly detection in the object-centric setting. We assume that the object-centric event data has been encoded to a numerical situation table containing a row for each distinct object [8]. The numerical features (columns) are either related to the lifecycle of the objects (for example, the duration of the lifecycle, the number of events, or the one-hot encoding of the events' activities) or the interactions between them (for instance, counting how many distinct order items are related to a given order document). We apply traditional anomaly detection algorithms to the situation table. Therefore, an anomaly score is assigned to each object. In Table 1, we see some anomalous objects and their anomaly scores based on the Isolation Forests[1] and Local Outlier Factor[2] techniques.

The three methodologies follow:

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

Listing 1: Anomalies spotted by an LLM starting from the feature table (**AF1**)

1. **High Duration of Lifecycle**: The range of lifecycle durations is broad, with the maximum value at 722,032,120 while the 75% quantile is 1,215,387, indicating potential abnormally long lifecycles.
2. **Activity Occurrence Consistencies**: "Submit Purchase Order for Approval" and " Approve Purchase Order" activities typically occur once per object but have outliers with up to 100 occurrences. Investigate objects where these activities are repeated excessively.
3. **Degree of Centrality**: There's a significant jump in degree centrality values from the 75% quantile to the maximum. High centrality objects, potential interaction hubs , warrant further investigation.
4. **Rare Activities**: Activities like "Reject Purchase Order" and "(Re)Open Purchase Order" are infrequent. Focusing on these can reveal anomalies.
5. **Object Interactions**: Objects with an unusually high number of interactions in the graph may indicate anomalies.
6. **Lifecycle Start and Completion Time**: Analyze objects with exceptionally short or long lifecycles compared to the dataset trend.
7. **Objects Starting/Ending Lifecycle Together**: Examine cases where a notably high number of objects start or end their lifecycle simultaneously with the current object.

Listing 2: Anomalies spotted by an LLM on the lifecycle of a single object (**AF2**).

1. **Duplicate Time Stamps**: Both the "Approve Requisition" and "Create Purchase Order" events are recorded at the exact timestamp of 2010–10–06 03:44:22, suggesting an immediate creation of the PO after requisition approval, which is atypical due to the expected minor delay between these actions. Similarly, " Submit Purchase Order for Approval" and "Approve Purchase Order" share the timestamp of 2010–10–06 03:45:41, indicating unusual instant approval.
2. **Order of Events**: "Approve Purchase Order" is followed by another "Submit Purchase Order for Approval" event at the same timestamp, suggesting redundancy and possibly a recording error or system glitch.
3. **Long Lifecycle Duration**: The lifecycle of PO_277871 extends unusually from 2010–10–06 to 2023–07–12, closed initially on 2010–10–08 and then reopened 13 years later, which deviates from standard P2P process durations.
4. **Close and Reopen of PO**: PO_277871 was closed on 2010–10–08 and reopened on 2023–07–12, a rare occurrence that may require verification with system administrators to understand if it reflects actual procedural needs or system setup anomalies.

AF1 *Detection of Anomalous Features using Oracles*: we assume that an *oracle* examines the set of values for each column of the situation table and assigns a *strangeness score* to each of its values. For example, considering two orders, the first having a lifecycle duration of a week and one related invoice, and the second having a lifecycle duration of a year and 100 related invoices, the oracle can assign the following values:

- *number of related invoices=100*: strangeness score *0.9/1.0*
- *lifecycle duration=1 year*: strangeness score *0.7/1.0*
- *lifecycle duration=1 week*: strangeness score *0.3/1.0*
- *number of related invoices=1*: strangeness score *0.1/10.0*

The strangeness scores help to tailor the subsequent analysis. For example, we could search for all the orders in the object-centric event data having at least twenty related invoices, or all the orders having a lifecycle duration of more than six months. This proposed methodology requires domain knowledge of the underlying process. This knowledge can be provided by a human analyst or, alternatively, a Large Language Model (LLM) can be used for the purpose. For instance, Listing 1 represents the output of the GPT-4 LLM on this task. The different values of the situation table are ranked by the LLM, and a textual summary is provided containing the values having the highest strangeness. A limitation of this technique is that the inter-correlations between the values of different columns are ignored, as the focus is on the values of a single column.

AF2 *Objects Lifecycle Assessment*: we exploit the anomaly scores obtained with the application of an anomaly detection algorithm to rank the objects and identify the most anomalous ones. Then, for each of the most anomalous objects, the set of events related to the object is explored to spot semantic anomalies or root causes of performance issues. This methodology also requires domain knowledge of the underlying process, which can be provided by a human analyst or by an LLM. For instance, Listing 2 contains the anomalies identified by GPT-4 on the lifecycle of a single anomalous object.

AF3 *Detection of Anomalous Features using the Scores*: we exploit the anomaly scores obtained by applying an anomaly detection algorithm to measure the positive/negative correlation of the values of an object's feature with the anomaly score. The features having a lower/higher correlation with the anomaly scores are reported. In Table 2, for instance, we see some features with a negative correlation against the anomaly score. This is the the only approach among the three that does not require the provision of domain knowledge, but potentially it results in a lengthy list of anomalous features that may challenge analysts.

Object ID	Isolation Forest Scores	Local Outlier Factor Scores
PO_23667	-0.200785	-40.049412
PO_23507	-0.200311	-7.200163
PO_23508	-0.200311	-7.200163
PO_23512	-0.200311	-7.200163
PO_23513	-0.200311	-7.200163
PO_23514	-0.200311	-7.200163
PO_23515	-0.200311	-7.200163
PO_23516	-0.200311	-7.200163
PO_23517	-0.200311	-7.200163
PO_277871	-0.195874	-7.622763
PO_23511	-0.189318	-7.200163
PO_3903	-0.187092	-54.929239
PO_133097	-0.175762	-8.086049
PO_23668	-0.174838	-39.503084
PO_23669	-0.174838	-39.503084
PO_23510	-0.174382	-7.217331
PO_86355	-0.172010	-3.117746
PO_85465	-0.171363	-0.125512
PO_23518	-0.170136	-7.212333
PO_23519	-0.170136	-7.212333
PO_23520	-0.170136	-7.212333
PO_23521	-0.170136	-7.212333
PO_23522	-0.170136	-7.212333
PO_84184	-0.169095	-1.549233
PO_3836	-0.168964	-213.993317
PO_3837	-0.168964	-213.982787
PO_3838	-0.168964	-213.974041
PO_3839	-0.168964	-213.967588
PO_3840	-0.168964	-213.960370
PO_3841	-0.168964	-213.953323

Table 1: Anomaly scores for some purchase orders of the considered log.

Feature (with Value)	**Count**	Correlation
1 Occurrence of the activity Cancel Purchase Order	300	-0.07
1 Occurrence of the activity (Re)Open Purchase Order	167	-0.12
44 other orders are terminating with the same event	45	-0.21
45 other objects are interacting with the order	45	-0.21
The activity Approve Purchase Order is not executed	131	-0.07
There are 2 activities in the lifecycle of the order	72	-0.09
29 other orders are terminating with the same event	30	-0.19
30 other objects are interacting with the order	30	-0.19
27 other orders are terminating with the same event	28	-0.19
28 other objects are interacting with the order	28	-0.19
20 other orders are terminating with the same event	21	-0.18
There is a single event in the lifecycle of the order	53	-0.04
The activity Submit Purchase Order for Approval is not executed	53	-0.04
There are 13 events in the lifecycle of the order	41	-0.05

Table 2: Features' values correlated with anomalies (methodology **AF3**).

4. Case Study

In this section, we discuss the application of the proposed techniques on top of a real-life P2P object-centric event log (ECE group).

Context: The ECE group uses SAP ERP supported by the xFlow document acquisition system. The Celonis platform was adopted in 2020 for traditional process mining. However, due to the deficiency/-convergence/divergence issues, ECE quickly adopted object-centric process mining. The results of a case study have been previously published in [9]. The company was interested in applying anomaly detection to discover deviations from the expected behavior (non-compliance, such as maverick buying, i.e., inserting the order only after its placement, and post-mortem changes to purchase requisitions) and identify behavior leading to a monetary loss in the P2P process (for example, invoice paid twice, discount rates not taken because of invoices taking long to process, or non-justified payment blocks).

Tools: Our analysis primarily utilized the pm4py process mining library [10] and the OC-PM Javascript-based tool [2], which both support object-centric feature extraction as outlined in [8]. In previous work, we used these tools in a case study [9]. pm4py provides a dataframe via `pm4py.extract_ocel_features`, compatible with any Python machine learning library. OC-PM, after feature extraction, employs the Isolation Forests anomaly detection algorithm.

Adopted Dimensionality Reduction Algorithms: Due to the large number of features, we adopted dimensionality reduction to mitigate the curse of dimensionality, reduce computational complexity, and improve model performance.

In our P2P object-centric setting, FastMap [11] was the preferred method due to its ability to maintain non-linear relationships and computational efficiency. Unlike PCA[3], which involves intensive eigen-decomposition and can be less suitable for large datasets with ambiguous component interpretations, FastMap efficiently reduces high-dimensional data into lower dimensions without requiring full distance matrix computations.

Adopted Anomaly Detection Algorithms: The findings highlight the strengths of Isolation Forests and LOF in anomaly detection. Isolation Forests are effective for high-dimensional data and large volumes, isolating anomalies using decision tree splittings without needing pairwise distance calculations. This accelerates anomaly detection in complex datasets.

LOF excels at identifying anomalies in specific subgroups by calculating local density deviations,

[3] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

which is useful for clustered data. However, LOF requires more computational resources for large datasets.

In our analysis, Isolation Forests successfully detect anomalies in object-centric event logs with traditional lifecycle features, while LOF is preferable for graph-based features, focusing on local context to identify anomalies in networks of object interactions.

Refinement (Activities): After performing an initial analysis, we performed some postprocessing of the object-centric event log to enhance the results. We had hundreds of activities in the object-centric event log, mostly related to changing field values (change tables in SAP). Most of them are not relevant for object-centric anomaly detection and increase the dimensionality of the data with little gain. After our first application of anomaly detection, we repeated it on an object-centric event log that was filtered keeping only the relevant activities. The selection of relevant activities proved challenging on its own. Some infrequent activities, which were the first candidates for removal, identify indeed important anomalies. We could distinguish between manual and automatic activities, with the latter being less important for anomaly detection.

Refinement (Feature Propagation): We discovered that a traditional object-centric feature map based on the lifecycle and interactions of object types gives an incomplete process view. For instance, we found that invoices were often blocked for orders lacking preliminary purchase requisition approval, a pattern not visible when considering only invoices. By extending invoice data with information from related purchase orders using the feature propagation described in [8], we identified the root cause of this performance issue. Another observation, illustrated in Figure 2, is that orders with multiple positions (e.g., maintenance contracts) might appear anomalous when viewed in isolation. However, considering each item's direct relation to an invoice, such behavior is not anomalous.

Main Results: Anomaly detection allowed us to identify several non-compliance issues in the P2P process. We identified a non-negligible amount of orders with the maverick buying problem. The order is placed to the supplier skipping all the approval steps, the supplier sends an invoice to the company, and only then the purchase order is formally created in the ERP system. Moreover, we recorded several change activities done to purchase requisitions after their approval in order to match the amounts/quantities of the purchase order (post-mortem changes to PRs). This is a deleterious behavior as the purchase requisition was deliberately proposed to the managers with a lower amount.

Looking at the inefficiencies in the process leading to a monetary loss, we observed orders invoiced (and paid) several times, which were not maintenance contracts. Moreover, we identified invoices with an excessive number of change activities, signaling an inefficiency in the process (as this behavior is correlated with longer processing times). Considering the interaction between purchase orders, invoices, and payments, we observed that inefficiencies in the purchase orders also lead to inefficient processing of payments.

Limitations of LLMs: We used LLMs to interpret results, following methods in [12]. Specifically, `pm4py.llm.abstract_ocel_features` was used for textual abstraction in method **AF1**, and `pm4py.llm.abstract_ocel` for **AF3**. The GPT-4-Turbo LLM model, available as of 09-04-2024, was chosen to generate insights due to its large context window.

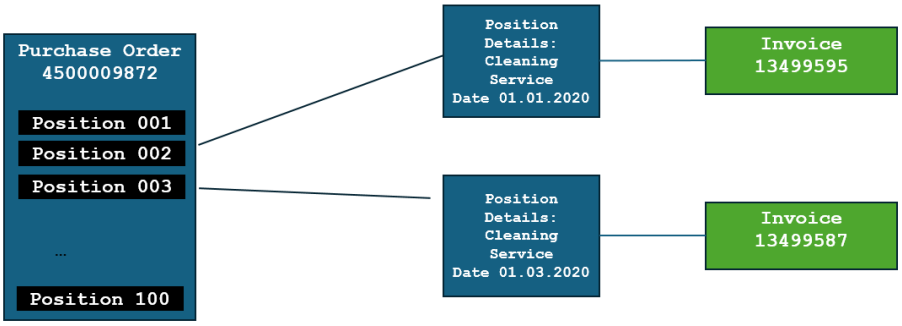

Figure 2: Interaction between maintenance contracts with several positions and invoices.

Applying LLMs to textual abstractions from our object-centric event log produced mixed results. For methodology **AF1**, the insights helped identify anomalous patterns and filter objects for further analysis using the OC-PM tool. However, several limitations arose. The context window of the LLM, despite improvements with the GPT-4-Turbo model, restricted the inclusion of objects' lifecycles containing many events, limiting the application of methodology **AF2** to objects with fewer events. Inconsistencies across different sessions were noted [13], sometimes requiring the merging of insights from different sessions as an "ensemble". Hallucinations and irrelevant outputs compared to the original prompt also occurred [13].

5. Conclusion

In this paper, we tackle anomaly detection in the object-centric setting. By transforming the object-centric event log into a tabular structure following the method described in [8], we are able to encode numerical features related to the lifecycle and the interaction of the different objects contained in the object-centric event log. Therefore, we are able to apply traditional anomaly detection methods and assign an anomaly score to each object.

A bigger challenge comes with explaining anomalies. The main contribution of this paper is to provide three methodologies for anomaly detection in the object-centric setting. In particular, two of them are based on having domain knowledge of the underlying process, while the third one is based on "transferring" the anomaly scores from the objects to the features level. For the methods requiring domain knowledge, we propose the usage of LLMs as domain knowledge providers given the large amount of process knowledge in their training datasets.

Applying the techniques in a real-life P2P setting, we found that the choice of the methodology is important, but also other design choices, such as the choice of the dimensionality reduction algorithm, the anomaly detection algorithm, and pre-processing the object-centric event log, are important.

In our case study, the application of object-centric anomaly detection allowed us to detect several anomalies in the underlying process (maverick buying, post-mortem changes to purchase requisitions, invoices with an excessive number of changes). Some problems, such as hallucinations and non-determinism, emerged in using LLMs as domain knowledge providers. However, for some of the use cases, LLMs provided excellent support.

References

[1] W. M. P. van der Aalst, Object-centric process mining: Dealing with divergence and convergence in event data, in: SEFM 2019, volume 11724, Springer, 2019, pp. 3–25.

[2] A. Berti, W. M. P. van der Aalst, OC-PM: analyzing object-centric event logs and process models, Int. J. Softw. Tools Technol. Transf. 25 (2023) 1–17.

[3] L. Liss, J. N. Adams, W. M. P. van der Aalst, Object-centric alignments, in: ER 2023, volume 14320, Springer, 2023, pp. 201–219.

[4] J. N. Adams, W. M. P. van der Aalst, Precision and fitness in object-centric process mining, in: ICPM 2021, IEEE, 2021, pp. 128–135.

[5] G. Park, W. M. P. van der Aalst, Monitoring constraints in business processes using object-centric constraint graphs, in: ICPM 2022 Workshops, volume 468, Springer, 2022, pp. 479–492.

[6] A. Niro, M. Werner, Detecting anomalous events in object-centric business processes via graph neural networks, CoRR abs/2403.00775 (2024).

[7] W. Guan, J. Cao, J. Gao, H. Zhao, S. Qian, Dabl: Detecting semantic anomalies in business processes using large language models, arXiv preprint arXiv:2406.15781 (2024).

[8] A. Berti, J. Herforth, M. S. Qafari, W. M. P. van der Aalst, Graph-based feature extraction on object-centric event logs, International Journal of Data Science and Analytics (2023).

[9] A. Berti, U. Jessen, G. Park, M. Rafiei, W. M. P. van der Aalst, Analyzing interconnected processes:

using object-centric process mining to analyze procurement processes, International Journal of Data Science and Analytics (2023).
[10] A. Berti, S. J. van Zelst, D. Schuster, Pm4py: A process mining library for python, Softw. Impacts 17 (2023) 100556.
[11] G. Ostrouchov, N. F. Samatova, On fastmap and the convex hull of multivariate data: Toward fast and robust dimension reduction, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1340–1343.
[12] A. Berti, D. Schuster, W. M. P. van der Aalst, Abstractions, scenarios, and prompt definitions for process mining with llms: A case study, in: BPM 2023 Workshops, volume 492, Springer, 2023, pp. 427–439.
[13] A. Berti, H. Kourani, H. Hafke, C. Yun-Li, D. Schuster, Evaluating Large Language Models in Process Mining: Capabilities, Benchmarks, Evaluation Strategies, and Future Challenges, in: Proceedings of the BPM-DS 2024 Working Conference, Springer, 2024.