Filtering and Sampling Object-Centric Event Logs

Alessandro Berti\textsuperscript{1,2}

\textsuperscript{1} Process and Data Science Group @ RWTH Aachen, Aachen, Germany \{a.berti, wvdaalst\}@pads.rwth-aachen.de
\textsuperscript{2} Fraunhofer Institute of Technology (FIT), Sankt Augustin, Germany

Abstract. The scalability of process mining techniques is one of the main challenges to tackling the massive amount of event data produced every day in enterprise information systems. To this purpose, filtering and sampling techniques are proposed to keep a subset of the behavior of the original log and make the application of process mining techniques feasible. While techniques for filtering/sampling traditional event logs have been already proposed, filtering/sampling object-centric event logs is more challenging as the number of factors (events, objects, object types) to consider is significantly higher. This paper provides some techniques to filter/sample object-centric event logs.

Keywords: Object-Centric Process Mining, Multi-perspective Process Mining, Scalable Process Mining

1 Introduction

Process mining is a branch of data science that exploits the event data contained in the information systems supporting mainstream processes (ERP, CRM) to infer process-related information. The primary assumption to apply process mining is the association of a case notion to the events. A case notion groups the events that belong to the same process execution. In a ticket management system, the identifier of the ticket is used to group all the events (creation, problem resolution, closure) related to the same ticket. However, in most of the other processes, the identification of a reasonable case notion is difficult. For example, in an Order-to-Cash process, a sales order might be associated with many deliveries. If the identifier of the sales order is chosen as a case notion, several instances of the activities done for different deliveries (create delivery, delivery successful) appear in the same case, and the divergence problem occurs. Conversely, if the identifier of the delivery is chosen as a case notion, we would need to record the same events related to the order (create order, pay order) in the same case, and the convergence problem occurs. Object-centric process mining is a recent branch of process mining that changes the assumption of having a unique case notion to the assumption that an event could be associated with different objects of different object types (e.g., order, delivery). Therefore, the convergence and
A. Berti et al.

Table 1: Tabular representation of an object-centric event log.

| Event ID | Event Activity | Event Timestamp | Orders | Items | Deliveries Weight Classes Goods Issues |
|----------|----------------|-----------------|--------|-------|----------------------------------------|
| e1       | Create Order   | 2007-04-01 07:29| [o1]   | [i1, i2] | Normal                                 |
| e2       | Pick Item      | 2007-04-01 15:36| [o1]   | [i1]   | Normal                                 |
| e3       | Pack Item      | 2007-04-01 22:01| [i1]   | [i1]   | Normal                                 |
| e4       | Pick Item      | 2007-04-02 22:15| [o1]   | [i1]   | Normal                                 |
| e5       | Pack Item      | 2007-04-01 23:19| [i1]   | [i1]   | Normal                                 |
| e6       | Deliver Successful | 2007-04-02 10:01| [i1]   | [d1]   | Normal                                 |
| e7       | Pay Order      | 2007-04-02 11:15| [o1]   | [i1]   | Normal                                 |
| e8       | Create Order   | 2007-04-02 12:15| [o2]   | [i4, i5, i6] | Normal                                 |
| e9       | Pick Item      | 2007-04-02 02:19| [o2]   | [i4]   | Normal                                 |
| e10      | Pack Item      | 2007-04-03 03:17| [i4]   | [i4]   | Normal                                 |
| e11      | Pick Item      | 2007-04-03 05:02| [o2]   | [i5]   | Normal                                 |
| e12      | Record Goods Issue | 2007-04-03 05:07| [o2]   | [i5]   | Normal                                 |
| e13      | Pick Item      | 2007-04-03 05:15| [o2]   | [i5]   | Normal                                 |
| e14      | Pack Item      | 2007-04-03 05:21| [i5]   | [d2]   | Normal                                 |
| e15      | Pick Item      | 2007-04-03 05:36| [o2]   | [i6]   | Normal                                 |
| e16      | Pack Item      | 2007-04-03 05:37| [i6]   | [d3]   | Normal                                 |
| e17      | Delivery Successful | 2007-04-03 16:01| [i6]   | [d2]   | Normal                                 |
| e18      | Delivery Successful | 2007-04-03 17:59| [o2]   | [d3]   | Normal                                 |
| e19      | Pay Order      | 2007-04-06 08:00| [o2]   | [i7]   | Normal                                 |
| e20      | Create Order   | 2007-04-08 19:38| [o3]   | [i7]   | Normal                                 |
| e21      | Pick Item      | 2007-04-08 23:54| [i7]   | [d4]   | Normal                                 |
| e22      | Pack Item      | 2007-04-08 23:58| [i7]   | [d4]   | Normal                                 |
| e23      | Delivery Successful | 2007-04-10 11:15| [i7]   | [d4]   | Normal                                 |
| e24      | Pay Order      | 2007-04-10 12:19| [o3]   | [i7]   | Normal                                 |

divergence problems can be resolved in the object-centric setting since we do not need to associate a delivery to the order events and an order to the delivery events. Table 1 contains a tabular representation of an object-centric event log. Each row corresponds to an event. Each column identifies either an attribute at the event level (identifier, activity, timestamp) or the list of objects for an object type (the orders, items, deliveries, weight classes, goods issues). What we can see in the table is the management of three orders. Considering o1, the order is associated to two items i1 and i2, which are both delivered using the delivery d1. Despite that, in no event o1 and d1 are directly associated, so at least for orders and deliveries the convergence/divergence problem is resolved. Object-centric event logs can be transformed (flattened) to traditional event logs with the choice of an object type. For example, choosing the delivery as object type, a flattened log would contain four cases (d1, d2, d3, d4). Each case would contain all the events related to the corresponding object (for example, Pack Item, Pack Item and Delivery Successful for d1). However, the extraction of an object-centric event log for mainstream information systems does not always solve the convergence/divergence problems, and some strange behavior could still be contained in the event log. We could see the following situations in the table: for the events with activity Pick Item, the order object related to the event is unnecessarily reported (convergence problem); all the events are associated with the same weight class Normal. Moreover, given the problems of scalability of most of the advanced object-centric process mining techniques (e.g., [8, 5]), keeping noisy behavior in the event log could significantly increase the execution time of the object-centric process mining technique. In Table 1, we have the object type Goods Issues which is associated to a single event e12. Moreover, there is only an event e12 with activity Record Goods Issue. Therefore, removing the object types Weight Classes and Goods Issues, which arguably are not the most
Fig. 1: Outline of the approach described in the paper. Object-centric process mining techniques may have scalability and quality issues starting from an object-centric event log containing lots of outlier behavior. Some filtering operations at different levels (object types, events) reduce the outlier behavior, such that the insights are provided with higher performance and quality. Eventually, a sampling operation can split the object-centric event log into smaller units and resolve the scalability issues.

significant ones for the given event log, could lead to significantly improving the execution times of some object-centric process mining algorithms. As additional results, the quality of the log (in the number of convergence/divergence problems) is improved by such filtering techniques. Even with removing outlier behavior, object-centric process mining techniques could require a significant computational effort. To tackle this, along with filtering techniques, sampling techniques might be considered. However, sampling in an object-centric event log is more challenging than in the traditional setting. While keeping a subset of cases is enough in the traditional setting, in an object-centric setting defining a set of events to sample requires considering the interaction between the different objects. In this paper, we will tackle both the filtering and sampling problems in the object-centric setting, with the purpose to enable object-centric process mining insights on big object-centric event logs. Several dimensions will be considered (filtering object types, events, object types per activity). Eventually, some strategies for sampling object-centric event logs are introduced. The rest of the paper is organized as follows. Section 2 introduces the core concepts in object-centric process mining. Section 3 introduces the approach (filtering and sampling object-centric event logs). Section 4 presents a tool in which the techniques are implemented. Section 5 applies the techniques to an event log extracted from an educational ERP instance. Section 6 presents the related work on object-centric process mining and sampling. Finally, Section 7 concludes the paper.

2 Preliminaries

In the following, we will introduce the definition of object-centric event log, which is the core concept on top of which the filters will be applied. In Def. 1, some universes needed for the definition of OCEL are introduced.

Definition 1 (Universes (for OCEL)). Some universes are used in the formal definition of object-centric event logs: $U_e$ is the universe of event identifiers;
$U_c$ is the universe of case identifiers; $U_{act}$ is the universe of activities; $U_{timest}$ is the universe of timestamps; $U_{att}$ is the universe of attribute names; $U_{val}$ is the universe of attribute values; $U_{typ}$ is the universe of attribute types; $U_o$ is the universe of object identifiers; $U_{ot}$ is the universe of objects types.

Def 2 introduces the formal definition of object-centric event log.

**Definition 2 (Object-Centric Event Log).** An object-centric event log is a tuple $L = (E, AN, AV, AT, OT, O, π_{typ}, π_{act}, π_{time}, π_{vmap}, π_{omap}, π_{otyp}, π_{ovmap}, ≤)$ such that:

- $E \subseteq U_c$ is the set of event identifiers.
- $AN \subseteq U_{att}$ is the set of attributes names.
- $AV \subseteq U_{val}$ is the set of attribute values (with the requirement that $AN \cap AV = \emptyset$).
- $AT \subseteq U_{typ}$ is the set of attribute types.
- $OT \subseteq U_{ot}$ is the set of object types.
- $O \subseteq U_o$ is the set of object identifiers.
- $π_{typ} : AN \cup AV \rightarrow AT$ is the function associating an attribute name or value to its corresponding type.
- $π_{act} : E \rightarrow U_{act}$ is the function associating an event (identifier) to its activity.
- $π_{time} : E \rightarrow U_{timest}$ is the function associating an event (identifier) to a timestamp.
- $π_{vmap} : E \rightarrow (AN \not\rightarrow AV)$ such that $π_{typ}(n) = π_{typ}(π_{vmap}(e))(n)$ $\forall e \in E$ $\forall n \in \text{dom}(π_{vmap}(e))$ is the function associating an event (identifier) to its attribute value assignments.
- $π_{omap} : E \rightarrow \mathcal{P}(O)$ is the function associating an event (identifier) to a set of related object identifiers.
- $π_{otyp} \in O \rightarrow OT$ assigns precisely one object type to each object identifier.
- $π_{ovmap} : O \rightarrow (AN \not\rightarrow AV)$ such that $π_{typ}(n) = π_{typ}(π_{ovmap}(o))(n)$ $\forall n \in \text{dom}(π_{ovmap}(o))$ $\forall o \in O$ is the function associating an object to its attribute value assignments.
- $≤$ is a total order (i.e., it respects the antisymmetry, transitivity, and connexity properties).

Recently, the OCEL standard [http://www.ocel-standard.org/](http://www.ocel-standard.org/) has been introduced for the storage of object-centric event logs [6]. Two different implementations of the standard are provided (JSON-OCEL and XML-OCEL), which are based on popular file formats.

### 3 Approach

The approach section is organized as follows. First, the different filters on the outliers (object types, events, object types per activity) are introduced. Then, the sampling methodologies are introduced.
3.1 Filter Outlier Object Types

Starting from an object-centric event log, it’s possible to filter a subset of object types. This will reduce the number of possible dimensions of the object-centric event log on top of which the interactions between the different object types are considered and possibly resolve some divergence problems. Def. 3 introduces object type-based filtering.

**Definition 3 (Filtering a Subset of Object Types).** Let \( L = \langle E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq \rangle \) be an object-centric event log. Given a set of object types \( \bar{OT} \subseteq OT \), we define \( \bar{O} = \{ o \in O \mid \pi_{otyp}(o) \in \bar{OT} \} \) as the set of objects having a type included in \( \bar{OT} \). We define the filtered log \( L_{\bar{OT}} = \langle E, AN, AV, AT, \bar{OT}, \bar{O}, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi'_{omap}, \pi'_{otyp}\rangle \), as

\[
L_{\bar{OT}} = \{ o \in \pi_{omap}(e) \mid o \in \bar{O} \}.
\]

In Def. 4, some filters on the object types are introduced. The first two filters (OTF1 and OTF2) count the number of different objects of an object type, the number of different events related to objects of a given type, and keep only the ones that have frequency \( \geq n \). The third (OTF3) measures the divergence problems for an object type since it counts if the activities are often repeated in the lifecycle of the objects of a type.

**Definition 4 (Outlier Filters on Object Types).** Let \( L = \langle E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq \rangle \) be an object-centric event log. We define the following filters at the object type level:

**OTF1** (Minimum Number of Different Objects per Object Type) given \( n \in \mathbb{N} \), we define: \( \bar{OT} = \{ ot \in OT \mid |\{ o \in O \mid \pi_{otyp}(o) = ot \}| \geq n \} \)

**OTF2** (Minimum Number of Events Associated per Object Type) given \( n \in \mathbb{N} \), we define: \( \bar{OT} = \{ ot \in OT \mid |\{ e \in E \mid \exists o \in \pi_{omap}(e) \pi_{otyp}(o) = ot \}| \geq n \} \)

**OTF3** (Minimum Ratio of Unique Activities) given \( r \in [0, 1] \), we define:

\[
\bar{OT} = \{ ot \in OT \mid \frac{\sum_{o \in O, \pi_{otyp}(o) = ot} |\{ \pi_{act}(e) \mid e \in E \wedge o \in \pi_{omap}(e) \}|}{\sum_{o \in O, \pi_{otyp}(o) = ot} |\{ e \in E \mid o \in \pi_{omap}(e) \}|} \geq r \}
\]

Table 2 can be obtained applying filtering on object types. The object type *Goods Issues* can be filtered out by OTF1 and OTF2 (since there is only an event associated with objects of the given type, and a single object is associated with the given type). The object type *Weight Classes* can be filtered out either by OTF1 (since there is a single object in the given object type) and OTF3 (for the object *Normal*, there are only six unique activities out of 24, so any ratio above 0.25 will remove the object type).

3.2 Filter Outlier Events

Another type of filtering possible on object-centric event logs is event-based filtering. Essentially, a subset of events of the original object-centric event log is selected.
Definition 5 (Filtering a Subset of Events). Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. Given a set of events \( E \subseteq E \), we define the filtered log \( L_{E|E} = (\bar{E}, AN, AV, AT, OT, O, \pi_{typ}|\bar{E}, \pi_{act}|\bar{E}, \pi_{time}|\bar{E}, \pi_{omap}|\bar{E}, \pi_{otyp}|\bar{E}, \pi_{ovmap}|\bar{E}, \leq | \bar{E} \times \bar{E}) \).

In the following, some criteria will be introduced to grade the importance of events. This will help to identify the events to keep/remove in the filtering. First, in Def. 6 the concepts of lifecycle (the sequence of events related to an object) and interaction lifecycle (the sequence of events related to a couple of objects; so, the steps in which the interactions between the two objects occur) are introduced.

Definition 6 (Objects Lifecycle and Interaction Lifecycle). Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. Given an object \( o \in O \), we define its lifecycle as the ordered sequence of events \( \pi_{o}(o) = [e_1, \ldots, e_n] \) such that \( o \in \pi_{omap}(e_i) \) for \( i \in \{1, \ldots, n\} \) and \( e_i - e_{i-1} \leq e_i \) for \( i \in \{2, \ldots, n\} \). We define the interaction lifecycle between two objects \( o_1, o_2 \in O \) as the ordered sequence of events \( \pi_{ilift}(o_1, o_2) = [e_1, \ldots, e_m] \) where \( o_1 \in \pi_{omap}(e_i) \wedge o_2 \in \pi_{omap}(e_i) \) for \( i \in \{1, \ldots, m\} \) and \( e_i - e_{i-1} \leq e_i \) for \( i \in \{2, \ldots, m\} \).

Essential events are defined in Def. 7 as the ones which either start/complete the lifecycle of an object, start/complete an interaction between two objects, or show a synchronization of the lifecycle [5].

Definition 7 (Essential Events). Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. We define an event \( e \in E \) as essential if one of the following conditions is met:

Table 2: Filtered object-centric event log, using the filters at the object type level, starting from the log introduced in Table 1. The activities highlighted in bold are essential events (Def. 7). The objects colored in gray belong to harmful object types-activities associations (Def. 10).

| Event ID | Event Activity | Event Timestamp | Orders | Items | Deliveries |
|----------|----------------|-----------------|--------|-------|------------|
| 1        | Create Order   | 2007-04-01 01:26 | [\textit{[\textit{6}, \textit{14}]]) | [\textit{[\textit{14}, \textit{52}]}) |  |
| 2        | Pick Item      | 2007-04-01 15:36 | [\textit{[\textit{11}]}) | [\textit{[\textit{14}]}) |  |
| 3        | Pack Item      | 2007-04-01 22:01 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 4        | Pick Item      | 2007-04-01 22:15 | [\textit{[\textit{12}]}) | [\textit{[\textit{14}]}) |  |
| 5        | Pack Item      | 2007-04-01 23:19 | [\textit{[\textit{12}]}) | [\textit{[\textit{14}]}) |  |
| 6        | Deliver Successful | 2007-04-02 10:01 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 7        | Pay Order      | 2007-04-02 11:15 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 8        | Create Order   | 2007-04-02 12:15 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 9        | Pick Item      | 2007-04-03 02:19 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 10       | Pack Item      | 2007-04-03 03:17 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 11       | Pick Item      | 2007-04-03 05:02 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 12       | Record Goods Issue | 2007-04-03 05:07 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 13       | Pick Item      | 2007-04-03 05:15 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 14       | Pack Item      | 2007-04-03 05:21 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 15       | Pick Item      | 2007-04-03 05:36 | [\textit{[\textit{16}]}) | [\textit{[\textit{16}]}) |  |
| 16       | Pack Item      | 2007-04-03 05:37 | [\textit{[\textit{16}]}) | [\textit{[\textit{16}]}) |  |
| 17       | Deliver Successful | 2007-04-03 16:01 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 18       | Deliver Successful | 2007-04-03 17:59 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 19       | Pay Order      | 2007-04-06 08:00 | [\textit{[\textit{15}]}) | [\textit{[\textit{15}]}) |  |
| 20       | Create Order   | 2007-04-08 19:38 | [\textit{[\textit{17}]}) | [\textit{[\textit{17}]}) |  |
| 21       | Pick Item      | 2007-04-08 23:54 | [\textit{[\textit{17}]}) | [\textit{[\textit{17}]}) |  |
| 22       | Pack Item      | 2007-04-08 23:54 | [\textit{[\textit{17}]}) | [\textit{[\textit{17}]}) |  |
| 23       | Deliver Successful | 2007-04-10 11:15 | [\textit{[\textit{14}]}) | [\textit{[\textit{14}]}) |  |
| 24       | Pay Order      | 2007-04-10 12:19 | [\textit{[\textit{18}]}) | [\textit{[\textit{18}]}) |  |
OE1 (Filter Activities Occurrences) \( \exists o \in O \; e = lif(o)(1) \).

OE2 (Filter on Essential Events) \( \exists e \in E \; e = lif(o)(\|lif(o)\|) \).

EE3 (Starting the Interaction between two Objects) \( \exists o_1, o_2 \in O, o_1 \neq o_2 \; e = intimf(o_1, o_2)(1) \).

EE4 (Ending the Interaction between two Objects) \( \exists o_1, o_2 \in O, o_1 \neq o_2 \; e = intimf(o_1, o_2)(1) \).

EE5 (Synchronization between the Lifecycles)

\[ \exists o_1, o_2 \in O, o_1 \neq o_2, e_1 \leq e_2 \leq e_3 \in E, e_3 \in intimf(o_1, o_2) \land e_2 \notin intimf(o_1, o_2) \land e_2 \in lif(o_1) \cup lif(o_2) \]

We define \( essel(L) \) as the set of essential events of the object-centric event log.

Essential events, together with the frequency of activities, can be used to formulate reasonable filters at the event level. We can either consider each filter in isolation or consider the combination between filtering essential events and frequency of activities: an event is kept in the log if it is essential or its number of occurrences exceeds a threshold.

Definition 8 (Outlier Filters on Events). Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{omap}, \pi_{atyp}, \pi_{omap}, \leq) \) be an object-centric event log. We define the following filters at the event level:

OE1 (Filter Activities Occurrences) Let \( E_a = \{ e \in E \mid \pi_{act}(e) = a \} \). Given \( n \in \mathbb{N} \), we define \( \bar{E} = \{ e \in E \mid |\pi_{act}(e)| \geq n \} \).

OE2 (Filter on Essential Events) \( \bar{E} = essel(L) \).

OE3 (Filter Activities Occurrences or Essential Events) Let \( E_a = \{ e \in E \mid \pi_{act}(e) = a \} \). Given \( n \in \mathbb{N} \), we define: \( \bar{E} = \{ e \in E \mid |\pi_{act}(e)| \geq n \lor e \in essel(L) \} \).

Let’s consider the event log reported in Table 2. The events highlighted in bold are essential events according to one of the criteria defined in Def. 7.

3.3 Filter Object Types per Activity

Another possibility is to filter the related objects to an event based on an activity-object type association. An object type, such as Orders in Table 2, is important for the definition of events having some activities (such as Create Order and Pay Order) but might lead to divergence issues for events having some other activities (such as Pick Item). Therefore, a strategy is to leave the associations between the objects of type Order and the events having as activity Create Order or Pay Order, but remove the associations with the events having as activity Pick Item. Def. 9 introduces such filtering.

Definition 9 (Filtering Object Types-Activities Relations). Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{omap}, \pi_{atyp}, \pi_{omap}, \leq) \) be an object-centric event log. Let \( OA \subseteq OT \times \{ \pi_{act}(e) \mid e \in E \} \) be a set of allowed object types - activities relations. We consider the filtered event log \( L_{OA \subseteq OA} = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{omap}, \pi_{omap}, \pi_{atyp}, \pi_{omap}, \leq) \) where

\[ \pi'_{omap}(e) = \{ o \in \pi_{omap}(e) \mid (\pi_{atyp}(o), \pi_{act}(e)) \in OA \} \]
A possible strategy to detect the associations between object types and activities is the one described in Def. 10. Essentially, for every possible combination of object types and activities, the number of objects of the given object type associated with events having the given activity is counted. This is divided by the summation, for all the objects of the given object type, of the number of events having the given activity related to the given object. A number between 0 and 1 is obtained with this division. The number is lower when several events of the given activity are related to the same object. Therefore, the associations leading to a divergence problem could be pruned out using this number.

**Definition 10 (Outlier Filter on Object Types-Activities Relations).**

Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. Given \( r \in [0, 1] \), we define the following filter (ratio of unique objects per type per activity):

\[
\overline{OA} = \left\{ (ot, a) \in OT \times \{ \pi_{act}(e) \mid e \in E \} \mid \frac{\sum_{o \in O, \pi_{otyp}(o)=ot} \mathbb{1}_{\exists e \in E, \pi_{act}(e)=a \land o \in \pi_{omap}(e)} \sum_{o \in O, \pi_{otyp}(o)=ot} \{|\{e \in E \mid \pi_{act}(e) = a \land o \in \pi_{omap}(e)\}| \geq r \} \right\}
\]

In Table 2, the objects of type *Orders* related to events having activity *Pick Item* are highlighted because the association leads to divergence problem. This is identified using the definition Def. 10. An alternative is a manual selection of the association between activities and object types (see Fig. 3).

### 3.4 Sampling

In this section, we will describe some sampling strategies that could be used on object-centric event logs. The first three are random sampling on different dimensions (events, objects, object types). The fourth considers as samples the connected events of the log. Initially, Def. 11 is introduced to allow filtering on objects.

**Definition 11 (Filtering a Subset of Objects).**

Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. Given a set of objects \( \bar{O} \subseteq O \), we define the filtered log \( L_{\bar{O}|O} = (E, AN, AV, AT, OT, \bar{O}, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) where \( \pi'_{omap}(e) = \{ o \in \pi_{omap}(e) \mid o \in \bar{O} \} \).

Def. 12 introduces the sampling strategies formally.

**Definition 12 (Sampling Strategies).**

Let \( L = (E, AN, AV, AT, OT, O, \pi_{typ}, \pi_{act}, \pi_{time}, \pi_{vmap}, \pi_{omap}, \pi_{otyp}, \pi_{ovmap}, \leq) \) be an object-centric event log. We can define the following sampling strategies:

**SS1 (Random sampling on events)** A random subset of events \( \tilde{E} \subseteq E \) is chosen. The sampled log is \( L_{E|\tilde{E}} \) (Def. 5).
SS2 (Random sampling on objects) A random subset of events \( \bar{O} \subseteq O \) is chosen. The sampled log is \( L_{\bar{O}|O} \) (Def. 11).

SS3 (Random sampling on object types) A random subset of object types \( \bar{OT} \subseteq OT \) is chosen. The sampled log is \( L_{OT|\bar{OT}} \) (Def. 3).

SS4 (Sampling connected events) Given the relation \( R \subseteq E \times E \), \((e_1, e_2) \in R \iff \pi_{omap}(e_1) \cap \pi_{omap}(e_2) \neq \emptyset\), consider the transitive closure\(^3\) \( R^+ \) of \( R \). The equivalence classes of \( R^+ \) are the samples.

Random sampling on object types removes some object types from the log, but the insights on the remaining object types are entirely valid. Random sampling events/objects trim the lifecycle of objects and remove relationships between them, so the quality of insights obtained from object-centric process mining techniques is hampered. The fourth strategy divides the event log into “self-contained units” on top of which meaningful insights can be discovered.

4 Tool

![Fig. 2: Screenshot of the process schema component of the OCPM tool.](image)

This section presents OCPM, a web interface for object-centric process mining. The repository of the process is https://github.com/Javert899/pm4js-sandbox. The application can be run by either 1) cloning the Github repository and 2) launching the file `ocel.html`, or accessing the application deployed at [http://www.ocel-standard.org/ocpm/ocel.html](http://www.ocel-standard.org/ocpm/ocel.html). The application is made by several components (process schema, events explorer, objects explorer, network analysis, filtering, statistics, conformance). The filtering component could be accessed by clicking “Advanced Filtering” on the top. There, the filters proposed in this paper are proposed. In addition, the deployed application is pre-loaded with the event log used in the assessment. Fig. 2 shows the process schema component of the tool.

\(^3\) \( R^+ = \bigcup_{i=1}^{\infty} R^i \) where \( R^i \) is defined inductively as \( R^1 = R \) and for \( i > 0 \) \( R^{i+1} = R \circ R^i \).
5 Assessment

The assessment proposed in this section is based on the tool OCPM (presented in the previous section) and an Order-to-Cash log extracted from an educational SAP instance. The original event log is composed by 4077 events, 4084 objects and 8 different object types (Inquiry with 316 objects, Quotation with 361 objects, Order with 640 objects, Delivery with 614 objects, Goods Movement with 914 objects, Shipment with 96 objects, WMS transfer order with 576 objects, and Invoice with 567 objects).

Filter on Object Types: at first, we would like to filter out some of the less significant object types. If we apply the filter OTF2 (filter object types having a minimum number of related events) with \( n = 700 \), the object types of the event log are reduced to Order, Delivery and Goods Movement (the object types with the most frequent occurrence in the log). The event log is reduced to 2168 objects (53%).

Filter on Events: at this point, we would like to filter the object-centric event log keeping only the essential events. In an Order-to-Cash process, the change activities operate on an existing object, and we could remove all the events which do not start/terminate the lifecycle of an object or start/terminate/synchronize the interaction between two objects. Therefore, the event log is further reduced to 2086 events (65%).

Sampling: using the sampling strategy SS\(4\) in Def. 12, we get 657 samples, with the most significant counting 5 events. Therefore, the event log is split into many small sub-logs containing only parts of the behavior.

6 Related Work

This section presents the related work on object-centric process mining and filtering/sampling event logs.

Object-centric Process Mining: in [1], two of the main problems of traditional event logs (convergence/divergence) are presented as motivation to work with object-centric process mining. Different process models have been proposed, including [8] (object-centric behavioral constraints models; from an object-centric
event log, a declarative model is discovered but with limited scalability due to the computationally-expensive constraints), [3] (a model is discovered as collation of directly-follows graphs) and [2] (discovering object-centric Petri nets). In [5], unbounded synchronizations in processes with many-to-many interactions are described, which require to know all the synchronizations between the life-cycles of the object. The OCEL standard for the storage of object-centric event logs is proposed in [6].

Filtering and Sampling: many scientific contributions exist in this area. For filtering, we cite as references [4] and [9]. For sampling, we cite [7].

7 Conclusion

In this paper, a methodology to filter outliers and sample object-centric event log is presented. Several dimensions (events, objects, object types) are considered for such operations. Filtering is done to increase the quality of the object-centric event log (for example, reducing the divergence problems). This helps to reduce the complexity of object-centric process mining operations and the resulting insights. A sampling operation can be applied to the original/filtered object-centric event log. This is especially useful on real-life datasets since some object-centric process mining techniques have limited scalability. We also propose a web-based tool implementing the aforementioned filtering and sampling operations. The assessment on an O2C object-centric event log extracted from an educational instance shows that the filtering technique helps to reduce the infrequent behavior in the object-centric event log, and the sampling techniques return several samples out of the considered log.

Future Work: The set of techniques proposed in the current paper is based on simple metrics. More advanced metrics could be considered. For example, having a set of features associated to each object (such as metrics on the lifecycle, on the interaction between objects, and on the object attributes) enables the application of anomaly detection techniques (such as isolation forests).

Acknowledgements

We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. van der Aalst, W.M.P.: Object-centric process mining: Dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) Software Engineering and Formal Methods - 17th International Conference, SEFM 2019, Oslo, Norway, September 18-20, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11724, pp. 3–25. Springer (2019). https://doi.org/10.1007/978-3-030-30446-1_1
2. van der Aalst, W.M.P., Berti, A.: Discovering object-centric petri nets. Fundam. Informaticae 175(1-4), 1–40 (2020). https://doi.org/10.3233/FI-2020-1946
3. Berti, A., van der Aalst, W.M.P.: Extracting multiple viewpoint models from relational databases. In: Ceravolo, P., van Keulen, M., López, M.T.G. (eds.) Data-Driven Process Discovery and Analysis - 8th IFIP WG 2.6 International Symposium, SIMPDA 2018, Seville, Spain, December 13-14, 2018, and 9th International Symposium, SIMPDA 2019, Bled, Slovenia, September 8, 2019, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 379, pp. 24–51. Springer (2019). https://doi.org/10.1007/978-3-030-46633-6_2
4. Conforti, R., Rosa, M.L., ter Hofstede, A.H.M.: Filtering out infrequent behavior from business process event logs. IEEE Trans. Knowl. Data Eng. 29(2), 300–314 (2017). https://doi.org/10.1109/TKDE.2016.2614680
5. Fahland, D.: Describing behavior of processes with many-to-many interactions. In: Donatelli, S., Haar, S. (eds.) Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11522, pp. 3–24. Springer (2019). https://doi.org/10.1007/978-3-030-21571-2_1
6. Ghahfarokhi, A.F., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: A standard for object-centric event logs. In: Bellatreche, L., Dumas, M., Karras, P., Matulevicius, R., Awad, A., Weidlich, M., Ivanovic, M., Hartig, O. (eds.) New Trends in Database and Information Systems - ADBIS 2021 Short Papers, Doctoral Consortium and Workshops: DOING, SIMPDA, MADEISD, MegaData, CAoNS, Tartu, Estonia, August 24-26, 2021, Proceedings. Communications in Computer and Information Science, vol. 1450, pp. 169–175. Springer (2021). https://doi.org/10.1007/978-3-030-85082-1_16
7. Kabierski, M., Nguyen, H.L., Grunske, L., Weidlich, M.: Sampling what matters: Relevance-guided sampling of event logs. In: Ciccio, C.D., Francescomarino, C.D., Soffer, P. (eds.) 3rd International Conference on Process Mining, ICPM 2021, Eindhoven, Netherlands, October 31 - Nov. 4, 2021. pp. 64–71. IEEE (2021). https://doi.org/10.1109/ICPM53251.2021.9576875
8. Li, G., de Carvalho, R.M., van der Aalst, W.M.P.: Automatic discovery of object-centric behavioral constraint models. In: Abramowicz, W. (ed.) Business Information Systems - 20th International Conference, BIS 2017, Poznan, Poland, June 28-30, 2017, Proceedings. Lecture Notes in Business Information Processing, vol. 288, pp. 43–58. Springer (2017). https://doi.org/10.1007/978-3-319-59336-4_4
9. Sani, M.F., van Zelst, S.J., van der Aalst, W.M.P.: Improving process discovery results by filtering outliers using conditional behavioural probabilities. In: Teniente, E., Weidlich, M. (eds.) Business Process Management Workshops - BPM 2017 International Workshops, Barcelona, Spain, September 10-11, 2017, Revised Papers. Lecture Notes in Business Information Processing, vol. 308, pp. 216–229. Springer (2017). https://doi.org/10.1007/978-3-319-74030-0_16