From Identities to Quantities: Introducing Items and Decoupling Points to Object-centric Process Mining*

Nina Graves, István Koren, Majid Rafiei, and Wil M.P. van der Aalst

Chair of Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany {graves,koren,majid.rafiei,wvdaalst}@pads.rwth-aachen.de
https://www.pads.rwth-aachen.de/

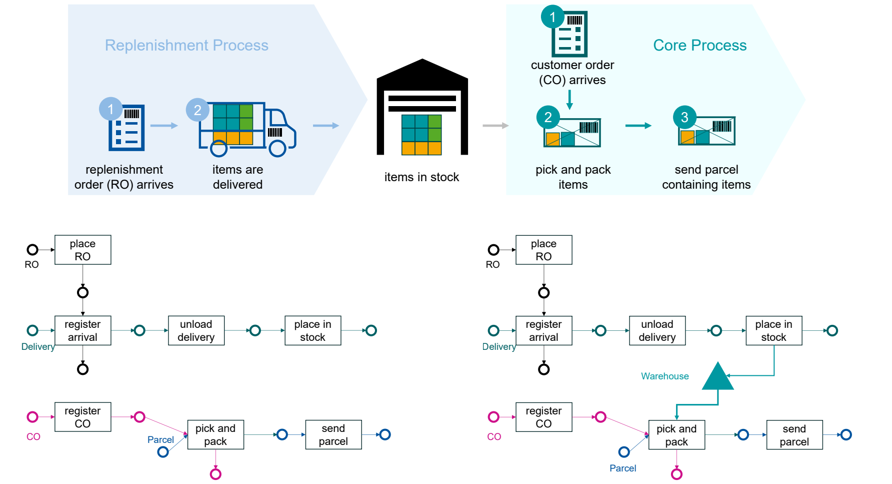
Abstract. Logistics processes ensure that the right product is at the right location at the right time in the right quantity. Their efficiency is crucial to industrial operations, as they generate costs while not adding value to the product. Process mining techniques improve processes using real-life data. However, the application of process mining to logistics processes poses several challenges, as (1) recorded material movements refer to quantities of items, not individual objects and (2) the required data are often scattered over several systems requiring additional pre-processing efforts. This work presents the concept of item quantities to describe the movement of not individually identifiable items across distributed processes. Subsequently, we introduce a framework to integrate the explicit consideration of item quantities into process mining, consisting of a quantity-related event log and an extension of object-centric Petri nets as a basis for quantity-dependent process analysis. The analysis of an artificial event log demonstrates the additional insights the consideration of quantities uncovers and highlights the potential for the application of process mining in the logistics domain.

Keywords: Process Mining · Logistics · Material Flow Analysis.

1 Introduction

In current times, a swift transformation towards sustainable practices is essential, requiring in-depth analysis and transformation of processes concerning the sourcing, processing, and transporting of material [12]. Process mining is a relatively young discipline, leveraging readily available event data to analyse, monitor, improve and support business processes [2]. Apart from being a growing field for research in academia [9], its industrial adoption is swiftly expanding across various industries [8]. Despite the benefits of process mining for intra-logistics [7] as well as inter-organisational processes [13], its application to the logistics domain is relatively low [9].

The main reason for this discrepancy lies in the data availability [15]. The event log used for object-centric process mining, contains information on the



(a) Disconnected OCPN without item IDs.

(b) OCPN extended by a decoupling point indicating a collection of items.

Fig. 1: Example of a quantity-dependent process composed of two decoupled sub-processes.

execution of an activity linked to at least one individually identifiable object [3]. Process discovery algorithms combine the events referring to the same object to detect dependencies among different activities – this means every object requires one unique identifier for the end-to-end process. There are two requirements data on logistics processes do not necessarily fulfil: First, logistics processes describe material movements, such as the addition or removal of a number of items without necessarily referring to uniquely identifiable objects [5,9]. For example, one book and two cups are added to the warehouse, instead of book-123, cup-456 and cup-789. Secondly, they can be scattered across several systems, even crossing organisational boundaries, leading to the required data being distributed without matching identifiers [4], i.e., the identifiers different organisations use for the same object are different.

Consider, for example, a warehouse management process, as depicted in Figure 1. The warehouse's core process is fulfilling customer orders (shown on the right of the figure), referring to its three item types: photo albums a, books b, and cups c. Whenever a customer order arrives, the requested number of items is picked from the warehouse and packed into a parcel before it is sent to the customer. When the number of items in the warehouse is low, a replenishment sub-process is executed. This sub-process begins with placing a replenishment order, leading to the delivery of items that are unpacked and placed in the warehouse. All uniquely identifiable objects (replenishment orders, deliveries, customer orders and parcels) are associated with their corresponding events. One delivery contains a number of items sufficient for fulfilling several customer orders. We see that (1) considering the overall process, there is no 1:1 relationship between the execution of a replenishment process and a core process – they are *decoupled*, (2) the two sub-processes are connected by the items added to and removed from the warehouse, and (3) there is a dependency in the execution of both sub-processes on the available items in the warehouse. Using an event log only referring to the uniquely identifiable objects, process mining techniques can only discover two disconnected sub-processes, as can be seen for example in Figure 1a, and cannot support their analysis in consideration of non-identifiable items. In contrast, the net in Figure 1b, is able to represent both a collection of items as well as the decoupling of the two processes by using an additional type of node: a *decoupling point*.

This paper presents a framework extending object-centric process mining to enable the detection and analysis of decoupled, quantity-dependent processes. To do so, we introduce *quantity-relating event logs* to enable the consideration of item quantities as well as an extension of object-centric Petri nets. We further demonstrate the additional insights that can be gained by taking available item quantities at the execution time of events into account using an artificial event log. After presenting related work (Section 2), Section 3 introduces the preliminaries including items, item collections and quantity operations. Quantity-relating event logs (QELs) and quantity nets are presented in Section 4. In Section 5, we analyse an artificial QEL to present the benefits and shortcomings of the framework. Section 6 concludes this work.

2 Related Work

The problems of missing or mismatching identifiers and the discovery and analysis of quantity-dependent processes have been addressed in literature. Federated process mining deals with the joint consideration of disconnected processes, described and formalised in [1]. Approaches to match identifiers of distributed processes using EDI messages [6] and leveraging Radio Frequency Identification (RFID) data [10] were discussed in the literature. In [11], the authors introduce an approach in which individually identifiable objects are grouped to map their identifiers to joint identifiers collected from RFID data. All of these methods rely on the existence of the individual and shared identifiers for the relevant entities of the process and allow for a fully end-to-end analysis of each of the relevant entities. In contrast, the authors of [20] present an abstraction-based, privacy-preserving approach to discovering inter-organisational processes. Although the presented procedure allows process analysis without requiring shared end-to-end identifiers, it requires a 1:1 mapping between process executions of the different systems.

A typical example of a quantity-based interdependency of process executions is batch processing. When processing in a batch, an activity or sub-process is executed a predetermined number of times without interruption before the objects are passed on to the next step [19]. Several works aim to discover different types of batched activities, such as [18] or batched sub-processes [17]. All of these approaches assume event data containing an entry for each batched element. The authors of [21] assume a mixture of events referring to batched objects as well as individual ones, the batched ones including the information on the number of batched objects.

Some works explicitly address the logistics-related challenges of process mining. In [5], an approach for the preparation and visualisation of material movement data is presented, allowing for the identification of inefficiencies in the paths the material takes. A methodology combining association rules with process mining to uncover dependencies between processes and performance indicators for supply chains is introduced in [16]. In [14], the authors present an approach to enrich event data with additional information to detect waste in a value stream. We see that existing literature focuses on preparing event data to use out-of-the-box process mining techniques instead of integrating the additionally available data to enhance process mining's capabilities.

3 Preliminaries

This section introduces some general mathematical operations as well as concepts related to object-centric process mining. Function projections, denoted $f\restriction_W$, define the application of a function $f: X \nrightarrow Y$ to a different domain W, with: $dom(f\restriction_W) = dom(f) \cap W$ and $f\restriction_W(x) = f(x)$ for $x \in dom(f\restriction_W)$. A sequence of length n over a set A is denoted $\sigma = \langle a_1, a_2, ..., a_n \rangle \in A^*$, with $\sigma = \langle\rangle$ as the empty sequence and $\sigma_1 \cdot \sigma_2$ the concatenation of sequences. The j-th element of a sequence $\sigma = \langle a_1, ..., a_j, ..., a_n \rangle \in A^*$ is denoted $\sigma[j] = a_j$ and the *prefix* of $\sigma[j]$ is referred to as $\sigma_{:j} = \langle a_1, ..., a_{j-1} \rangle$.

To define an Object-Centric Event Log (OCEL), the basis for object-centric process mining, we introduce the universe of activities $\mathcal{U}_{act}$, the universe of events $\mathcal{U}_{ev}$, the universe of objects $\mathcal{U}_o$, the universe of object types $\mathcal{U}_{ot}$, and the universe of timestamps $\mathcal{U}_{time}$.

Definition 1 (Object-centric Event Log). *An object-centric event log is a tuple $OCEL = (E, O, act, otypes, time, E2O)$, where $E \subseteq \mathcal{U}_{ev}$ is a set of events, $O \subseteq \mathcal{U}_o$ is a set of objects, $act: E \rightarrow \mathcal{U}_{act}$ is a function assigning activities to events, $otype: O \rightarrow \mathcal{U}_{ot}$ maps each object identifier to an object type, $time: E \rightarrow \mathcal{U}_{time}$ assigns a timestamp to each event, and $E2O \subseteq (E \times O)$ describes the relation between events and objects.*

We denote $A(OCEL)$ for the set of activities, and $OT(OCEL)$ as the set of object types. As introduced in [3], Object-Centric Petri Nets (OCPNs) are defined in correspondence to an OCEL by specifying the set of object types $OT(OCEL)$ and the tokens being associated with objects of the log. In addition to a labelled Petri net $N = (P, T, F, l)$, defined in the usual way, OCPNs include a mapping assigning an object type to every place, and a set of variable arcs.

Definition 2 (Object-Centric Petri net). *An object-centric Petri net is a tuple $ON = (N, pt, F_{var})$ where $N = (P, T, F, l)$ is a labelled Petri net, $pt: P \rightarrow OT$ maps places onto object types, and $F_{var} \subseteq F$ is a subset of variable arcs.*

Normal arcs describe the removal or addition of a single token, whereas variable arcs indicate a variable number of tokens to be removed or added. The firing of a transition occurs in regard to a *binding* (t, b), which includes the transition to be fired as well as a binding function. The binding defines the sets of tokens to be consumed, $cons(t, b)$, and produced, $prod(t, b)$.
The example presented in the introduction considers two operationally decoupled sub-processes tied by their impact on a known collection of items - a warehouse. We consider *items* to be elements of the process relevant to the control flow that occur in varying quantities. Although each item refers to a particular item type, they differ from objects as they are not described by an identifier but by the quantity they occur in. The universe of item types is denoted $\mathcal{U}_{it}$, and $I \subseteq \mathcal{U}_{it}$ refers to a set of item types. In the logistics domain, we have to distinguish between the demand for items of different types (negative quantity) and the actual presence of such items (positive quantity). Thus, we introduce *item quantities*, which assign a signed integer to each item type.

Definition 3 (Item Quantity). *Let $I \subseteq \mathcal{U}_{it}$ be a finite set of item types. An item quantity $q : I \to \mathbb{Z}$ is a function that maps each item type $i \in I$ to a signed integer. The set of all possible item quantities over I is denoted $\mathcal{I}(I) = \{q : I \to \mathbb{Z}\}$. Item quantities can be changed by addition and subtraction, given $q_1, q_2 \in \mathcal{I}(I)$ be two item quantities over $I \subseteq \mathcal{U}_{it}$:*
$q_1 \oplus q_2 = q_3$ where $\forall i \in I : q_3(i) = q_1(i) + q_2(i)$, and
$q_1 \ominus q_2 = q_3$ where $\forall i \in I : q_3(i) = q_1(i) - q_2(i)$.

We use a notation for item quantities similar to multisets but emphasising the difference using different brackets. Some examples for item quantities over $I = \{x, y, z\}$ with $I \subseteq \mathcal{U}_{it}$: $q_1 = [\![\ ]\!]$, $q_2 = [\![x, x, x, y]\!]$, $q_3 = [\![x, y^{-1}, y^{-1}, z, z]\!]$, $q_4 = [\![x^3, y]\!]$, and $q_5 = [\![x, y^{-2}, z^2]\!]$. q_1 is an empty item quantity, q_2 and q_4 are two notations for the same item quantity, just as q_3 and q_5. $q_2(y) = 1$ refers to the quantity associated with item type y for q_2, just as $q_3(y) = -2$ for item quantity q_3, and $q_1(y) = 0$ for q_1. The set of all item types q assigns a non-zero quantity to is denoted, $set(q) = \{i \mid q(i) \neq 0\}$, e.g. $set(q_2) = \{x, y\}$ and $set(q_1) = \emptyset$. Examples for operations on item quantities: $q_2 \oplus q_3 = [\![x^4, y^{-1}, z^2]\!]$, $q_4 \ominus q_5 = [\![x^2, y^3, z^{-2}]\!]$, and $q_5 \ominus q_3 = q_1$.

The positive item quantity of q is denoted $q^+ = b\restriction_{\{i \in set(q) \mid q(i) > 0\}}$ and $q^- = q\restriction_{\{i \in set(q) \mid q(i) < 0\}}$ as the negative item quantity of q, e.g. $q_3^+ = [\![x, z^2]\!]$ and $q_5^- = [\![y^{-2}]\!]$. An item quantity $q \in \mathcal{I}(I)$ is considered *fully positive*, iff $q = q^+$ and *fully negative* iff $q = q^-$.

A location or entity dedicated to collecting items, such as a warehouse or a buffer, is referred to as a collection point. Every collection point refers to an item quantity, its *item level*, describing the availability or lack of items of specific item types. We consider a finite set of collection points $CP \subseteq \mathcal{U}_{cp}$ from the universe of collection points.

Definition 4 (Item Levels). *Given a set of item types $I \subseteq \mathcal{U}_{it}$ and a set of collection points $CP \subseteq \mathcal{U}_{cp}$, the mapping $m_q : CP \to \mathcal{I}(I)$ describes the item levels of the collection points in CP.*

In line with Definition 3, the item level of a particular item type $i \in I$ of collection point $cp \in CP$ is denoted $m_{q,I}(cp)(i)$. Consider, a warehouse $cp \in CP$ containing items of three different types: $I = \{cups, books, albums\}$. Currently, there is a stock of 21 cups and the item level of books lies at $m_{q,I}(cp)(books) = 17$, but a demand for 25 albums. Hence, the item level for the warehouse is $m_{q,I}(cp) = [\![cups^{21}, books^{17}, albums^{-25}]\!]$.

A collection point's item level is changed in a *quantity operation* by adding another item quantity $q \in \mathcal{I}(I)$ to the current item level, denoted $m_{q,I}(cp) \xrightarrow{q} m^{'}_{q,I}(cp)$, where $m^{'}_{q,I}(cp) = m_{q,I}(cp) \oplus q$. After the execution of a quantity operation, the item level of all item types with a negative item quantity $i \in set(q^-)$ is reduced and increased for all available item types $i \in set(q^+)$. The execution of a sequence of quantity operations on the initial item level $m^{init}_{q,I}(cp) \xrightarrow{q_1} m_{q,I}{'}(cp) \xrightarrow{q_2} ... \xrightarrow{q_n} m^{''}_{q,I}(cp)$, is denoted $m^{init}_{q,I}(cp) \xrightarrow{\sigma} m_{q,I}{''}(cp)$, with $\sigma = \langle q_1, q_2, ..., q_n \rangle$ describing the sequence of item quantities. Using item quantities, we can now represent collections of items as well as material movements.

4 Quantity-Dependent Process Mining

The underlying concepts of item quantities, collection points, item levels and quantity operations are nothing new: Inventory management is one of the largest areas in logistics. However, current process mining techniques cannot describe item quantities and business processes jointly [9]. This section proposes a framework extending object-centric process mining to (1) connect the execution of events with changes to item levels, (2) detect the item levels of known collection points at the time individual processes are executed, and (3) model decoupled quantity-dependent processes. The framework is based on the assumption that the execution of quantity operations is connected to events described exhaustively in the event log.

4.1 Quantity-relating Event Logs

Our goal is to define an event log that enables the identification of dependencies between the execution of activities, sub-processes, and item levels without needing identifiers for all items. This requires an event log that connects events to quantity operations and, thereby, describes the item levels' development. We achieve this by adding collection points and item types to the log and mapping quantity operations to events.

Definition 5 (Quantity Event Log). *A quantity-relating event log is a tuple* $QEL = (OCEL, I, CP, eqty)$, *where* $OCEL = (E, O, act, otypes, time, E2O)$ *is an object-centric event log,* $I \subseteq \mathcal{U}_{it}$ *is a set of item types,* $CP \subseteq \mathcal{U}_{CP}$ *is the set of known collection points, and* $eqty : E \rightarrow (CP \nrightarrow \mathcal{I}(I))$ *assigns quantity updates to events and collection points.*

As this event log extends usual OCELs, applying other process mining techniques is not limited by using QELs. Table 1 shows a QEL for the example process of

Table 1: Example of a QEL displayed in a single table

events ($E, act, time$)			$eqty$		objects ($O, otypes, E2O$)			
event	activity	timestamp	quantity	cp	RO	CO	delivery	parcel
ev-zz	register co	22.12.2020 12:22	$[\![a^{-3}]\!]$	cp1		co-883		
ev-vl	pick and pack	22.12.2020 12:24	$[\![a^{-3}, b^{-10}]\!]$	cp2		co-882		p-942
ev-tg	send parcel	22.12.2020 12:31						p-941
ev-rg	register co	22.12.2020 13:56	$[\![b^{-3}, c^{-1}]\!]$	cp1		co-884		
ev-pa	send parcel	22.12.2020 13:57						p-942
ev-yq	pick and pack	22.12.2020 14:01	$[\![a^{-3}]\!]$	cp2		co-883		p-943
ev-kj	send parcel	22.12.2020 14:11						p-943
ev-bn	register delivery	22.12.2020 15:19			ro-11		d-11	
ev-iq	pick and pack	22.12.2020 15:23	$[\![b^{-3}, c^{-1}]\!]$	cp2		co-884		p-944
ev-gq	send parcel	22.12.2020 15:39						p-944
ev-id	unpack delivery	22.12.2020 16:12					d-11	
ev-ya	register co	23.12.2020 09:40	$[\![b^{-5}]\!]$	cp1		co-891		
ev-mr	pick and pack	23.12.2020 10:22	$[\![b^{-5}]\!]$	cp2		co-891		p-951
ev-sj	send parcel	23.12.2020 10:38						p-951
ev-oo	place in stock	28.12.2020 10:43	$[\![b^{990}]\!]$	cp2			d-11	
ev-qk	register co	28.12.2020 10:50	$[\![b^{-5}, c^{-3}]\!]$	cp1		co-925		

Figure 1, in which all information is aggregated into a single table: We see the event details, information on the related quantity operations and the involved objects of the different object types. Please note, that this is only possible as every event only refers to one object of each type and one collection point. We see that the events "register co", "pick and pack" and "place in stock" are associated with quantity operations regarding two different collection points.

As we see in the example log, we cannot determine the item level of the individual collection point based on this information. We, therefore, introduce a preliminary simplification in which we assume the existence of an *init*-event mapping each collection point to its initial item level, dated prior to every other event with a quantity operation. It is clear, that associating the quantity operations with events allows for considering sequences of quantity operations. Using a projection that keeps the sequence's length and summing over each entry's prefix, the quantity level of every collection point can be determined during any event in the log. The left of Figure 2, shows a sequence σ of 10 entries, each referring to a collection point and an item quantity, derived by ordering the quantity operations according to the timestamp of the corresponding event. By considering the initial item levels $m_{q,I}^{init}(cp_1) = [\![a^{21}, b^{25}, c^{10}]\!]$, and $m_{q,I}^{init}(cp_2) = [\![a^{31}, b^{27}, c^{6}]\!]$ it is possible to determine the development of each collection point's item level at any time within the sequence, as can be seen on the right.

4.2 Quantity Nets

One of the main benefits of process mining is its ability to identify process models capable of relaying the causalities uncovered by analysing event data – as seen in

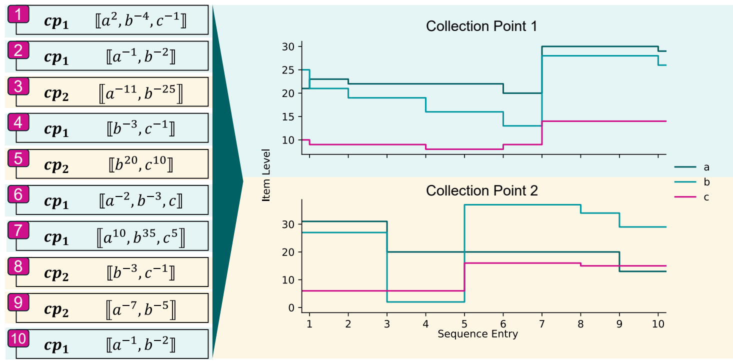


Fig. 2: Sequence of quantity operations (left) and item levels of both collection points with initial levels of $m_{q,I}^{init}(cp_1) = [\![a^{21}, b^{25}, c^{10}]\!]$ and $m_{q,I}^{init}(cp_2) = [\![a^{31}, b^{27}, c^6]\!]$.

the introduction OCPNs are not capable of doing so for non-identifiable items. In this work, we consider a collection of items as a point in which several sub-processes are decoupled, indicating an item quantity-related dependency of their process executions by adding *decoupling points* to OCPNs.

Definition 6 (Labelled Quantity Net). *A labelled quantity net is a tripartite graph $QN = (P, T, DP, F, F_{var}, pt, l)$, with P a set of places, T a set of transitions, $DP \subseteq \mathcal{U}_{cp}$ a set of decoupling points, $F \subseteq ((P \times T) \cup (T \times P) \cup (T \times DP) \cup (DP \times T)$ a set of arcs, $F_{var} \subseteq (F \cap ((P \times T) \cup (T \times P))$ a subset of variable arcs between places and transitions, $pt : P \rightarrow \mathcal{U}_{ot}$ a mapping of places to object types, $l : T \nrightarrow \mathcal{U}_{act}$ a function assigning activity names to transitions.*

The semantics of OCPNs are described by bindings, which we keep unchanged but refer to the binding function as *place binding*. The binding of the quantity net is composed of the place binding and a valid quantity-binding b^q, thus $b = (t, b^p, b^q)$. A quantity binding is a function $b^q : DP \nrightarrow \mathcal{I}(I)$ describing a quantity item for decoupling points, the semantics of which are considered equivalent to those of quantity operations for collection points. A binding is valid, if it only assigns a function value to places and decoupling points connected to the transition it refers to. Depending on the type of arc, the change in the item level $m_{q,I}(dp) \xrightarrow{q} m'_{q,I}(dp)$ by an item quantity $q \in \mathcal{I}(I)$ is either an addition $m'_{q,I}(dp) = m_{q,I}(dp) \oplus Q$ (incoming arc) or a subtraction $m'_{q,I}(dp) = m_{q,I}(dp) \ominus q$ (outgoing arc). Please note, that these semantics can lead to an increase in the item level if a negative item quantity is subtracted from a negative item level. Figure 3 shows an example, in which, for simplicity, only the transition, decoupling points and quantity binding function (graph notation) are detailed. The execution of b_1 fires transition t_1 and adds $[\![a^2, c]\!]$ to dp_1's current item level, $m_{q,I}(dp_1) \xrightarrow{b_1^q} m'_{q,I}(dp_1) = [\![a^6, b^6, c^3]\!]$, leaving dp_2's item level unchanged. Executing b_2 removes items from dp_1, $m_{q,I}(dp_1)'' = m_{q,I}(dp_1)' \ominus [\![a, b^2]\!]$, and adds $[\![b, c]\!]$ to dp_2, so that $m''_{q,I}(dp_2) = m_{q,I}(dp'') \oplus b_2^q(dp_2)$.

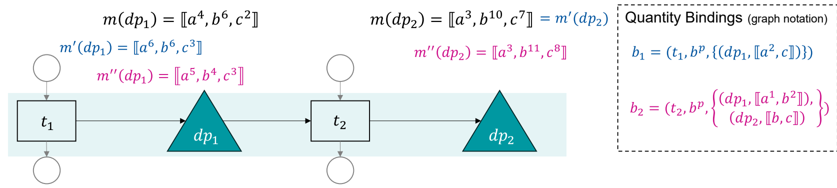


Fig. 3: The individual processes of the running example can be connected through a decoupling point, despite a lack of item identifiers.

As quantity nets offer the same representation and semantics as OCPNs, they do not limit the modelling capabilities, yet offer a further representation of behaviour. The decoupling points represent the connection of otherwise disconnected processes as well as an inter-dependency of a collection of items and the process. We briefly describe a very basic form of discovery by imposing two strong requirements. The first is that only fully positive and fully negative item quantities are included in the QEL. The second is that all events belonging to the same activity must "agree" in their impact on a collection point – there are no quantity operations by events in which one refers to a fully positive, and the other refers to a fully negative item quantity for the same collection point. Given a QEL fulfilling these requirements, the OCPN discovered by the event log can be extended by first adding each collection point as a decoupling point. Subsequently, arcs between the activities referring to at least one quantity operation and the corresponding collection point are added – the arc's direction depends on the sign of the quantity operation.

5 Example Application

The main goal of inventory management is maintaining a continuous capability to meet variable and uncertain demand while minimising cost. Current approaches for inventory management consider specific parameters related to item quantities and lead times. Still, they cannot additionally consider the end-to-end process leading to increases and decreases of stock levels [5]. This section applies the previously introduced framework to increase its comprehension and demonstrate how it enhances process mining's capabilities. To do so, we present an analysis of a simulated QEL[1] describing a typical example for a decoupled process: inventory management. The log describes a process similar to the process used as the running example – the only difference is that customer orders can loop around the activity "pick and pack". Figure 5 shows the corresponding quantity net created by extending the discovered OCPN. The selected log fulfils the criteria for discovering quantity nets, as it has one activity referring to fully positive quantity operations ("place in stock") and another ("pick and pack") with only negative item quantities.

The log shows a decoupled process, as significantly more customer orders are processed than replenishment orders placed: One delivery of albums covers

[1] Link to the data: `https://git.rwth-aachen.de/ninagraves/intro_qrpm`

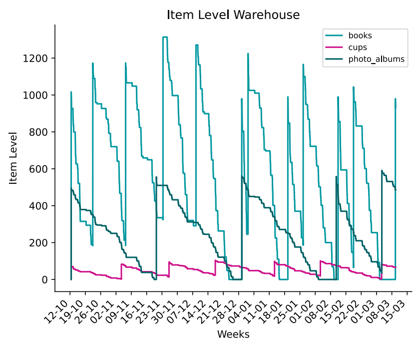


Fig. 4: Warehouse's Item level over time.

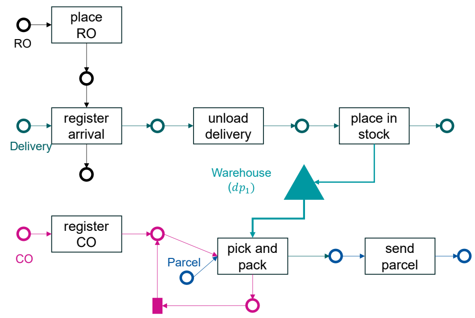


Fig. 5: Quantity net of example process.

about 519 customer orders in a turnover time of 33 days, every delivery of books lasts for about eight business days, and delivered cups last for 17 days. Figure 6 shows the distribution of the number of ordered items per type (only considering orders with a demand for this item type) per customer order – assuming the accumulated item quantities removed regarding the same customer order represent the ordered quantity. The average customer order arrives 15.6 times per business day and refers to an item quantity of 1.35 photo albums, 6.9 books and 0.29 cups. In comparison, 21 deliveries arrived in the same period. Eleven deliveries contain 980 or 990 books, six include 81 or 82 cups, and four deliveries add 556 to 560 photo albums to the warehouse. An overview of the delivered (and assumed to be ordered) items can be found in Figure 7.

Further process insights can be gained from the QEL by considering the item level of the warehouse at the time specific events were executed. A closer look at the replenishment orders shows that the time between the placement of replenishment orders varies throughout the process, making a time-based dependency unlikely. Taking the item level at the time of each order's placement into account (added as a shade to Figure 7), we see that the sum of the current item level and the requested quantity appear to be somehow connected, indicating a quantity-related dependency of the event's execution on the item level. This indirect dependency on the item level is not depicted in the quantity net, and further investigation is out of this work's scope.

The process model in Figure 5 suggests several executions of the activity "pick and pack" regarding the same customer order. For 106 customer orders, the activity was executed several times – removing items from the warehouse and sending a parcel to a customer every time. Closer consideration of the timestamps shows that this behaviour is not distributed evenly over the period but aggregates selectively. By applying standard process mining techniques to detect the waiting times, we further see that the average waiting time of objects before the activity "pick and pack" is performed is up to 54% higher than in other periods. The warehouse's item levels reveal that this behaviour coincides with stock-outs of books or albums. Within these periods, only a few customer orders are processed, none leading to a removal of items of the out-of-stock item type. Shortly after,

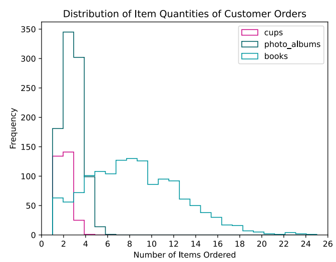


Fig. 6: Demand customer orders.

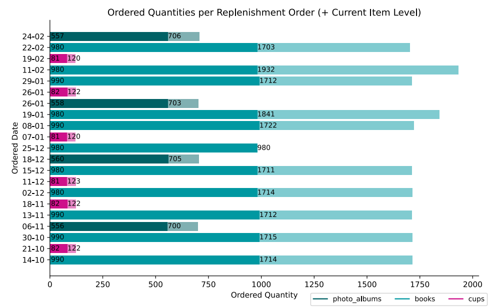


Fig. 7: Ordered items per replenishment order.

the density of events removing items of this type is higher than usual; this can also be seen in the steep decreases after the arrival of deliveries in Figure 4.

The example of a quantity-related analysis provided in this section revealed dependencies that could not have been uncovered with standard object-centric process mining techniques. Despite the semantics of the quantity net not being able to depict all of them, they are capable of (1) displaying the two processes as decoupled and (2) depicting the dependency of the execution of the activity “pick and pack” on the collection point’s item level.

6 Conclusion

In the high uncertainty and variability within current fast-paced environments, managing logistics processes requires increased transparency over the impacting end-to-end processes [13]. This presentation of early-stage research provides a foundation for the application of process mining for logistics processes by allowing the joint analysis of item quantities and end-to-end processes. The exemplary analysis indicates that the framework can capture direct dependencies between activities and known collection points and its support in revealing further dependencies. The extraction of the required quantity event log, the detection of additional quantity-related dependencies, and supporting software are future research topics. Additionally, the consideration of collection points supports the analysis of process networks, thereby serving as an abstraction for federated process mining. We conclude the presented framework as a promising first step in enabling process mining techniques to consider quantities instead of identities.

Acknowledgement

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC-2023 Internet of Production - 390621612. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. van der Aalst, W.M.: Federated Process Mining: Exploiting Event Data Across Organizational Boundaries. In: Proceedings - 2021 International Conference on Smart Data Services, SMDS 2021. pp. 61–71. IEEE (2021)
2. van der Aalst, W.M.: Process Mining: A 360 Degree Overview. In: Process Mining Handbook, vol. 448, pp. 3–34. Springer (2022)
3. van der Aalst, W.M., Berti, A.: Discovering Object-centric Petri Nets. Fundamenta Informaticae **175**(1-4), 1–40 (2020)
4. Becker, T., Intoyoad, W.: Context Aware Process Mining in Logistics. In: Procedia CIRP. vol. 63, pp. 557–562. Elsevier (2017)
5. van Cruchten, R.M.E.R., Weigand, H.H.: Process mining in logistics: The need for rule-based data abstraction. In: RCIS Proceedings. pp. 1–9. IEEE (2018)
6. Engel, R., van der Aalst, W.M., Zapletal, M., Pichler, C., Werthner, H.: Mining Inter-organizational Business Process Models from EDI Messages: A Case Study from the Automotive Sector. In: CAiSE 2012. vol. 141, pp. 222–237. Springer (2012)
7. Friederich, J., Lugaresi, G., Lazarova-Molnar, S., Matta, A.: Process Mining for Dynamic Modeling of Smart Manufacturing Systems: Data Requirements. In: Procedia CIRP. vol. 107, pp. 546–551. Springer (2022)
8. Galic, G., Wolf, M.: Global Process Mining Survey 2021 - Delivering Value with Process Analytics – Adoption and Success Factors of Process Mining. Tech. rep., Deloitte (2021)
9. Garcia, C.d.S., et al.: Process mining techniques and applications – A systematic mapping study. Expert Systems with Applications **133**, 260–295 (2019)
10. Gerke, K., Claus, A., Mendling, J.: Process Mining of RFID-Based Supply Chains. In: 2009 CEC. pp. 285–292. IEEE (2009)
11. Gerke, K., Mendling, J., Tarmyshov, K.: Case Construction for Mining Supply Chain Processes. In: Business Information Systems. vol. 21, pp. 181–192. Springer (2009)
12. Ghisellini, P., Cialani, C., Ulgiati, S.: A review on circular economy. Journal of Cleaner Production **114**, 11–32 (2016)
13. Jacobi, C., Meier, M., Herborn, L., Furmans, K.: Maturity Model for Applying Process Mining in Supply Chains. Logistics Journal p. Issue 12 (2020)
14. Knoll, D., Reinhart, G., Prüglmeier, M.: Enabling value stream mapping for internal logistics using multidimensional process mining. Expert Systems with Applications **124**, 130–142 (2019)
15. Knoll, D., Waldmann, J., Reinhart, G.: Developing an internal logistics ontology for process mining. In: Procedia CIRP. vol. 79, pp. 427–432. Elsevier (2019)
16. Lau, H., Ho, G., Zhao, Y., Chung, N.: Development of a process mining system for supporting knowledge discovery in a supply chain network. International Journal of Production Economics **122**(1), 176–187 (2009)
17. Martin, N., Pufahl, L., Mannhardt, F.: Detection of batch activities from event logs. Information Systems **95**, 101642 (2021)
18. Pika, A., Ouyang, C., Ter Hofstede, A.H.M.: Configurable Batch-Processing Discovery from Event Logs. ACM **13**(3), 1–25 (2022)
19. Pufahl, L.: Modeling and executing batch activities in business processes. PhD Thesis, Universität Potsdam (2018)
20. Rafiei, M., van der Aalst, W.M.: An Abstraction-Based Approach for Privacy-Aware Federated Process Mining. IEEE Access **11**, 33697–33714 (2023)
21. Wen, Y., Chen, Z., Liu, J., Chen, J.: Mining batch processing workflow models from event logs. Concurrency and Computation **25**(13), 1928–1942 (2013)