




# Unveiling Bottlenecks in Logistics: A Case Study on Process Mining for Root Cause Identification and Diagnostics in an Air Cargo Terminal

Chiao-Yun Li<sup>1,2</sup>, Tejaswini Shinde<sup>1</sup>, Wanyi He<sup>3</sup>, Sean Shing Fung Lau<sup>3</sup>, Morgan Xian Biao Hiew<sup>3</sup>, Nicholas T. L. Tam<sup>3</sup>, Aparna Joshi<sup>1</sup>, and Wil M. P. van der Aalst<sup>1,2</sup>

<sup>1</sup> RWTH Aachen University, Aachen, Germany  
{tejaswini.shinde,aparna.joshi}@rwth-aachen.de,  
wvdaalst@pads.rwth-aachen.de

<sup>2</sup> Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany  
chiaoyun.li@pads.rwth-aachen.de

<sup>3</sup> Hong Kong Industrial Artificial Intelligence and Robotics Centre Limited, Shatin, NT, Hong Kong  
{dorahe,seanlau,morganhiew,nicholastam}@hkflair.org

**Abstract.** To improve processes in logistics, it is crucial to understand the factors influencing performance. To achieve this, process mining utilizes *event data* to extract insights into operational processes. In this paper, we present a case study conducted in an air cargo terminal, where process mining is applied to event data collected during package distribution. The primary objective is to identify the root causes of bottlenecks in the system. However, practical limitations, including noisy sensor data, scalability challenges, and abstraction limitations, require a different approach than conventional process mining projects. Building upon existing process mining techniques, we develop a two-fold approach to identify root causes at the data level and provide diagnostics at the business level. Through a comprehensive analysis of the provided datasets, we substantiate the effectiveness and practical applicability of our approach in analyzing root causes.

**Keywords:** Process mining · Logistic · Root cause identification · Root cause diagnostics · Performance Spectrum · Case study

## 1 Introduction

Efficient processes are crucial for success in the logistics industry. Businesses must understand their process performance to attain this objective. Material

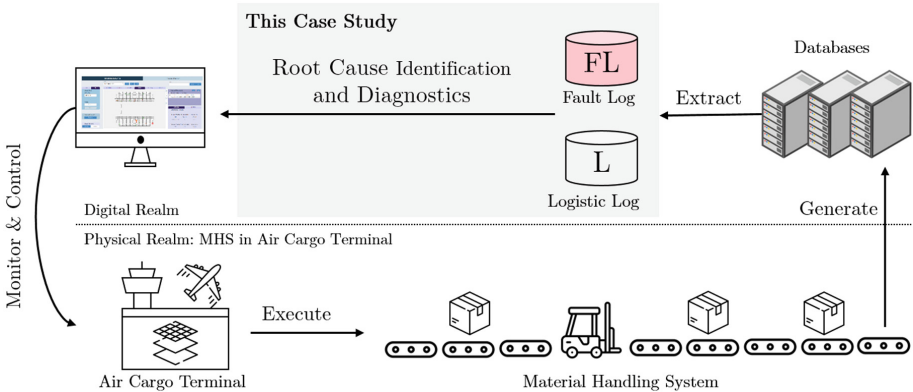
---

This work was supported by the InnoHK funding launched by Innovation and Technology Commission, Hong Kong SAR. Additionally, we thank Sebastiaan van Zelst for his support.

Handling Systems (MHS) are vital to efficient logistics management, facilitating the timely movement of materials. Evaluating MHS performance enables organizations to optimize their operational processes by minimizing delays, reducing manual labor, and overcoming obstacles encountered during the distribution [11].

*Process mining* aims to enhance the understanding of operational processes by utilizing *event data*, consisting of records of process execution stored in information systems. A process mining project typically involves discovering a process model, which abstracts the process behavior, from event data using *discovery techniques* [2, 14]. The model is then compared with the expectations to identify deviations or to *repair* the model [3, 18]. Process efficiency is evaluated by analyzing the model annotated with performance information derived from the event data [10]. In the case of a large system, activities (i.e., well-defined process steps) can be abstracted, alongside performance information aggregated, to reduce the process complexity for human analysis [4]. Leveraging the context provided by the model and domain knowledge, one can diagnose bottlenecks and optimize processes. Throughout the process described, it is evident that a reliable model depicting the behavior in real life is central to a process mining project.

In this study, we aim to discern the underlying causes of bottlenecks affecting the distribution of packages within the MHS of an air cargo terminal, as depicted in Fig. 1. The figure delineates our specific focus and case study scope. Using a logistic log, which captures package distribution data, and a fault log, encompassing information on equipment malfunctions and maintenance, we identify and diagnose the root causes of detected bottlenecks. Subsequently, these identified root causes are mapped onto the transport layout for business owners to gain a visual understanding, aiding them in monitoring and further controlling the system effectively. However, we encounter challenges that compel us to deviate from the conventional process mining approach [7].



**Fig. 1.** In this case study, we aim to uncover package distribution inefficiencies at an air cargo terminal by analyzing logistic and fault logs. The resulting insights will inform an integrated solution for optimizing the distribution process.

- Noisy event data: The system consists of thousands of equipment pieces with sensors generating event data. Sensor data are prone to noise [12], which can distort the actual relationships between different pieces of equipment.
- Scalability limitations: Given the large number of pieces of equipment and the substantial volume of event data, existing open-source tools [17] cannot adequately support the performance requirements to discover a model and to interactively explore the process performance using the model.
- Abstraction limitations: Typically, abstraction resolves complexity and scalability challenges caused by the excessive number of concepts in a process, like the conveyor equipment in this case study. Yet, due to the queuing behavior and equipment faults within the process, abstraction may lead to misleading conclusions since it does not fully capture the queuing phenomenon.

To address these challenges, we devised a solution that delivers transparent and reliable results, empowering business owners to make well-informed decisions regarding their service efficiency. We uncover unbiased behavior and identify incidents that contribute to bottlenecks, including identifying package distributions that cause bottlenecks on specific pieces of equipment at particular points in time. Despite the constraints, we devise a two-fold approach that uncovers inefficiencies and provides explanations without relying on a process model. First, we programmatically detect the root causes of bottlenecks at the *data level* to narrow down the analysis scope. Building upon the findings, we derive diagnostics at the *business level*, leveraging the *Performance Spectrum Miner* (PSM) [5]. Our approach is quantitatively evaluated and integrated into the system, aiming to enhance the overall service performance within the air cargo terminal.

In Sect. 2, we introduce the techniques and notations applied. Section 3 describes the available datasets in the case study. Package distribution behavior is depicted in Sect. 4, while root cause identification is in Sect. 5. Sect. 6 demonstrates the results. We discuss the related work in Sect. 7. Lastly, Sect. 8 summarizes the case study and discusses future work.

## 2 Background

In this section, we introduce the techniques applied for root cause analysis and mathematical notations.

### 2.1 Performance Spectrum

PSM is a visual analytic tool that formats the performance of activities in a process within the context of a *case* (i.e., a process instance) [5]. By visualizing how cases progress through activities over time, the tool enables the observation of efficiency dynamics and facilitates the analysis of interactions between cases. Numerous extensions have been developed to quantify [5], predict [6], and visualize the performance within the context of a process model [1]. In our case study, we employed the implementation which enables the interactive exploration of performance based on a process model supported by PSM [1].

However, scalability emerges as a practical challenge in this case study. First, the existing discovery techniques within the tool exhibit limited scalability when faced with a significant number of activities, as exemplified in the case study with thousands of equipment pieces. As the complexity of the model increases, it progressively poses greater challenges for human analysts to effectively identify all root causes effectively. Given the case study's scale, it is challenging to pinpoint the areas requiring analysis without adequate guidance for navigating the process model. To address these issues, we have devised a programmatic approach that detects root causes at the data level. By narrowing down the analysis scope, we leverage PSM to facilitate the subsequent diagnosis at the business level.

## 2.2 Notations

Let  $X$  be an arbitrary set. A sequence is a function  $\sigma: \{1, 2, \dots, n\} \rightarrow X$ , where  $\sigma = \langle x_1, x_2, x_2, \dots, x_n \rangle$  is a sequence over  $X$ , and  $\sigma(i) = x_i$  denotes the  $i^{th}$  element in  $\sigma$ . We denote  $|\sigma|$  as the length of  $\sigma$  and  $X^*$  as the set of all possible sequences over  $X$ . We write  $x \in \sigma \iff \exists k \in \mathbb{N}$  s.t.  $1 \leq k \leq |\sigma|$  and  $\sigma(k) = x$ . The index of  $x \in \sigma$  is denoted as  $\sigma^{-1}(x) \in \mathbb{N}$  and  $\sigma^{-1}(x) = \min\{1 \leq i \leq |\sigma| \mid \sigma(i) = x\}$ . A *path* from  $m$  to  $n$  in  $\sigma$ , where  $1 \leq m < n \leq |\sigma|$ , refers to a segment of  $\sigma$ , written as  $path_\sigma(m, n) = \langle x_m, x_{m+1}, \dots, x_n \rangle$ . Note that  $path_\sigma(1, |\sigma|) = \sigma$ .

## 3 Datasets

The case study incorporates two logs: a logistic log and a fault log.<sup>1</sup> The logistic log captures the package distribution within the system and the fault log documents fault instances related to the conveyor equipment.

### 3.1 Representation of Logistic Log

We represent the logistic log as an *event log*, i.e., a typical input for most process mining techniques.  $\mathcal{U}_{pkg}$  is the universe of package identifiers,  $\mathcal{U}_{eqt}$  is the universe of equipment identifiers, and  $\mathcal{U}_{time}$  is the universe of timestamps.

**Definition 1 (Event Log).**  $\mathcal{E}$  is the universe of events.  $e \in \mathcal{E}$  represents a data sample collected by sensors for the package distribution, which is characterized with the corresponding package identifier  $\pi_{pkg}(e) \in \mathcal{U}_{pkg}$ , equipment identifier  $\pi_{eqt}(e) \in \mathcal{U}_{eqt}$ , and the arrival timestamp  $\pi_{arr}(e) \in \mathcal{U}_{time}$  of  $\pi_{pkg}(e)$  on  $\pi_{eqt}(e)$ . An event log  $L$  is a set of events  $L \subseteq \mathcal{E}$ .

A case is a collection of events describing a complete package distribution, i.e., given  $pid \in \mathcal{U}_{pkg}$ , the case of  $pid$  is  $c = \{e \in L \mid \pi_{pkg}(e) = pid\}$ . The trace of a case  $c$ , denoted as  $\pi_{trace}(c)$ , is a chronologically ordered sequence of events in a case, where  $\pi_{trace}(c) = \langle e_1, e_2, \dots, e_{|c|} \rangle$  such that  $\forall 1 \leq i < j \leq |c|, \pi_{arr}(e_i) \leq \pi_{arr}(e_j)$ . Additionally, the time that a package distribution exits the system is provided and we write as  $\pi_{exit}(c) \in \mathcal{U}_{time}$ , where  $\pi_{exit}(c) \geq \max\{\pi_{arr}(e) \mid e \in c\}$ .

<sup>1</sup> For confidentiality, we pseudo-anonymized the datasets, preserving the relative relation between incidents. In this case study, we only present pertinent attributes.

**Table 1.** An excerpt of an event log  $L$ . Each row represents an event describing the arrival of a package (represented by PKG) on a specific piece of equipment (represented by EQT) at a particular time (represented by ARR). For ease of reference, the event identifier (represented by Event ID) is provided using the row index, e.g., the first event is labeled as  $e_1$ . The completion time of the distribution is also included, denoted as EXIT, providing additional details about the distribution.

Event ID	PKG ( $\pi_{pkg}(e)$ )	EQT ( $\pi_{eqt}(e)$ )	ARR ( $\pi_{arr}(e)$ )	EXIT ( $\pi_{exit}(c)$ )
1	2365884457	HXUF1928	2023-05-05 12:12:34	2023-05-05 14:00:31
2	2365884457	TFGT3578	2023-05-05 12:12:53	2023-05-05 14:00:31
3	2365884457	UENF3008	2023-05-05 13:59:51	2023-05-05 14:00:31
4	2459856232	GJWK4805	2023-05-05 13:33:56	2023-05-05 17:44:28
5	2459856232	UENF3008	2023-05-05 17:38:00	2023-05-05 17:44:28
6	2459856232	ITSC0915	2023-05-05 17:38:41	2023-05-05 17:44:28
7	2459856232	LKHS8902	2023-05-05 17:38:54	2023-05-05 17:44:28
8	2459856232	CJIF5952	2023-05-05 17:39:06	2023-05-05 17:44:28

Table 1 displays an excerpt from the event log, illustrating the package distribution. For example, the case  $c = \{e_1, e_2, e_3\}$  describes the distribution of package 2365884457, which undergoes three pieces of equipment in the system. It first arrives on  $\pi_{eqt}(e_1) = \text{HXUF1928}$  at  $\pi_{arr}(e_1) = 12:12:34$ , then moves to  $\pi_{eqt}(e_2) = \text{TFGT3578}$  at  $\pi_{arr}(e_2) = 12:12:53$ , and finally reaches  $\pi_{eqt}(e_3) = \text{UENF3008}$  at  $\pi_{arr}(e_3) = 12:59:51$  before leaving the system at  $\pi_{exit}(c) = 14:00:31$ .

### 3.2 Fault Log

A fault refers to an incident that occurs on a piece of equipment and is unrelated to the package distribution process. A fault log is a compilation of such incidents, which we formalize as follows.

**Definition 2 (Fault Log).**  $\mathcal{F}$  is the universe of faults.  $f \in \mathcal{F}$  is a fault, which is characterized by the corresponding equipment identifier  $\pi_{eqt}(f) \in \mathcal{U}_{eqt}$ , downtime of  $\pi_{dt}(f) \in \mathcal{U}_{time}$ , and the corresponding uptime  $\pi_{ut}(f) \in \mathcal{U}_{time}$  where  $\pi_{dt}(f) < \pi_{ut}(f)$ . A fault log  $FL$  is a set of faults  $FL \subseteq \mathcal{F}$ . Since at most one fault can occur on a piece of equipment at any point in time, the faults on the equipment form a sequence of faults in time, i.e.,  $\forall f_1 \in FL \forall f_2 \in FL, f_1 \neq f_2 \implies (\pi_{dt}(f_1) \geq \pi_{ut}(f_2)) \vee (\pi_{dt}(f_2) \geq \pi_{ut}(f_1))$ .

Table 2 showcases a sample from the fault log, with each row depicting an instance of a fault occurrence. For instance, equipment UENF3008 experiences a fault from 12:15:23 to 12:16:49. Notably, the excerpt demonstrates a sequential occurrence of five faults on UENF3008.

## 4 Behavioral Analysis

This section introduces identified constraints and outlines the assumptions of the system behavior, which serve as the basis for defining the bottlenecks.

**Table 2.** Every row in the dataset represents a fault occurrence in the system. A fault is uniquely identified by three key pieces of information: the piece of equipment where the fault happens (represented by EQT), the start time of the fault (represented by DOWN), and the end time of the fault (represented by UP). Likewise, we provide the fault identifier (represented by Fault ID) using the row index for ease of reference.

Fault ID	EQT ( $\pi_{eqt}(f)$ )	DOWN ( $\pi_{dt}(f)$ )	UP ( $\pi_{ut}(f)$ )
1	UENF3008	2023-05-05 12:15:23	2023-05-05 12:16:49
2	UENF3008	2023-05-05 12:28:07	2023-05-05 12:30:26
3	UENF3008	2023-05-05 12:31:16	2023-05-05 12:38:00
4	UENF3008	2023-05-05 12:47:23	2023-05-05 12:49:49
5	UENF3008	2023-05-05 13:11:40	2023-05-05 13:30:00
6	UAZB1814	2023-05-05 14:58:35	2023-05-05 15:16:05

#### 4.1 Package Distribution – Constraints and Assumptions

The analysis reveals the following constraints of the package distribution. In collaboration with domain experts, we validate the constraints and impose specific assumptions to facilitate the identification of bottlenecks.

**Departure Time.** We assume that a package departs from one piece of equipment at the same time as it arrives on the next piece of equipment along its trajectory. Let  $L \subseteq \mathcal{E}$ . Given a case  $c \subseteq L$ , let  $\sigma = \pi_{trace}(c)$ . For an event  $e \in \sigma$ , we define the function  $dep_c(e) = \pi_{arr}(\sigma(\sigma^{-1}(e) + 1)) \iff \sigma^{-1}(e) < |\sigma|$  and  $dep_c(e) = \pi_{exit}(c) \iff \sigma(|\sigma|) = e$ . We name the duration as the *dwelt time* of a package on a piece of equipment. As an illustration, considering Table 1, we assume that package 2365884457 departs TFGT3578 at 13:59:51, and the dwelt time of 2365884457 on TFGT3578 is 1 h, 46 min, and 58 s.

**Equipment Capacity.** At any given time, one equipment piece can accommodate a maximum of one package. Let  $L \subseteq \mathcal{E}$  denote an event log. Given  $eqt \in \mathcal{U}_{eqt}$ ,  $\forall e_1 \in L(\pi_{eqt}(e_1) = eqt) \forall e_2 \in L(\pi_{eqt}(e_2) = eqt), e_1 \neq e_2 \implies (\pi_{arr}(e_1) > dep_c(e_2)) \vee (\pi_{arr}(e_2) > dep_c(e_1))$ . This constraint leads to a queuing behavior in the system, wherein packages are distributed in a sequential manner, allowing a package to move to the next piece of equipment only when the preceding package in its trajectory departs from that piece of equipment.

**Fault Impact on Package Distribution.** The faults in the fault log can be classified into three categories: warning, maintenance, and *real* fault. Warning and maintenance faults do not have any impact on package distribution. However, a real fault disrupts the distribution process and may also affect the overall system performance. Considering a real fault  $f \in \mathcal{F}$  and an event log  $L \subseteq \mathcal{E}$ , during the fault,  $\pi_{eqt}(f)$  is unable to send or receive any packages. In other

words, there does not exist an event  $e \in L$  such that  $\pi_{dt}(f) < \pi_{arr}(e) < \pi_{ut}(f)$  or  $\pi_{dt}(f) < dep_c(e) < \pi_{ut}(f)$ . Since there are no specific attributes available to directly determine the category of a fault in the fault log  $f \in FL$ , we define a real fault using the function  $real(f, L) \rightarrow \mathcal{F}$  if there are no packages arriving or departing from  $\pi_{eqt}(f)$  during the time period of  $\pi_{dt}(f)$  and  $\pi_{ut}(f)$ .

## 4.2 Bottleneck Definition

A *bottleneck event* refers to an event indicating that a package remains on a piece of equipment for a longer duration than anticipated. Let  $\mathcal{U}_{dur}$  be the universe of time durations, e.g., 5 min, 3 s, etc. A bottleneck event in the system is defined as follows.

**Definition 3 (Bottleneck Event).** *Let  $eqt \in \mathcal{U}_{eqt}$  be a piece of equipment identifier, and  $thr(eqt) \in \mathcal{U}_{dur}$  denotes the theoretical service time of  $eqt$ . Let  $L$  be an event log. Given an event  $e \in L$  and a case  $c$ , where  $e \in \pi_{trace}(c)$ ,  $e$  is a bottleneck event iff  $dep_c(e) - \pi_{arr}(e) > thr(\pi_{eqt}(e))$ .*

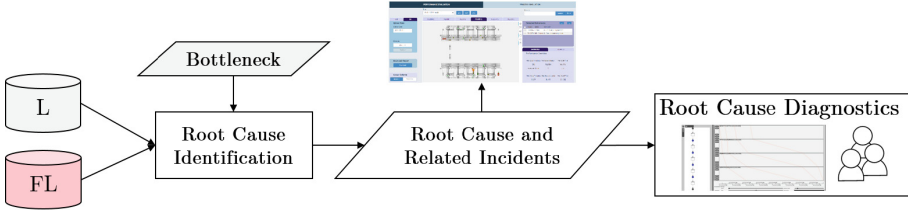
A baseline for comparison is crucial when evaluating process performance. In this case study, since the efficiency is significantly influenced by the dynamic nature of cases within the system, a rigid benchmark is impractical. Therefore, we employ statistical metrics as a benchmark to evaluate the performance of the equipment. Specifically, we establish benchmarks for each equipment type by examining the first quartile of dwell time per equipment type, recognizing that the dwell time of a piece of equipment varies depending on its type. For instance, the dwell time of a package on a lift shaft differs from that on a conveyor belt. Bottleneck events are identified when the corresponding dwell times exceed the benchmark. Throughout the paper, we refer to these events as bottlenecks.

## 5 Root Cause Identification and Diagnostics

Considering the inherent complexity of the system, we devise a two-stage approach for root cause analysis, as illustrated in Fig. 2. In the first stage, given a bottleneck, we narrow down the scope by identifying the root cause at the data level—extracting specific location and timing information that triggers the bottlenecks. Next, we gather the relevant incidents and collaborate with domain experts to visualize the entire process leading to the bottleneck using PSM. This collaborative process facilitates a comprehensive examination and diagnosis of the identified root causes from a business perspective. We integrate root cause identification into our partner’s system, visually displaying the bottleneck and its cause on their logistic map, enhancing stakeholder understanding.

### 5.1 Root Cause Identification

The scale and complexity of the system, comprising approximately 800,000 events from around 5,700 equipment pieces, present significant challenges in



**Fig. 2.** Two-Stage Root Cause Analysis. The first stage involves narrowing down the scope through root cause identification. The resulting root cause is visualized on the logistic map, while the process leading to the bottleneck is visualized with PSM, allowing for collaborative discussions with stakeholders.

systematically uncovering the underlying causes of bottlenecks. To address this, root cause identification narrows down and extracts a subset of events and/or faults that contribute to bottleneck occurrences. By focusing on this subset, we efficiently pinpoint the incidents that are most relevant to our analysis, ensuring a more targeted approach. In this context, a root cause is defined as an incident on a piece of equipment that triggers a specific bottleneck. The formal definition of a root cause is outlined below.

**Definition 4 (Root Cause).** *Let  $L \subseteq \mathcal{E}$  and  $FL \in \mathcal{F}$ . Given a bottleneck  $bn \in L$ , a root cause  $rc$  is an incident occurring on a piece of equipment in the system  $rc \in L \cup FL$  that causes  $bn$ .*

Prior to the algorithm, we establish two conditions for identifying root causes. First, we determine if a bottleneck occurs due to a fault occurring on the piece of equipment associated with it. Since real faults do not allow for package reception nor sending, a package gets *stuck due to fault* if it arrives on the piece of equipment before a fault happens and remains there until the fault is repaired. During this period, the fault prevents the piece of equipment from processing any packages, resulting in packages becoming *stuck* until the fault is resolved.

**Definition 5 (Stuck due to Fault).** *Let  $L \subseteq \mathcal{E}$  and  $FL \subseteq \mathcal{F}$ . Given a bottleneck  $bn \in L$ ,  $stuck(bn, FL) = \sigma \in \mathcal{F}^*$  extracts a sequence of root causes where  $1 \leq i \leq |\sigma|(\pi_{eqt}(\sigma(i)) = \pi_{eqt}(bn) \wedge \pi_{arr}(bn) < \pi_{dt}(\sigma(i)) \wedge dep_c(bn) > \pi_{ut}(\sigma(i)))$ .*

If a bottleneck is not due to a fault in the associated equipment piece, we investigate the condition of the subsequent equipment along its trajectory. Equipment condition is determined by the incidents at a specific time. In this case study, two types of incidents are considered: an event indicating the availability of the equipment piece (i.e., a package is present on the equipment piece) and a fault indicating the unavailability of the equipment piece.

**Definition 6 (Equipment Condition).** *Let  $L \subseteq \mathcal{E}$ ,  $eqt \in \mathcal{U}_{eqt}$ , and  $t \in \mathcal{U}_{time}$ .  $CON_{occ}: \mathcal{U}_{eqt} \times \mathcal{U}_{time} \times \mathcal{E} \rightarrow \mathcal{E}$ , where  $CON_{occ}(eqt, t, L) = e \in L \iff \pi_{eqt}(e) = eqt \wedge \pi_{arr}(e) < t < dep_c(e)$ . Given  $FL \subseteq \mathcal{F}$ ,  $CON_{flt}: \mathcal{U}_{eqt} \times \mathcal{U}_{time} \times \mathcal{F} \rightarrow \mathcal{F}$ , where  $CON_{flt}(eqt, t, FL) = f \in FL \iff \pi_{eqt}(f) = eqt \wedge \pi_{dt}(f) < t < \pi_{ut}(f)$ .*



Algorithm 1 outlines the identification of the root cause given a bottleneck. The algorithm checks if the bottleneck is caused by being stuck on the associated equipment piece. If no faults are detected, the algorithm proceeds to examine the condition of the equipment on the trajectory of the bottleneck and extracts the last incident until one of the following final conditions is met:

- The associated equipment piece is the last equipment piece on its trajectory;
- The associated equipment piece is empty and without a real fault;
- The associated equipment piece is at real fault.

A piece of equipment can be both at fault and occupied simultaneously. However, considering that the business owner’s primary interest lies in identifying and addressing faults, the developed method places a stronger emphasis on identifying root causes related to faults. This focus allows for a more targeted approach in determining the actions to be taken to address the identified faults.

---

**Algorithm 1.** Root Cause Identification

---

**Require:** event log  $L$ , fault log  $FL$ , bottleneck  $bn \in L$

**Ensure:** a root cause  $rc \in \mathcal{E} \cup \mathcal{F}$

```

1: if  $|stuck(bn, FL)| > 0$  then return  $stuck(bn, FL)(1)$ 
2:  $c \leftarrow$  the corresponding case of  $bn$ 
3:  $\sigma \leftarrow \pi_{trace}(c)$ 
4:  $\sigma' \leftarrow path_{\sigma}(\sigma^{-1}(bn), |\sigma|)$ 
5:  $current \leftarrow bn$ 
6: for  $1 \leq i < |\sigma'|$  do
7:   if  $\pi_{eqt}(current) = \pi_{eqt}(\sigma'(i))$  then return  $current$ 
8:    $time \leftarrow \pi_{arr}(current) + thr(\pi_{eqt}(current))$ 
9:    $next \leftarrow \pi_{eqt}(\sigma'(i + 1))$ 
10:   $f \leftarrow CON_{flt}(next, time, FL)$ 
11:  if  $real(f, L)$  then return  $f$ 
12:   $e \leftarrow CON_{occ}(next, time, L)$ 
13:  if  $e = \perp$  then return  $current$ 
14:   $current \leftarrow e$ 

```

---

## 5.2 Root Cause Diagnostics

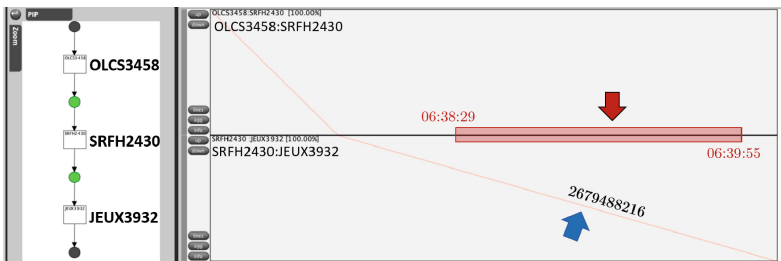
In this section, we delve into a comprehensive analysis of the process leading to bottlenecks, going beyond the identification of the root cause. We collect and analyze incidents contributing to the bottleneck since the identified root cause. This enables us to uncover the cause-effect relationships that influence bottleneck occurrences, facilitating a more profound understanding of the underlying process. To strengthen our analysis, we utilize PSM to visually represent the behavior of the incidents. Furthermore, engaging in effective discussions with domain experts provides valuable insights and perspectives. The following diagnostics illustrate their implications from a business standpoint.

*Diagnosis 1 (Stuck on Faulty Equipment).* Figure 3 depicts the diagnosis of an internal root cause where a piece of faulty equipment causes a package to be *stuck* on it, as described in Definition 5. The performance spectrum displayed on the right side depicts the efficiency of the selected places in the discovered Petri net shown on the left side. Selected places representing a single path are colored in green, while aggregated paths are colored in blue. The example shows that faulty equipment SRFH2430 *blocks* the package distribution, which is impossible to reroute without human intervention.

We project the faults onto the timeline, displaying the downtime and uptime for the respective piece of equipment. We utilize blue and red arrows to highlight bottlenecks and the corresponding root causes. This visualization is consistently applied across figures throughout the subsequent diagnostics.

*Diagnosis 2 (Waiting for Repair).* Figure 4 depicts a scenario with two bottlenecks stemming from the same root cause, specifically a fault on equipment UAZB1814. Once the fault is fixed, the distribution process resumes. Furthermore, Fig. 5 illustrates two bottlenecks resulting from a sequence of faults on equipment UENF3008. For the bottleneck on equipment LFYV0354 in case 2654852459, the first fault is the root cause, while for the bottleneck in case 2365884457, the second fault is identified as its root cause. The figure also highlights distribution prioritization, with package 2365884457 being given higher priority despite arriving later on equipment TFGT3578 due to its importance.

*Diagnosis 3 (Waiting to Exit).* Figure 6 showcases the cascading package waiting, highlighting the impact of capacity constraints on the distribution. The packages queue to exit, resulting in a sequence of bottlenecks. The root cause of the bottlenecks is identified as the package 2968579218 on equipment KXLJ5003, which is also a bottleneck itself and is observed waiting at the last piece of equipment along the distribution trajectory of the bottlenecks. The visualization emphasizes how the waiting of a single package on a piece of equipment affects subsequent distributions, leading to inefficiencies propagating throughout the system. Further investigation reveals that the root cause originates from the package 2968579218 waiting to be loaded onto an aircraft.



**Fig. 3.** Package distribution of package 2679488216 gets *stuck* at faulty equipment SRFH2430 during the distribution process. (Color figure online)

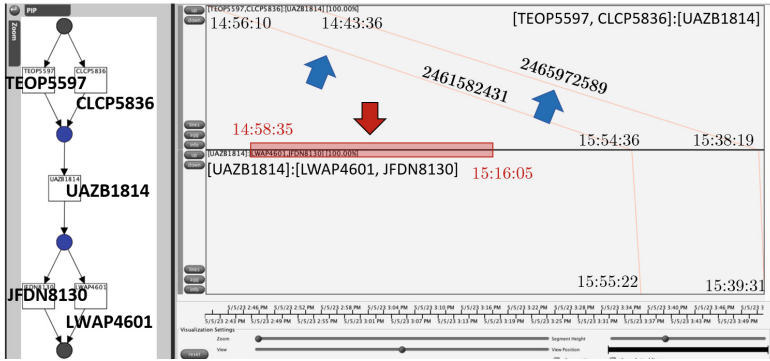


Fig. 4. Two bottlenecks waiting for faulty equipment UAZB1814 to be repaired.

*Diagnosis 4 (Unjustified Waiting).* If no incidents are identified for a bottleneck, the root cause is defined as unjustified waiting, indicating the next piece of equipment on the package’s trajectory is available for transfer without any detected incidents. Nevertheless, the distribution inexplicably ceases. While one reason could be the equipment piece serving as a storage place within the system, there are root causes that remain unexplainable from a business perspective.

By gathering and analyzing the incidents that contribute to a bottleneck, we facilitate the diagnostic at the business level through the utilization of a visual analytic method inspired by PSM. This enables the process owner to identify the appropriate measures to address the identified bottlenecks effectively.

### 5.3 Impact of Bottlenecks

Some bottlenecks may be circumvented by navigating around the identified obstacles. To identify potential *detours* and their relationship with bottlenecks,

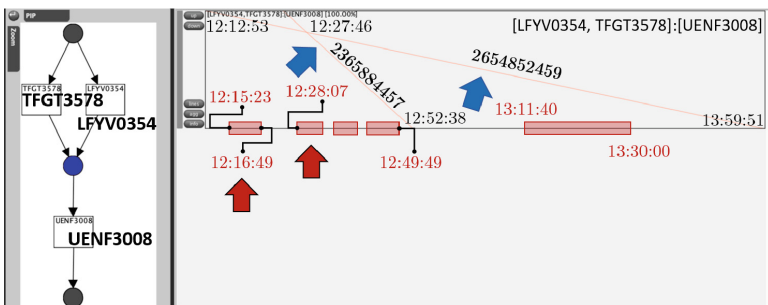
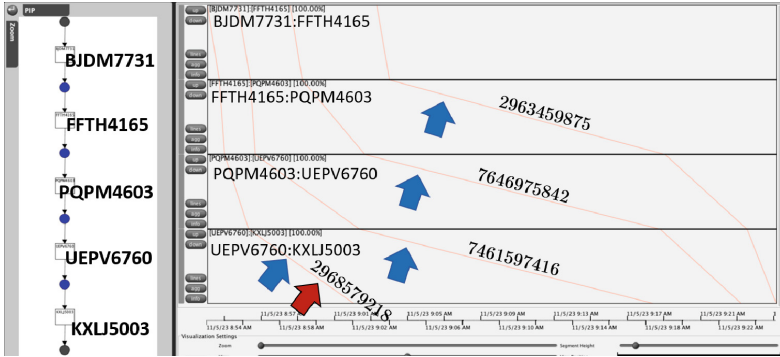


Fig. 5. Two bottlenecks due to waiting for sequential faults to be repaired. The intersection highlights the distribution priority of packages. Specific timestamps are annotated to demonstrate the relationship between the arrival time of the packages and the downtime and uptime of the faults.



**Fig. 6.** Packages waiting to be loaded onto an aircraft, where the inefficiency cascades through the equipment.

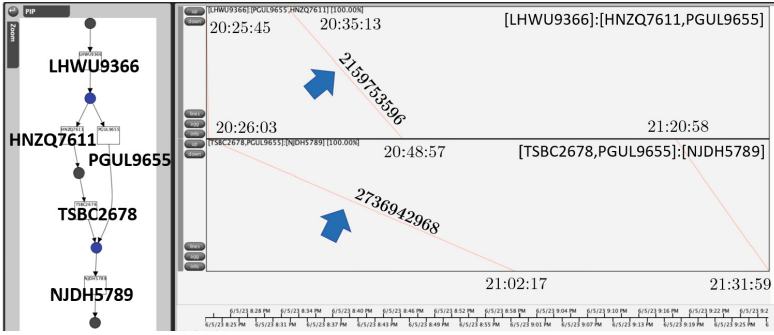
we make an assumption that cases with the same source and destination pieces of equipment follow the same *planned* route if no obstacles are encountered during distribution. Building upon this assumption, first, we extract cases that share the same source and destination as the trajectory of a bottleneck. Next, we identify the *change points* of these cases, i.e., the equipment piece where a case deviates from the originally planned route. To relate the detour with the bottleneck, we select the cases with the change points preceding the bottleneck on the route, and the events at these change points temporally take place *after* the bottleneck.

Figure 7 exemplifies the impact of a bottleneck, which results in a detour within the system. In this example, case 2159753596 *detours* on LHWU9366, i.e., the change point for its distribution, due to the bottleneck caused by the distribution of package 2736942968 on equipment PGUL9655. Additionally, package 2159753596 experiences a temporary waiting period on LHWU9366 until the decision to detour is made. As a result, the distribution of 2159753596 is compelled to deviate from its initial planned route, leading to a longer path to reach its destination. This scenario highlights how bottlenecks impact the overall distribution process, causing deviations and delays for affected cases.

By illustrating the impact of a bottleneck, we highlight that inefficiencies may not be readily observable solely based on the presence of a bottleneck. The distribution, taking a detour to circumvent the bottleneck, follows a longer route, ultimately leading to increased throughput time in its distribution process.

## 6 Results

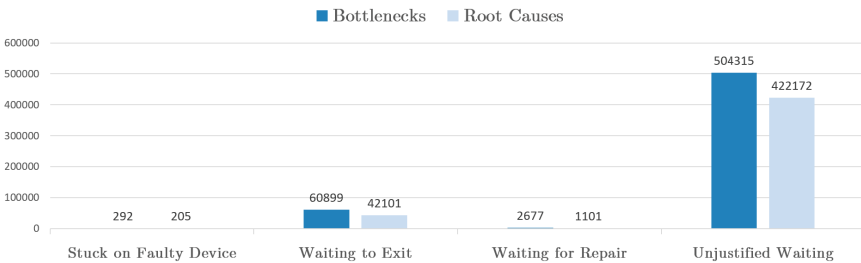
We present the quantitative results from our diagnostics, as depicted in Fig. 8. The figure provides insights into the distribution of bottlenecks based on their root causes. Notably, as the number of bottlenecks increased, we observed a trend where multiple bottlenecks shared the same root causes, highlighting their interconnectedness and shared contributing factors within the system.



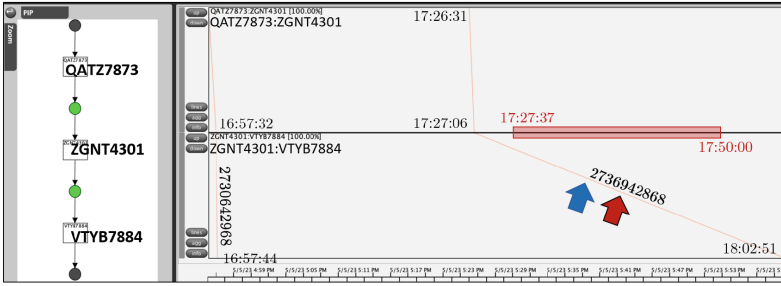
**Fig. 7.** An example highlighting the impact of a bottleneck: package 2736942968 on equipment PGUL9655 triggers a cascading effect, causing 2159753596 to experience additional waiting and detour, resulting in extended throughput time.

Interestingly, although only a small portion of bottlenecks appeared to be attributed to equipment faults, our analysis of the root causes of unjustified waiting reveals another possibility. Approximately 3% of the root causes of the unjustified waiting could be attributed to storage-related reasons, which are regarded as more of a business decision rather than operational inefficiencies. For other root causes, we identified an example that illustrates the impact of design decisions on threshold settings, as shown in Fig. 9. This instance resulted in undetected root causes, where equipment ZGNT4301 was evaluated at 17:27:20 with a 14-s threshold, while the fault occurred 17s later, causing package 2736942868 to become stuck on equipment ZGNT4301. These findings highlight the importance of thoroughly considering design choices to accurately detect and address root causes.

The identification of root causes demonstrates efficiency, with an average time of approximately 0.2s and a maximum of 2s per bottleneck. These metrics highlight a rapid identification process, influential in preserving optimal system performance. Leveraging this efficiency, the root cause identification developed is



**Fig. 8.** The number of bottlenecks and the corresponding root causes based on diagnostics.

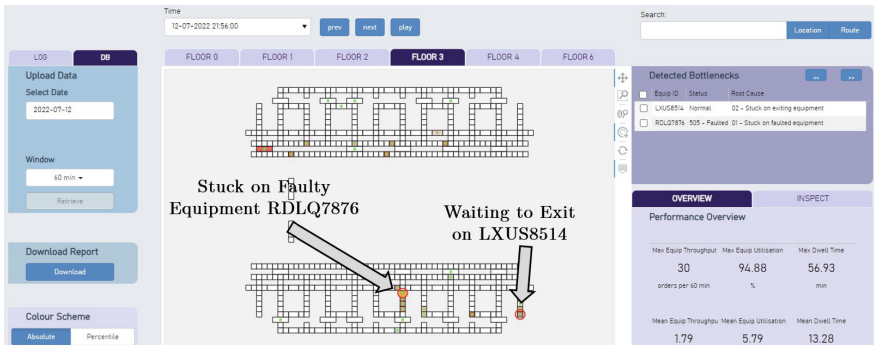


**Fig. 9.** An example of unjustified waiting, highlighting the impact of design decisions where the equipment fault occurred after evaluating the equipment condition.

integrated into the system, as exemplified in Fig. 10. Utilizing a logistic map that visually represents the equipment layout and relationships within the system, the root causes are swiftly detected based on the bottlenecks highlighted on the map. The right panel provides an interactive interface for exploring the detected root causes, enabling a comprehensive analysis of the distribution process. This integration emphasizes its practical applicability and highlights its potential to enhance operational efficiency.

## 7 Related Work

Diagnosing the root cause is crucial for optimizing service performance. Comprehensive process models are typically seen as essential for root cause diagnostics [8, 13, 16]. For instance, in one study [13], a descriptive process model with statistical metrics is used to identify root causes by observing resource status during the bottleneck time periods. Another work in [8] employs conformance checking



**Fig. 10.** Visualizing root causes of bottlenecks in the distribution process on the logistic map. note that the diagnostics are renamed for user clarity.

to diagnose bottlenecks through deviations and cause-effect correlations, demonstrated through an offshore oil and gas industry case study. However, discovering a suitable process model is often challenging due to scalability issues with discovery algorithms. Additionally, these models require a substantial amount of data for each resource, making them less applicable in scenarios where resource utilization is sparse, as presented in our case study.

Certain approaches rely heavily on knowledge-intensive domain understanding, demanding significant effort and expertise to represent complex causal relationships. While innovative methodologies show promise in representing knowledge [9, 15, 16], their effective implementation requires a high level of expertise. For example, in an approach [15], fusion-based clustering and a hyperbolic neural network are utilized to represent domain knowledge. Inspired by causality theory, the authors [9] strive to avoid imposing assumptions on the data, enhancing reliability in practical applications. In the work by Unger et al. [16], an event log derived from business lawsuits is defined and subsequently analyzed using process mining techniques. Although the analysis yields valuable insights, identifying the root causes necessitates human analysis and a profound understanding of the domain to interpret the performance metrics accurately and diagnose the underlying reasons for bottlenecks. These methods demand specialized knowledge, limiting their practical adoption and applicability.

In practical applications, scalability, accessibility to domain knowledge, and the necessity of a process model pose significant challenges. In contrast, the proposed solution automatically identifies root causes at the data level, demonstrating scalability and potential for real-time application. We emphasize transparency based on unbiased raw data and facilitate business-level interpretation through visualization using PSM.

## 8 Conclusion

In this paper, we presented a case study focusing on the identification and diagnostics of root causes in the package distribution process of an air cargo terminal. The process efficiency is closely tied to the dynamic nature of the system. We formalized the provided datasets and analyzed the observed behavior within the system. By identifying bottlenecks, we proposed a data-level method for extracting root causes and conducting targeted diagnostics. Moreover, we demonstrated the effectiveness of the visualization inspired by PSM in aiding the diagnostic process. Additionally, we showcased the impact of bottlenecks, which led to inefficiencies in the system that cannot be directly observed in individual package distributions. The results of the case study further establish the practicality and relevance of our method in real-world scenarios. For future work, we aim to extend the visualization to include the status of the equipment, which can be seen as the concept of resources in process mining, as it significantly impacts the system. Developing a visual analytic tool considering equipment or resource status would benefit scenarios similar to this case study.

## References

1. van der Aalst, W.M.P., Tacke Genannt Unterberg, D., Denisov, V., Fahland, D.: Visualizing token flows using interactive performance spectra. In: Janicki, R., Sidorova, N., Chatain, T. (eds.) PETRI NETS 2020. LNCS, vol. 12152, pp. 369–380. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-51831-8\\_18](https://doi.org/10.1007/978-3-030-51831-8_18)
2. Burke, A., Leemans, S.J.J., Wynn, M.T.: Stochastic process discovery by weight estimation. In: Leemans, S., Leopold, H. (eds.) ICPM 2020. LNBIP, vol. 406, pp. 260–272. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-72693-5\\_20](https://doi.org/10.1007/978-3-030-72693-5_20)
3. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-99414-7>
4. Chapela-Campa, D., Mucientes, M., Lama, M.: Simplification of complex process models by abstracting infrequent behaviour. In: Yangui, S., Bouassida Rodriguez, I., Drira, K., Tari, Z. (eds.) ICSOC 2019. LNCS, vol. 11895, pp. 415–430. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-33702-5\\_32](https://doi.org/10.1007/978-3-030-33702-5_32)
5. Denisov, V., Belkina, E., Fahland, D., van der Aalst, W.M.P.: The performance spectrum miner: visual analytics for fine-grained performance analysis of processes. In: International Conference on Business Process Management (Dissertation/Demos/Industry), vol. 2196 (2018)
6. Denisov, V., Fahland, D., van der Aalst, W.M.P.: Predictive performance monitoring of material handling systems using the performance spectrum. In: International Conference on Process Mining (2019)
7. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM<sup>2</sup>: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-19069-3\\_19](https://doi.org/10.1007/978-3-319-19069-3_19)
8. Ge, J., Sigsgaard, K.W., Mortensen, N.H., Hansen, K.B., Agergaard, J.K.: Structured process mining in maintenance performance analysis: a case study in the offshore oil and gas industry. In: International Symposium on System Security, Safety, and Reliability (2023)
9. Van Houdt, G., Depaire, B., Martin, N.: Root cause analysis in process mining with probabilistic temporal logic. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 73–84. Springer, Cham (2022). [https://doi.org/10.1007/978-3-030-98581-3\\_6](https://doi.org/10.1007/978-3-030-98581-3_6)
10. de Leoni, M., Maggi, F.M., van der Aalst, W.M.P.: Aligning event logs and declarative process models for conformance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 82–97. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-32885-5\\_6](https://doi.org/10.1007/978-3-642-32885-5_6)
11. Leung, C.S.K., Lau, H.Y.K.: Simulation-based optimization for material handling systems in manufacturing and distribution industries. *Wirel. Netw.* **26**(7), 4839–4860 (2020)
12. Mansouri, T., Moghadam, M.R.S., Monshizadeh, F., Zareravasan, A.: IOT data quality issues and potential solutions: a literature review. *Comput. J.* **66**(3), 615–625 (2023)
13. Rudnitckaia, J., Venkatachalam, H.S., Essmann, R., Hruska, T., Colombo, A.W.: Screening process mining and value stream techniques on industrial manufacturing processes: process modelling and bottleneck analysis. *IEEE Access* **10**, 24203–24214 (2022)



14. Sommers, D., Menkovski, V., Fahland, D.: Process discovery using graph neural networks. In: International Conference on Process Mining (2021)
15. Tang, J., Liu, Y., Lin, K., Li, L.: Process bottlenecks identification and its root cause analysis using fusion-based clustering and knowledge graph. *Adv. Eng. Inform.* **55**, 101862 (2023)
16. Unger, A.J., dos Santos Neto, J.F., Fantinato, M., Peres, S.M., Trecenti, J., Hirota, R.: Process mining-enabled jurimetrics: analysis of a Brazilian court's judicial performance in the business law processing. In: International Conference for Artificial Intelligence and Law (2021)
17. Verbeek, E., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Prom 6: the process mining toolkit. In: International Conference on Business Process Management, vol. 615 (2010)
18. Yasmin, F.A., Bukhsh, F.A., de Alencar Silva, P.: Process enhancement in process mining: a literature review. In: CEUR Workshop Proceedings, vol. 2270 (2018)