The present work was submitted to:

Chair of Process and Data Science
RWTH Aachen University

Combining Process Mining and Inventory Management by using Quantity Event Log (QEL) to Identify and Visualize Key Metrics

Bachelor Thesis

presented by

Ahmad Noureddin Mannoun
437528

First Examiner: Prof. Dr. Wil M. P. van der Aalst
Second Examiner: Prof. Sander J. J. Leemans

Advisor: Nina Graves

Aachen, March 31, 2025

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Nina Graves, for her valuable guidance and support throughout this thesis. Her insightful feedback and encouragement have been instrumental in shaping my work. I would also like to thank my family and my girlfriend for their support, especially during challenging times. Their patience and motivation have been invaluable throughout this journey.

Abstract

In the domain of inventory management, particularly within logistics, organizations face the challenge of maintaining optimal stock levels while minimizing hidden costs. This thesis addresses the problem by making use of Quantity Event Logs to derive key inventory metrics like lead time, demand, and service level. A systematic meta-review and methodical derivation process define the data requirements and computation of these metrics, which are then implemented in the DISQVER web application to enable actionable, data-driven decision-making. Evaluation on a QEL data demonstrates that this approach effectively uncovers process inefficiencies and enhances inventory control strategies.

Contents

List of Figures

List of Tables

1 Introduction

The American Production and Inventory Control Society describes inventory management as a field of business management focused on the planning and regulation of stock levels. Its primary objective is to ensure that specific products or items are maintained at an appropriate quantity. Effective inventory management systems should account for the product, the customer, and the processes involved in making the product accessible. While the expenses associated with managing inventory throughout the process are not immediately visible, they ultimately contribute to the overall product cost [1]. Inventory management basically serves two main goals [2]. Primarily, effective inventory management ensures the availability of goods, which is crucial for maintaining smooth operational processes. It is essential to have the necessary materials available in the appropriate quantities, of the required quality, and at the right time to meet a defined service level. A secondary objective is to maintain this service level while minimizing costs. Since it is not feasible to stock all items at any cost, strategic decisions must be made to balance inventory availability and cost efficiency. [3].

Process Mining (PM) is recognized as a valuable tool for analyzing and improving complex processes based on event logs [4]. Nevertheless, the utilisation of PM within the domain of logistics remains limited. [5]. The main reason for this discrepancy lies in the data availability [6]. Logistics processes do not necessarily fulfil all requirements data needed. fortunately the concept of Quantity Event Log (QEL) has been introduced. As an extension of Object-centric Event Log (OCEL) enabling the identification of dependencies between the execution of activities, sub-processes, and item levels without needing identifiers for all items [6]. Current process mining methods typically rely on two main assumptions. First, they assume that the control flow is determined solely by the current state of identifiable objects. Second, they consider activities and object types as the key entities, with each being represented by uniquely identifiable events and objects in the event log. However, these assumptions do not apply to inventory management processes (IMPs), where the execution of an event can also be influenced by quantities of products that lack unique identifiers (For instance, it is not logical to assign an ID to every paperclip or rubber band.) [6].

In this work, we provide 1) a set of key metrics relevant for inventory management processes, and 2) a method for calculating and visualizing these key metrics. To achieve this, the DISQVER web application is implemented, enabling the application of Object-Centric Process Mining (OCPM) in inventory management. This approach facilitates data-driven decision-making based on the analyses generated by the tool.

Figure 1.1 illustrates the overall methodology and workflow of our approach. The figure highlights how domain knowledge and the QEL is leveraged to calculate and present key metrics namely, lead time, demand, and service level which are critical for effective inventory management.

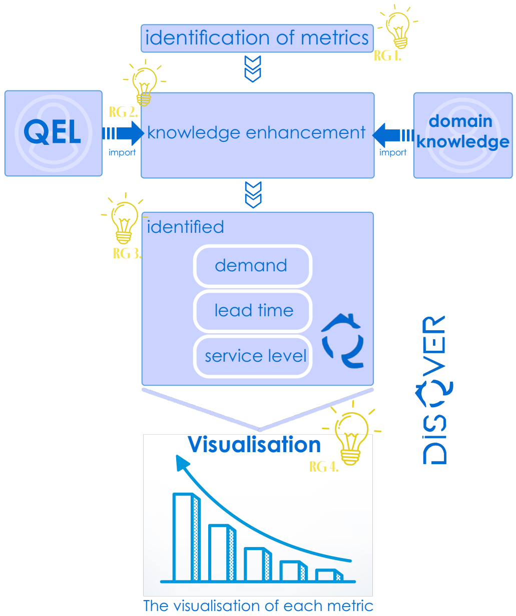


Figure 1.1: Overview of the methodology: integrating QEL, domain knowledge, key metric identification (lead time, demand, service level), and targeted visualization to enhance decision-making.

1.1 Motivation

It has been shown that inventory is often the biggest hidden cost in a business. It is unwise to assume that the cost of inventory will be absorbed by any party. Those who do not pass on the cost of inventory risk going out of business. [1]. Therefore, making informed and data-driven decisions in inventory management is critical to maintaining both profitability and operational efficiency. The introduction of the Quantity Event Log addresses a gap by extending existing process mining methods to incorporate quantity-based dependencies. Using QEL, it becomes possible to identify relationships between aggregated counts of items, subprocesses, and activities without requiring unique identifiers for every item. Our proposed method leverages QEL to derive key metrics for inventory management. By visualizing these key metrics, decision-makers gain actionable insights into their inventory processes.

1.2 Problem Statement

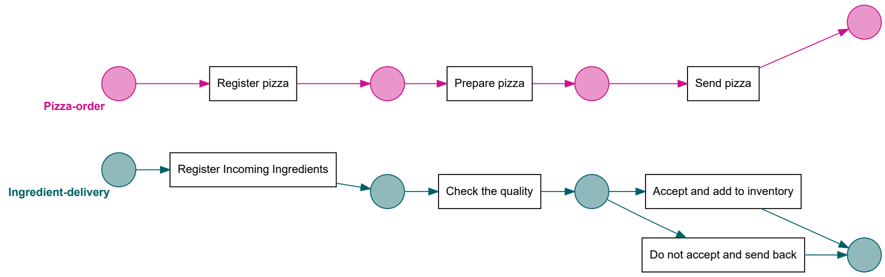


Figure 1.2: An OCPN discovered from an OCEL without quantity operations.

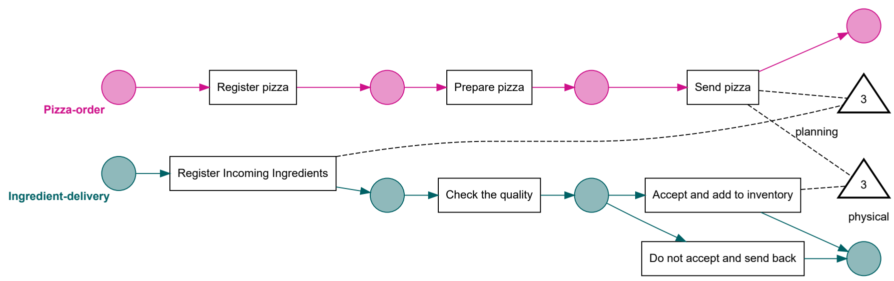


Figure 1.3: A quantity net derived from the same OCEL with integrated quantity operations.

Figure 1.4: The example illustrates a quantity-dependent process composed of two independent subprocesses.

Inventory management in logistics is a complex domain that does not fit well with traditional process mining methods [6]. As seen in Figure 1.4, the OCPN (Figure 1.2) reveals two disconnected subprocesses, which, when considered in isolation, do not adequately represent the overall process—for example, a pizza cannot be made without its ingredients. These subprocesses are inherently connected through the flow of items into and out of collection points.

Traditional process mining techniques are limited to showing these independent subprocesses, as illustrated in Figure 1.2. In contrast, the quantity net in Figure 1.3

demonstrates the connection between the two subprocesses by incorporating quantity operations into the event log. This enriched representation provides deeper insights into the flow and transformation of inventory.

Despite the potential benefits, quantity event logs have not yet been fully exploited. Furthermore, typical process mining techniques struggle to identify standard inventory levels or key performance metrics [6]. Given that logistics is also one of the least explored areas in process mining [5], there remains significant untapped potential for making smarter, data-driven decisions that can improve inventory management.

1.3 Research Questions and Goals

To address the problem of improving inventory management in logistics using process mining and Quantity event logs , the work is structured around the following research questions (RQs)

RQ1 What are the key Metrics relevant to inventory management?

RQ2 What information is needed to fully support these Key Metrics, and how should this data be structured in an event log?

RQ3 How can key inventory management metrics be derived from a Quantity Event Log?

RQ4 How can these metrics be effectively visualized to support data-driven decision-making in inventory management?

To answer these research questions, the following research goals (RGs) are defined:

RG1 Identify the key metrics that are critical for effective inventory management in logistics.

RG2 Determine the necessary data requirements and structure for event logs to fully support these key metrics.

RG3 Develop and implement a method to extract and calculate key inventory management metrics from QELs.

RG4 Implement a web application for visualizing key metrics in inventory management.

1.4 Contributions

This thesis makes several contributions to the field of inventory management, addressing both scientific and practical aspects. In achieving the research goals (RG1–RG4), the following contributions have been made:

1. **Scientific Contributions:**

- A systematic meta-review of literature reviews and studies on multi-echelon systems and inventory management. This review identified the key metrics critical for effective inventory management, thereby addressing RQ1.
- A detailed specification of the data requirements and event log structure necessary to support these key metrics. This contribution establishes a solid foundation for structuring QELs to facilitate metric derivation, addressing RQ2.
- A methodological framework for deriving key inventory management metrics from QELs. This framework includes the calculation of demand, lead time, and service level measures, which directly addresses RQ3.

2. **Implementation Contributions:**
   - The development and implementation of the DISQVER application, a web-based tool that visualizes key inventory management metrics. This tool supports data-driven decision-making by providing interactive visualizations for demand, lead time, and service level analyses, thereby addressing RQ4.

1.5 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2 Preliminaries**: Presents the fundamental mathematical concepts and background information that underpin the analysis throughout the thesis.
- **Chapter 3 Related Work**: Reviews existing literature on inventory management analysis using process mining.
- **Chapter 4 Key Metrics Identification**: Identifies and discusses the critical metrics for effective inventory management.
- **Chapter 5 Derivation of Relevant Metrics from a QEL**: Describes the methodology for calculating key inventory management metrics from Quantity Event Logs.
- **Chapter 6 Implementation**: Details the design and development of the DISQVER application.
- **Chapter 7 Evaluation**: Presents an evaluation of both the derived metrics and the DISQVER application.
- **Chapter 8 Discussion**: Discusses the benefits and limitations of the proposed methods and application.
- **Chapter 9 Conclusion**: Summarizes the contributions of the thesis and outlines directions for future research.

2 Preliminaries

In this chapter, we introduce terms and concepts from both domains, object-centric process mining and inventory management, which we refer to in the remaining work.

2.1 Mathematical Foundations

I. $\mathbb{N} = \{1, 2, 3, \dots\}$ denotes the set of *natural numbers*, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ the set of *natural numbers including zero*, $\mathbb{Z}$ the set of *integers* and $\mathbb{R}$ the set of *real numbers. Infinity* is denoted by ∞ and is not contained in any of the aforementioned sets.The cardinality of a set A is denoted as |A|.

II. For a set X, $\mathcal{P}(X)$ denotes the *powerset* of X, i.e., the set of subsets of X. And $\mathcal{B}(X)$ the set of all multisets over X. A multiset is an ordered pair (X, m) where X is a set and m is a function that assigns to each element of X a non-negative integer, called its *multiplicity.* For example, let $X = \{a, b, c\}$ and $y : X \to \mathbb{N}_0$ assigns a natural number to every element of a set, then $y_1 = [a^2]$ and $y_2 = [a, b^3]$ are multisets of X and $y_1, y_2 \in \mathcal{B}(X)$

III. A *total function* $f \colon X \to Y$ assigns a value $f(x) \in Y$ to each $x \in X$. A *partial function* $f \colon X \nrightarrow Y$ does not necessarily map all $x \in X$ to a value in Y. For those $x \in X$ where $f(x)$ is undefined, we write $f(x) = \perp$.

For a function or partial function $f \colon X \to Y$, the *range* of f is given by $\mathrm{rng}(f) = \{f(x) \mid x \in X\} \setminus \{\perp\}$. The *domain* of f is denoted by $\mathrm{dom}(f) = \{x \in X \mid f(y) \neq \perp\}$.

2.2 Quantity Event Logs

In this section, we introduce the concept of object-centric event logs as well as the concept of quantity event logs which are an extension to an OCEL and will be used to identify inventory management metrics.

To define object-centric event logs and quantity event logs, we first need to give some preliminary definitions:

Definition 2.2.1. (Universes). Let $\mathbb{U}_\Sigma$ be the universe of of strings. The following universes are defined as pairwise disjoint sets:

- $\mathbb{U}_{ev} \subseteq \mathbb{U}_\Sigma$ is the universe of events,
- $\mathbb{U}_{etype} \subseteq \mathbb{U}_\Sigma$ is the universe of event types,

- $\mathbb{U}_{obj} \subseteq \mathbb{U}_\Sigma$ is the universe of objects,
- $\mathbb{U}_{otype} \subseteq \mathbb{U}_\Sigma$ is the universe of object types,
- $\mathbb{U}_{attr} \subseteq \mathbb{U}_\Sigma$ is the universe of attribute names,
- $\mathbb{U}_{val} \subseteq \mathbb{U}_\Sigma$ is the universe of attribute values,
- $\mathbb{U}_{time} \subseteq \mathbb{U}_\Sigma$ is the universe of timestamps, including $0, \infty \in \mathbb{U}_{time}$, totally ordered with $0 \leq t \leq \infty$ for all $t \in \mathbb{U}_{time}$.
- $\mathbb{U}_{qual} \subseteq \mathbb{U}_\Sigma$ is the universe of qualifiers.
- $\mathbb{U}_{it} \subseteq \mathbb{U}_\Sigma$ is the universe of item types.
- $\mathbb{U}_{cp} \subseteq \mathbb{U}_\Sigma$ is the universe of Collection points.

Definition 2.2.2 (Counter, Item quantity)**.** A counter is a function that maps each item type to an item quantity $c : I \to \mathbb{Z}$ where I is the set of item types $I \subseteq \mathbb{U}_{it}$. We denote the set of collection counters over a set of item types $I \subseteq \mathbb{U}_{it}$ as $\mathcal{I}(I) = I \to \mathbb{Z}$. For any two counters $c_1, c_2 \in \mathbb{Z}$, the sum of the counters $c_1 \oplus c_2$ is defined as $(c_1 \oplus c_2)(it) = c_1(it) + c_2(it)$. Two counters $c_1, c_2 \in \mathcal{I}(I)$ are considered equal, $c_1 = c_2$ iff $c_1(it) = c_2(it)$ Similar to [6], we write $set^-(c) = \{i \mid c(i) \leq 0\}$ to refer to the set of all item types a counter $c \in \mathcal{I}(I)$ assigns a negative quantity to. Given a counter $c \in \mathcal{I}(I)$, we denote the absolute counter $abs(c) \in \mathcal{I}(I)$ where $abs(c)(it) = |c(it)|$ for every item type $it \in I$.

For example let $I = \{a, b, c\}$,$c_1 = [a^2, b]$, c_2 =$[a, b^{-2}, c^3]$ and $c_3 = [a^2, b]$ then: $c_2 \oplus c_3$ = $c_2(it) + c_3(it) = [a^3, b^{-1}, c^3]$ and $c_1 = c_3$ since the item count is equal for all items.

Definition 2.2.3 (Collection Counter)**.** A collection counter is a function that maps each collection point to a counter $cc : CP \to \mathcal{P}(I)$ and $C(CP, I) = CP \to \mathcal{I}(I)$ where CP is the set of collection points $CP \subseteq \mathbb{U}_{cp}$ and $\mathcal{P}(I)$ is the set of all possible counters over I. For any two collection counters cc_1, cc_2, the sum of the collection counters $cc_1 \oplus cc_2$ is defined as $(cc_1 \oplus cc_2)(cp) = cc_1(cp) \oplus cc_2(cp)$. And the equality is defined as $(cc_1 = cc_2)(cp) \leftrightarrow cc_1(cp) = cc_2(cp)$. Given a collection counter $cc : CP \to \mathcal{I}(I)$, we denote the projection of the collection counter to only the negative item quantities as cc^-, where for all $cp \in CP$: $cc^-(cp) = cc(cp)\restriction_{set^-(cc(cp))}$.

For example let $CP = \{cp_1\}$,item types $I = \{a, b, c\}$, $cc_1 = (cp_1$,$[a, b^{-2}, c^3])$, $cc_2 = (cp_1$,$[a^2, b])$ and cc_1 =$(cp_1$,$[a, b^{-2}, c^3])$ then: $cc_1 \oplus cc_2 = cc_1(cp_1) + cc_2(cp_1) = [a^3, b^{-1}, c^3]$ and $cc_1 = cc_3$. The projection cc_1^- keeps only the items with negative quantities. In this case, the only item with a negative exponent is b^{-2}. Thus, the projected collection counter is: $cc_1^-(cp_1) = [b^{-2}]$

Based on these preliminary definitions, we can now define an OCEL and a QEL. There for we will use the definition from [7] and [6]:

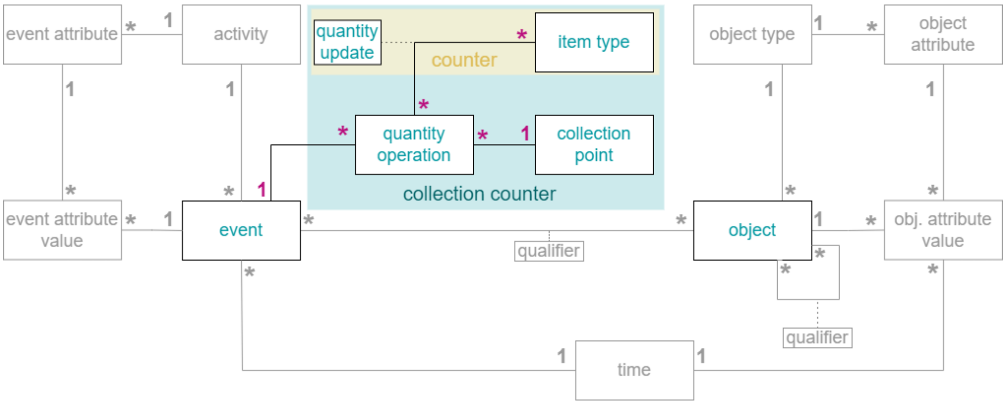


Figure 2.1: OCEL and QEL Metamodel[6]

Definition 2.2.4 (Object-Centric Event Log)**.** An Object-Centric Event Log (OCEL) is a tuple

$$L = (E, O, EA, OA, evtype, time, objtype, eatype, oatype, eaval, oaval, E2O, O2O)$$

where

- $E \subseteq \mathbb{U}_{ev}$ is the set of events,
- $O \subseteq \mathbb{U}_{obj}$ is the set of objects,
- $evtype \colon E \to \mathbb{U}_{etype}$ assigns event types to events,
- $time \colon E \to \mathbb{U}_{time}$ assigns timestamps to events,
- $objtype \colon O \to \mathbb{U}_{otype}$ assigns object types to objects,
- $EA \subseteq \mathbb{U}_{attr}$ is the set of event attributes,
- $OA \subseteq \mathbb{U}_{attr}$ is the set of object attributes,
- $eatype \colon EA \to \mathbb{U}_{etype}$ assigns event types to event attributes,
- $oatype \colon OA \to \mathbb{U}_{otype}$ assigns object types to object attributes,
- $eaval \colon (E \times EA) \nrightarrow \mathbb{U}_{val}$ assigns values to event attributes,
- $oaval \colon (O \times OA \times \mathbb{U}_{time}) \nrightarrow \mathbb{U}_{val}$ assigns values to object attributes,
- $E2O \subseteq E \times \mathbb{U}_{qual} \times O$ are the qualified event-to-object relations,
- $O2O \subseteq O \times \mathbb{U}_{qual} \times O$ are the qualified object-to-object relations.

We introduce also the following notations given an OCEL L:

- $A(L) = \{evtype(e)|e \in E\}$ is the set of activities
- $OT(L) = \{objtype(o)|o \in O\}$ is the set of object types

Definition 2.2.5 (Quantity Event Log)**.** A Quantity Event Log (QEL) is a tuple

$$QEL = (OCEL, I, CP, eqty, \prec_e)$$

where

- $OCEL$ is an OCEL,
- $I \subseteq \mathbb{U}_{it}$ is the set of item types,
- $CP \subseteq \mathbb{U}_{cp}$ is the set of collection points,
- $eqty : E \to C(CP, I)$ a mapping of events to quantity operations for all collection points,
- $\prec_e \subseteq (E \times E)$ is a total order on events.

Also we need to denote $eqty(e) = eqty(e^{'})$ if and only if $C(CP, I) = C^{'}(CP^{'}, I^{'})$ for any $e, e^{'} \in E$.

Definition 2.2.6 (Quantity Net)**.** A Quantity Net is a tuple

$$QN = (OCPN, CP, QA)$$

where

- $OCPN$ is a Object-Centric Petri Net,
- CP is a set of collection points,
- QA is a set of undirected arcs between collection points and events.

To further understand these concepts, provide a basis for chapter 5 and make the methods we propose more understandable, we introduce an example QEL and the data it contains.

2.3 Example QEL

We present a simple process, illustrated in Figure 2.2, to demonstrate the concept of a QEL. The process involves two object types Ingredient-delivery and Pizza-order and seven activities: Register Incoming Ingredients, Check the Quality, Accept and Add to Inventory, Do Not Accept and Send Back, Register Pizza, Prepare Pizza, and Send Pizza. Additionally, there are two collection points: Planning and Physical.

Table 2.1: Data from Event

ocel_id	ocel_type	ocel_time
ev_1	Register Incoming Ingredients	2025-01-01 00:00:01
ev_2	Register Incoming Ingredients	2025-01-01 00:00:10
ev_3	Check the quality	2025-01-01 00:00:03
ev_4	Check the quality	2025-01-01 00:00:14
ev_5	Accept and add to inventory	2025-01-01 00:00:05
ev_6	Do not accept and send back	2025-01-01 00:00:16
ev_7	Register pizza	2025-01-01 00:00:07
ev_8	Register pizza	2025-01-01 00:00:18
ev_9	Prepare pizza	2025-02-01 00:00:07
ev_10	Prepare pizza	2025-03-01 00:00:18
ev_11	Send pizza	2025-01-01 00:25:07
ev_12	Send pizza	2025-01-01 00:30:18
ev_13	Register pizza	2025-02-01 00:00:09
ev_14	Register pizza	2025-03-01 00:00:20
ev_15	Prepare pizza	2025-01-01 00:00:11
ev_16	Prepare pizza	2025-01-01 00:00:22
ev_17	Send pizza	2025-02-01 00:20:09
ev_18	Send pizza	2025-03-01 00:40:20

Table 2.2: Data from Object

ocel_id	ocel_type
o_1	Ingredient-delivery
o_2	Ingredient-delivery
o_3	Pizza-order
o_4	Pizza-order
o_5	Pizza-order
o_6	Pizza-order

Table 2.3: Data from event_object

ocel_event_id	ocel_object_id	ocel_qualifier
ev_1	o_1	None
ev_3	o_1	None
ev_5	o_1	None
ev_2	o_2	None
ev_4	o_2	None
ev_6	o_2	None
ev_7	o_3	None
ev_9	o_3	None
ev_11	o_3	None
ev_8	o_4	None
ev_10	o_4	None
ev_12	o_4	None
ev_13	o_5	None
ev_15	o_5	None
ev_17	o_5	None
ev_14	o_6	None
ev_16	o_6	None
ev_18	o_6	None

Table 2.4: Data from Quantity Operations

ocel_id	ocel_cpid	Dough	Tomato_sauce	Cheese
init	planning	None	None	None
init	physical	10	10	10
ev_1	planning	10	10	10
ev_2	planning	5	5	5
ev_5	physical	10	10	10
ev_11	physical	-1	-2	-3
ev_11	planning	-1	-2	-3
ev_12	physical	-1	-2	-3
ev_12	planning	-1	-2	-3
ev_17	physical	-3	-7	-4
ev_17	planning	-3	-7	-6
ev_18	physical	-2	-3	-3
ev_18	planning	-2	-3	-3

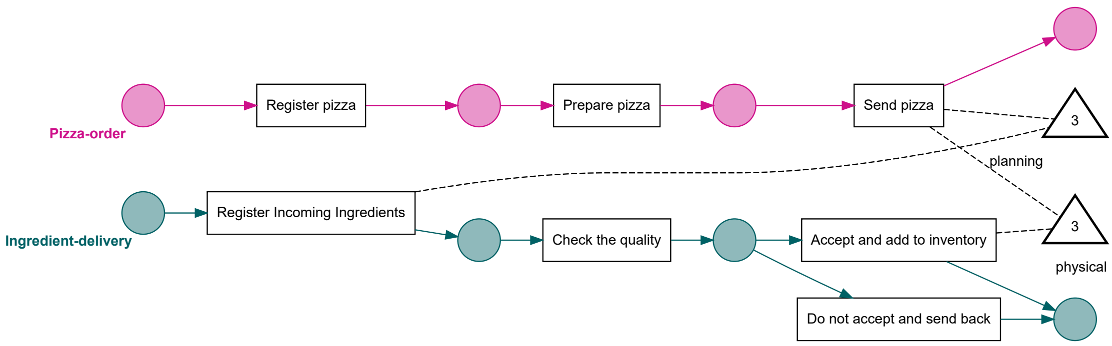


Figure 2.2: Example Quantity Event Log

Consider a small pizza store that purchases ingredients and sells pizzas. Whenever ingredients are needed, a replenishment order is placed. Upon arrival, the ingredients undergo a quality check. If they meet the required standards, they are accepted and added to inventory; otherwise, they are rejected and returned. When a customer places an order, the store registers the order, prepares the pizza, and then delivers it to the customer.

The quantity net in Figure 2.2 was discovered from the QEL data that is presented in the Tables [2.1,2.2,2.3,2.4] .

2.4 Inventory Management performance indicators (Metrics)

In this section, we introduce the concept of inventory management performance indicators, which are used to evaluate the performance of inventory management processes. Logistics performance is typically assessed using both quantitative (hard) and qualitative (soft) metrics. Quantitative metrics focus on measurable factors such as order cycle time, fill rates, and costs. In contrast, qualitative metrics capture aspects like managerial assessments of customer satisfaction and loyalty. While quantitative indicators can be calculated using straightforward mathematical formulas, evaluating qualitative metrics often requires more advanced analytical techniques.[8]

3 Related Work

This chapter discusses related work in the field that combines process mining and inventory and logistics management.

Kretzschmann et al.[9] introduce an object centric data model (OCDM) that integrate and standardize data from multiple information systems (e.g., demand forecasting, ERP systems, and document metadata sources). This model abstracts the complexity inherent in traditional inventory processes (Order-to-Cash and Purchase-to-Pay) and forms a unified view of the data. It is also allow the computation of key inventory-related metrics such as Economic Order Quantity, Reorder Point, Safety Stock, Maximum Stock, and Overstock. From this integrated data, a single object-centric event log (OCEL A) is extracted. By enriching the activities from (OCEL A) with the existing metrics a more informative log (OCEL B) that allows for an event-driven analysis of process behaviors and inefficiencies. With (OCEL B) in hand, the authors apply process mining techniques to systematically analyze process behaviors. This analysis focuses on identifying root causes for common inventory issues like understock and overstock.

Knoll et al.[10] address the limitations of traditional, pen-and-paper value stream mapping (VSM) in complex, dynamic manufacturing environments. They propose a novel methodology that combines multidimensional process mining (MDPM) with established lean production principles and VSM to support continuous improvement in internal logistics for mixed-model assembly lines. After automatically mapping raw transfer orders from the warehouse management system to physical logistics activities, the orders are transformed into a standardized event logs. then this event log is filtered and enriched using the Extensible Event Stream (XES) standard. Additional information from the transfer orders and activity model is added. The authors then apply MDPM to analyze the logs, by first discovering a process models to analysis the current state maps created via VSM. then a performance analysis is conducted calculating performance metrics and lastly a conformance analysis is done.

Focusing on the Purchase-to-Pay (P2P) process—a common business process dealing with material flows and services—[11] investigates a real-life P2P process through a three-part analysis. First, they utilize Microsoft PowerBI to gain valuable insights and effectively assess the performance of the P2P process by calculating performance metrics (e.g., average quantity per purchase order). Secondly, they employ Celonis to perform a process-centric analysis, visualizing actual flows, identifying deviations, and measuring throughput times across different process stages (e.g., from purchase order creation to the arrival of the order). Lastly, they conduct a predictive analysis using machine learning to predict process throughput times and identify undesired activities that cause significant delays.

In a different study, [12] also focuses on the P2P process, but with a distinct approach using object-centric process mining. After extracting data from an SAP ERP system and creating the OCEL, the event log is preprocessed and then subjected to two types of analysis. First, graph-based analyses are performed, during which an ”object creation graph” is discovered and examined. This graph represents long-term dependencies between different objects, analyzes the duration between purchase requisitions and orders, and identifies orders with more than one arrival. Second, statistical analyses are conducted to determine activities correlated with high processing times, and SQL queries are executed to evaluate processing capacity, invoice processing time, and the number of ”no-touch” orders.

Terlouw[13] propose a hybrid methodology that first applies conventional process mining techniques to uncover the underlying structure of logistics processes. The initial discovery using the inductive mining algorithm provides insights into activity frequencies, dependencies, and timing aspects. The approach further incorporates principles from Lean Six Sigma to determine optimal inventory levels. Using a continuous review model, the authors calculate the inventory reorder point and the optimal order quantity thereby bridging process mining insights with practical inventory control.

In [14] the Open Trip Model (OTM) (an open-source data sharing standard used in logistics) to overcome challenges related to interoperability, data loss, and quality when aggregating event data from diverse systems. that by using the OTM data model with a standard event log structure (not an OCEL), the basic requirements for traditional process mining can be met. After creating an event log from the mapped data, the authors use the process mining tool Disco to import the log, generate a process model, and extract performance metrics (e.g., activity frequencies, process variants, and durations). This step demonstrates that the OTM can serve as a viable foundation for building process mining applications in the logistics domain.

Looking at all of these approaches—whether using OCPM or traditional process mining—they all rely on data from specific activities, incorporate these into existing activities, or create new activities to represent the desired outcomes. Moreover, they typically focus solely on item or order movements, without considering the overall situation of stocks. In contrast, the concept of the QEL incorporates collection points, which not only capture item movements but also represent the state of stocks. In our approach, we leverage these collection points to calculate performance metrics, thereby providing a more comprehensive view of inventory dynamics.

4 Key Metrics Identification

In this chapter, we identify the key metrics and domain knowledge necessary to address inventory management challenges in logistics. To answer RQ1 (What are the key Metrics relevant to inventory management?) and RQ2 (What information is needed to fully support these Key Metrics, and how should this data be structured in an event log?), we conducted a meta-review on a set of literature reviews and some Studies focusing on the themes of multi-echelon systems and inventory management. This chapter summarizes the findings from these studies, highlights their similarities, and derives the essential knowledge needed.

4.1 Search Method

To identify the key metrics and domain knowledge necessary for addressing inventory management challenges, a systematic literature search was conducted.

A meta-review was performed on existing literature reviews and studies focusing on multi-echelon systems and inventory management in logistics. This search aimed to capture a broad spectrum of perspectives and methodologies. The following academic databases were used:

- Scopus
- Web of Science
- Google Scholar

A combination of keywords was employed, including:

- *"inventory management"*,
- *"multi-echelon inventory models"*,
- *"performance measurement in logistics"* and,
- *"warehouse performance"*.

This search allowed us to identify:

- Performance metrics commonly used in logistics,
- The components of existing inventory management systems,

- Key performance indicators used to evaluate these systems, and
- The data needed to support these metrics.

which is exactly the objective of our search.

4.2 Overview of the Reviewed Literature

Warehouse performance measurement: a literature review

This literature review examines how warehouses are assessed in terms of operational performance. As supply chains grow increasingly complex, warehouse managers face the challenge of selecting appropriate performance metrics—such as those related to time, cost, quality, and productivity—given the absence of a universally accepted set of measures. This review addresses this issue by systematically analyzing prior research and identifying key metrics commonly used in warehouse evaluations. The authors synthesized a gruppe of performance metrics from a selection of studies. These metrics were categorized into two main groups: direct indicators, which measure concrete outcomes such as processing speed or order accuracy, and indirect indicators, which influence performance more broadly, such as resource utilization. Additionally, the study introduces a framework that delineates the scope of these indicators, clarifying their definitions and their relevance to various aspects of warehouse operations. Overall, the findings reveal a wide range of performance metrics used in the literature, yet no clear consensus on which are most effective, posing a challenge for managers in selecting the most suitable measures[8].

A Typology and Literature Review on Stochastic Multi-Echelon Inventory Models

This literature review looked into the development of stochastic multi-echelon inventory models, a field that has been evolving for more than 50 years. this field is examined in this review of the literature. From early theoretical models to more useful optimization techniques applied in actual supply chains, it identifies important research aspects. Additionally, it lists the kinds of metercs that have been studied along with their primary accomplishments. The authors used a systematic approach to organize the review. They started by compiling citations from important review papers and handbooks[15].

A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization

Here is the guaranteed-service models (GSM) is being explored for optimizing multi-echelon inventory systems. Although multi-echelon inventory management has been widely researched, much of the focus has been on stochastic-service models (SSM), creating a noticeable gap in the literature regarding GSM. this paper addresses that gap by giving a thorough overview of existing GSM models, the performance maguaris they use and the methods used to solve them. To put this review together, the authors carried out a systematic search across databases like Web of Science, Google Scholar, and ScienceDirect[16].

A Review and Reflection on Inventory Management of Perishable Products in a Singleechelon Model

This paper dives into managing inventory for perishable goods, particularly using single-echelon continuous review models. It makes the point that conventional inventory models, which frequently depend on profit or total cost measures, don't necessarily capture the complexity of supply chains in the real world, particularly when maximizing system performance is the aim. The study recommends switching to a multi-metric strategy that takes into account more variables than only financial ones. In today's dynamic and fast-paced business environment, it argues for a more complete approach to managing perishable inventory by examining the limitations of single-metric solutions[17].

Logistics-oriented inventory analysis

The paper explores how modern businesses can optimize their inventory management to meet increasingly demanding customer expectations while keeping costs under control. The authors present the idea of inventory operating curves as a useful analytical tool because they understand that success nowadays depends not only on operational effectiveness and product quality but also on important logistical metrics like lead time, service level, and delivery reliability. Businesses can use these curves to calculate the optimal inventory levels required to meet service performance goals. Even companies without complex IT infrastructures can utilize the strategy because it makes use of data that is normally available from ordinary ERP or warehouse management systems[18].

Performance Measurement on Inventory Management and Logistics Through Various Forecasting Techniques

This study looks into how manufacturing efficiency can be increased through effective inventory management and logistics assessment. In order to handle stochastic demand and avoid stock-outs, it uses a variety of forecasting approaches in addition to economic order amount, reorder point, and safety stock calculations. It also uses an advanced categorization system to categorize inventory products into fast, slow, and non-moving groups. Furthermore, it assesses logistics performance, particularly on-time delivery and truck capacity utilization, and discovers that vehicles are running at slightly more than 50% of their potential. Recommendations for maintaining suitable safety stocks and enhancing packing techniques are included in the paper's conclusion to improve supply chain performance overall[19].

The studies in this section addressed a range of inventory management systems across various industries and modeling approaches. By synthesizing their findings, we identified key common metrics that are consistently relevant across different inventory models. These shared aspects directly answer RG1, as they provide a structured understanding of the most critical metrics in inventory management.

Furthermore, since some of the reviewed papers also discuss how these metrics are measured, tracked, and applied in decision-making, their insights help define the necessary data structures and event log attributes needed to support these metrics. This directly contributes to answering RG2, ensuring that the event log aligns with real-world inventory management requirements.

4.3 Findings from the Literature

In this section, we summarize the findings from the literature regarding key inventory management metrics. The analysis is based on several studies, and the results are organized into two tables. The table 4.1 provides an overview on which metrics are covered in each study, while the table 4.2 lists the key metrics identified in the literature as well as an explantion and the necessary data to support them.

Table 4.1: Overview of Literature Findings on Key Inventory Management Metrics. **Legend:** Green cells indicate that the literature source addresses the corresponding metric; red cells indicate that the metric is not covered.

	Metrics															
Literature	E[1]	S[2]	ToA[3]	I[4]	C[5]	LT[6]	D[7]	RtS[8]	RP[9]	LS[10]	SR[11]	IC[12]	TP[13]	TO[14]	FR[15]	RP[16]
[8]	Red	Red	Red	Red	Green	Green	Green	Green	Green	Red	Green	Green	Green	Green	Green	Green
[15]	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red	Red
[16]	Green	Red	Red	Red	Green	Green	Green	Red	Green	Red	Green	Red	Red	Red	Red	Red
[17]	Green	Green	Red	Red	Red	Green	Green	Green	Green	Red	Green	Red	Red	Red	Green	Red
[18]	Red	Red	Red	Red	Red	Green	Green	Red	Red	Green	Green	Red	Green	Red	Green	Red
[19]	Red	Red	Red	Red	Green	Green	Green	Red	Red	Red	Green	Green	Red	Green	Red	Green

E = Echelons, 2) S = Structure, 3) ToA = Timing of Activities, 4) I = Information, 5) C = Capacity, 6) LT = Lead Time, 7) D = Demand, 8) RtS = Reactions to Stockout, 9) RP = Replenishment Policy, 10) LS = Lot Size, 11) SR = Service Requirements, 12) IC = Inventory Cost, 13) TP = Throughput, 14) TO = Turnover, 15) FR = Fill Rate, 16) RP = Reorder Point.

4.4 Common Insights and Derivation of Necessary Domain Knowledge

The literature review reveals three particularly important commonalities frequently discussed in the examined studies. These insights are fundamental to designing efficient inventory management systems and guide the derivation of necessary domain knowledge.

Demand

Demand is a key driver in inventory management, varying significantly by market and product. To maintain optimal inventory levels, demand must be carefully analyzed and accurately forecasted. Since each item exhibits its own demand pattern, orders must include detailed information about the items purchased, such as the type and quantity of each item. Moreover, if orders contain retailer or supplier identifiers, it becomes possible to determine demand on a per-retailer or per-supplier basis. From an event log perspective, this necessitates that the order objects, or the activity corresponding with the placement of an order, capture comprehensive item data. If we take a look at the example log from Chapter 2 and consider the *Ingredient-delivery* as an order type, we can see from the provided data that the order object does not have the necessary data, but it has an activity that does, which is *Register Incoming Ingredients.* Take `o_1` for example; `o_1` is mapped to `ev_1` and `ev_1` has the following quantity operation: {*Collection: planning, Dough: 10, Tomato_sauce: 10, Cheese: 10*}.

Lead Time

Delivery time is critical to the efficiency and reliability of an inventory management system. Shorter lead times can reduce inventory levels and enhance service quality. Lead time is applicable to orders, although its interpretation may differ between customer orders and replenishment orders. Additionally, lead time can be associated with specific suppliers in replenishment orders and with retailers or customers in customer orders. Therefore, it is essential that the event log records the timestamps for both order placement and arrival. Customer orders and replenishment orders need to be distinguishable objects. If we consider the example log, we see that we have two distinguishable objects that represent the replenishment order and customer order (namely, *Ingredient-delivery* and *Pizza-order*), each with its own placement and arrival activity.

Service Requirements

Service requirements, such as cost constraints and service level targets, are critical for aligning inventory management practices with business objectives. In the context of service levels, evaluating the fulfillment of customer orders is essential. This involves comparing the quantity ordered with the quantity received. Effective analysis requires that the event log captures both the placed and arrived quantities, which in turn supports the calculation of performance metrics such as alpha and beta service levels. For instance, consider object `o_1`: the placement event is `ev_1` and the arrival event is `ev_5`. In this example, the quantity placed is 30, and the quantity arrived is also 30.

This chapter analyzed key inventory management metrics derived from multi-echelon inventory literature. The findings highlight three fundamental aspects **lead time, demand, and service requirements** which are essential for effective inventory control. So in the next chapter we will take a closer look on RQ3 (How can key inventory management metrics be derived from a QEL?).

Table 4.2: Key Inventory Management Metrics Identified in the Literature

Key Metric	**Explanation**	**Necessary Data**
Echelons*	Number of echelons in the supply chain	–
Structure*	Type of structure (Serial, Convergent, Divergent, etc.)	–
Timing of activities*	Actions possible at any or at specific points in time	–
Information*	Source and level of available information	–
Capacity	Finite and infinite storage capacity	Stock information (e.g., how much goes in and out)
Lead time	Time it takes to deliver an item/order	Timestamps for order placement and delivery
Demand	Demand for an item (Deterministic, Stochastic, Normal, Poisson, etc.)	The quantity of each order
Reactions to stockout	System responses when stock is depleted (Backordering, Guaranteed service, Lost sales)	Order quantities, the canceled orders as well as the orders with multiple deliveries and their timestamps
Replenishment policy	How the stock is being restocked (Base stock, (s, S), (s, S, Q), etc.)	Reorder point, reorder quantities, safety stock, and stock limit
Lot size	Release quantity (Flexible/Fixed)	The quantity of each replenishment
Service requirements	Objectives achieved via control policies	Placement quantities as well as arrival quantities of orders and there cost
Inventory Cost	Total storage and handling costs	Storage cost
Throughput	Graphical representation of an item's inventory history over time	Inventory input, output as well as the demand and inventory level
Turnover ratio	Ratio between sold goods and average inventory	Order quantity, cost per item and the average inventory level
Fill rate	Proportion of orders fulfilled on the first shipment	Orders that did not split into multiple deliveries and the total orders
Reorder Point	Stock level that triggers a new order	Demand data, reorder trigger events

* These metrics were part of an inventory systems, so not much data was presented for them.

5 Derivation of Relevant Metrics from a QEL

After identifying the relevant metrics in the previous chapter, we introduce methods for their determination in this chapter. By doing so, we provide an answer to the third research question (RQ3) and reach research goal (RG3). We will first limit our scope to a subset of metrics as seen in the previous chapter, before considering the overall data required for their identification using a QEL. For each of these metrics we will provide the following:

- A definition of each of these metrics in the context of a QEL.
- A consideration of the additional information required for this metric in particular and the requirements on the event log they pose.
- A method to determine the measure using a QEL and the additionally provided domain knowledge.
- An example of the metric's determination using the example QEL from 2.3.

Due to the large number of metrics identified above, we selected the most relevant ones as part of this thesis' scope:

- Lead Time
- Demand
- Service Level

Considering these measures, we quickly realize that a QEL alone is not sufficient for their determination. Additional information is required regarding both the involved object types and the executed activities.

For all these measures, it is essential to know which object type represents an order, denoted as $ot^{order} \in OT(L)$. In the following sections, we present methods for deriving these measures using a QEL and the additional information provided.

In the following, we assume that the QEL includes one or more object types specifying an order. An order can be a purchase order or a customer order. All metrics introduced below are defined with regard to a selected order object type. We denote this order object type as $ot^{order} \in OT$, with $O^{order} = \left\{ o^{order} \in O \mid objtype(o^{order}) = ot^{order} \right\}$ the set of order objects.

5.1 Demand

As we saw in chapter 4 calculating the demand and forecasting it is a very crucial task. In the review, we saw that for the demand the following aspects are of interest: its distribution and the aggregated demand for a given period of time. In the following, we show how the aggregated demand for a given time period can be determined before adding a requirement to be able to determine the demand for a single order object.

To determine the demand for a period m, we assume that the demand expressed by removals of items from a subset of demand-specifying collection points $CP^{demand} \subseteq CP$. We first identify all events in the corresponding time period, e.g., a month, denoted $E^m \subseteq E$. To determine the total demand per item type for a the period of time m, we must consider the quantity operations of all events to to demand-specifying collection points.

Definition 5.1.1 (Period Demand)**.** Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log, m time period and let $E^m \subseteq E$ be the subset of placement events in m, and $CP^{dem} \subseteq CP$ the subset of demand-specifying collection points. The demand for an item type $it \in I^{demand}$ is determined by adding all item quantities of this item type:

$$d(it, m) = \sum_{e \in E_m^{pl}} \sum_{cp \in CP^{dem}} abs(eqty(e)(cp)^{-})(it)$$

In this approach, we consider only placement events contributing to the inventory levels at specific collection points. This ensures that the demand is derived from actual recorded movements within the QEL.

Example 1 (Demand Determination from QEL)*.* Given the QEL in Tables [2.1,2.2,2.3,2.4], we now calculate the monthly demand for pizza item types at the planning collection point.

For this example, assume the following:

- The item types for which we can calculate the demand for are $I^{demand} = \{Dough, Tomato\_sauce, Cheese\}$. in this example we only consider the $Tomato\_sauce$ item type.
- The collection point for demand is the *planning* collection point.
- The demand is calculated based on arrival events, which in this case are the events with ocel_type *Send pizza*.

We extract the following relevant events for *Send pizza*:

- **January 2025**:
  - Event `ev_11` at `2025-01-01 00:00:09`
  - Event `ev_12` at `2025-01-01 00:00:20`

  - Quantity Operations: $abs(eqty^-)(ev\_11)(planning, Tomato\_sauce) = 2$ and $abs(eqty^-)(ev\_12)(planning, Tomato\_sauce) = 2$.

- **February 2025**:
  - Event `ev_17` at `2025-02-01 00:00:11`
  - Quantity Operation: $abs(eqty^-)(ev\_17)(planning, Tomato\_sauce) = 7$.

- **March 2025**:
  - Event `ev_18` at `2025-03-01 00:00:22`
  - Quantity Operation: $abs(eqty^-)(ev\_18)(planning, Tomato\_sauce) = 3$.

Thus, the monthly demand for pizzas at the planning collection point is computed as:

$$\begin{aligned}
d(Tomato_sauce, planning, \text{Jan 2025}) &= 2 + 2 = 4,\\
d(Tomato_sauce, planning, \text{Feb 2025}) &= 7,\\
d(Tomato_sauce, planning, \text{Mar 2025}) &= 3.
\end{aligned}$$

This example demonstrates how the QEL can be used to derive monthly demand for pizza items at a specified collection point.

For the following, we assume that for every event with a removing quantity operation to the demand-specifying collection points only refers to a single order object. Using this assumption, we can determine the demand associated with a single order object:

Definition 5.1.2 (Order Demand)**.** Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log, $O^{order} \subseteq O$ the set of orders, and $CP^{dem} \subseteq CP$ the subset of demand-specifying collection points. Assuming every event with a removing quantity operation to the demand-specifying collection points only refers to a single order object, we can determine the demand associated with the order:

$$dem(o^{order}) = \bigoplus_{e \in E_{o^{order}}, cp \in CP^{dem}} abs(eqty(e)^-)(cp)$$

where $E_{o^{order}} = \left\{ e \in E \mid (e, qual, o^{order}) \in E2O \right\}$

5.2 Lead Time

One of the most important measures according to our review is the Lead Time. For order object $o^{order} \in O^{order}$, we need to be able to identify 1) the event of placing this order, and 2) the event describing the order's arrival. Hence, we assume that the QEL contains two activities $a^{pl} \in A$ and $a^{arr} \in A$ describing the placement and arrival of orders (or rather the deliveries associated to such every order). To determine the demand and the lead time, we assume that every order is involved in exactly one event of each activity.

Definition 5.2.1 (Order Placement/Arrival Events)**.** Let $QEL = (OCEL, I, CP, eqty, \prec_e$ $)$ be a quantity event log, and $O^{order} \subseteq O$ the set of orders. The placement function $pl : O^{order} \not\rightarrow E$ specifying the placement event for every order object if the following requirement is met. For every order $o^{order} \in dom(pl)$ there is exactly one event of the placement activity with the order object, i.e., $|\left\{(e, q, o^{order}) \in E2O \mid act(e) = a^{pl}\right\}| = 1$. If no such single event exists, we write $pl(o^{order}) =\perp$.

Equivalently, the arrival function $arr : O^{order} \not\rightarrow E$ specifies the arrival event for every order object if the following requirement is met. For every order $o^{order} \in dom(arr)$ there is exactly one event of the arrival activity with the order object,
i.e., $|\left\{(e, q, o^{order}) \in E2O \mid act(e) = a^{arr}\right\}| = 1$. If no such single event exists, we write $arr(o^{order}) =\perp$.

Using these events, we can determine the lead time by determining the timedelta between these event's timestamps.

Definition 5.2.2 (Lead Time Detection)**.** Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log and $O^{order} \subseteq O$ the set of orders. Given the two functions pl : $O^{order} \rightarrow E \cup \{\perp\}$ and $arr : O^{order} \rightarrow E \cup \{\perp\}$ which associate every order with one or none placement and arrival events, the lead time for any order $o^{order} \in O^{order}$ is:

$$lt(o^{order}) = \begin{cases} time(arr(o^{order})) - time(pl(o^{order})) & \text{if } arr(o^{order}) \neq \perp \text{ and } pl(o^{order}) \neq \perp \\ \perp & \text{else.} \end{cases}$$

We will now demonstrate the determination of the lead time for the example QEL in Section 2.3.

Example 2 (Lead Time Determination). Given the QEL in Section 2.3, we can determine the lead time for every order $o^{order} \in O^{order}$ by identifying the placement and arrival events for each order. Let $O^{order} = \{o_3, o_4, o_5, o_6\}$ be the set of orders which are the pizza that we sold in the example. For every order $o^{order} \in O^{order}$, the placement events would be as presented in table 5.1, and the arrival events would be as presented in table 5.2. Using the placement and arrival events, we can determine the lead time for every

Table 5.1: Order Placement Events

o^{order}	$pl(o^{order})$
o_3	ev_7
o_4	ev_8
o_5	ev_13
o_6	ev_14

Table 5.2: Order Arrival Events

o^{order}	$arr(o^{order})$
o_3	ev_{11}
o_4	ev_{12}
o_5	ev_{17}
o_6	ev_{18}

order $o^{order} \in O^{order}$ as follows: The lead time for every order $o^{order} \in O^{order}$ is as presented in Table 5.3.

Table 5.3: Lead Time for Orders

o^{order}	$lt(o^{order})$	lead time (min)
o_3	$time(ev_{11}) - time(ev_7)$	25
o_4	$time(ev_{12}) - time(ev_8)$	30
o_5	$time(ev_{17}) - time(ev_{13})$	20
o_6	$time(ev_{18}) - time(ev_{14})$	40

5.3 Service Level

One of the main goals of inventory management is to serve the customer [1]. Therefore, observing and monitoring service levels is crucial. Bad service levels can lead to customer loss and, consequently, a decline in sales [20]. Studies indicate a strong correlation between service levels and customer satisfaction [20].
Two types of service levels are considered:

- **Alpha Service Level** (α): Defined as the probability that incoming demand can be fully met from the physical inventory available at the time of its arrival [21].

- **Beta Service Level** (β): Represents the percentage of demand that can be immediately satisfied from stock on hand [22].

For the **service level assessment**, we should be able to consider the multiple arrival events per order. Hence, we slacken this requirement specified in Definition 5.2.1. Instead, we define $Arr : O^{order} \to \mathcal{P}(E) \cup \{\emptyset\}$ which assigns a set of arrival events to orders. This may occur in situations such as stockouts where the order can be fulfilled in multiple deliveries.

To be able to compute service levels, we need to be able to compare the demand of an order with the quantity of items delivered in the first delivery. To do so, we require two functions:

- the demand associated with every order $dem : O^{order} \not\rightarrow \mathcal{I}(I)$, and

- the quantities of items delivered in the fist delivery $del : O^{order} \not\rightarrow \mathcal{I}(I)$ of an order.

In the previous section, we described a (strongly constrained) way to derive such a demand function. For the delivery function, we assume that for every order $o^{order} \in O^{order}$ and arrival event $e^{arr} \in Arr(o^{order})$ the absolute quantity operations to a subset of delivery-specifying collection points $CP^{delivery} \subseteq CP$ describe the number of delivered items.

Definition 5.3.1 (Initially Delivered Quantities)**.** Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log, $O^{order} \subseteq O$ the set of orders, and $CP^{del} \subseteq CP$ the subset of delivery-describing collection points Further,

$$Arr(o^{order}) = \left\{ e \in E \mid \begin{array}{c} act(e) = a^{arr} \wedge \\ (e, q, o^{order}) \in E2O \wedge \\ \left| \left\{ (e, q, o) \in E2O \mid o \in O^{order} \right\} \right| = 1 \end{array} \right\}$$

provides the arrival events per order object. We determine the delivered quantity of the first delivery:

$$del(o^{order}) = \bigoplus_{cp \in CP^{del}} abs(eqty(argmin_{e \in Arr(o^{order})} \{time(e)\}^{-})(cp)$$

If $Arr(o^{order}) = \emptyset$ for an object $o^{order} \in O^{order}$, $del(o^{order}) = \perp$.

Using these two counters per object, we can easily determine the alpha and beta service levels.

Alpha Service Level Calculation

To calculate the alpha service level, we analyze all orders and determine whether the demand quantity is equal to the quantity of the first delivery. The alpha service level is given by:

Definition 5.3.2 (Alpha Service Level). Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log, $O^{order} \subseteq O$ the set of orders, $dem : O^{order} \nrightarrow \mathcal{I}(I)$ the demanded quantities per order, and $del : O^{order} \nrightarrow \mathcal{I}(I)$ the initially delivered items per order.

$$\alpha = 1 - \frac{|\{ o \in O^{order} | dem(o^{order}) \neq del(o^{order}) \wedge dem(o^{order}) \neq \perp \wedge del(o^{order}) \neq \perp \}|}{|O^{order}|}.$$

In other words, α represents the fraction of orders that are fully satisfied.

Beta Service Level Calculation

The beta service level can be determined on the item type level and over all item types. The fulfillment ratio for item type i is then defined as:

Definition 5.3.3 (Item Type Beta Service Level). Let $QEL = (OCEL, I, CP, eqty, \prec_e)$ be a quantity event log, $O^{order} \subseteq O$ the set of orders, $dem : O^{order} \nrightarrow \mathcal{I}(I)$ the demanded quantities per order, and $del : O^{order} \nrightarrow \mathcal{I}(I)$ the initially delivered items per order. Given an item type $it \in I$, the beta service level is

$$\beta(it) = \frac{\sum_{o^{order} \in O^{order}} del(o^{order})(it)}{\sum_{o^{order} \in O^{order}} dem(o^{order})(it)}$$

The overall beta service level is computed as the average fulfillment ratio over all items:

Definition 5.3.4 (Overall Beta Service Level)**.**

$$\beta = \frac{1}{|I|} \sum_{i \in I} \beta(it).$$

We will now demonstrate the determination of the alpha and beta service level for the example QEL in Section 2.3.

Example 3 (Service Level Determination)*.* Consider a simplified scenario extracted from the QEL:

- **Pizza-order** o_5**:**
  - Demanded quantities (recorded at placement): 3 units of item type Dough, 7 of item type Tomato-sauce and 6 of item type Cheese.
  - Recorded item levels at arrival: 3 units of item type Dough, 7 of item type Tomato-sauce and 4 of item type Cheese.
- **Pizza-order** o_6**:**
  - Demanded quantities (recorded at placement): 2 units of item type Dough, 3 of item type Tomato-sauce and 3 of item type Cheese.
  - Recorded item levels at arrival: 2 units of item type Dough, 3 of item type Tomato-sauce and 3 of item type Cheese.

For the **alpha service level**, we observe:

- Order o_5 is not fully satisfied since $eqty(pl(o_5)) \neq eqty(arr(o_5))$.
- Order o_6 is fully satisfied.

Thus,

$$\alpha = \frac{1}{2} = 0.5 = 50\%.$$

For the **beta service level**, we aggregate the bought and arrived quantities for each item:

- **Item type Dough:**
  - Total bought: $3 + 2 = 5$ units.
  - Total arrived: $3 + 2 = 5$ units.
  - Fulfillment ratio: $\beta_{Dough} = \frac{5}{5} = 1.0$.
- **Item type Tomato-sauce:**
  - Total bought: $7 + 3 = 10$ units.
  - Total arrived: $7 + 3 = 10$ units.
  - Fulfillment ratio: $\beta_{Tomato-sauce} = \frac{10}{10} = 1.0$.

- **Item type Cheese:**
  - Total bought: $6 + 3 = 9$ units.
  - Total arrived: $4 + 3 = 7$ units.
  - Fulfillment ratio: $\beta_{Cheese} = \frac{7}{9} = 0.78$.

The overall beta service level is then:

$$\beta = \frac{\beta_{Dough} + \beta_{Tomato-sauce} + \beta_{Cheese}}{3} = \frac{0.78 + 1.0 + 1.0}{3} = 0.926 = 92.6\%.$$

This example illustrates how both the alpha and beta service levels can be derived from the QEL data.

6 DISQVER

This chapter provides a description of the DISQVER[1] application. This includes implementation details and information on the libraries used (Section 6.1) and a typical user journey (Section 6.2).

6.1 Architecture

The DISQVER application is a web-based system designed for processing and analyzing QEL files. It consists of two main components: 1. A Python-based backend responsible for importing QEL files, executing computations, and deriving key inventory metrics as discussed in Chapter 5. 2. A TypeScript-based frontend, implemented with React[2], providing an interactive user interface for data exploration and analysis.

Backend

The backend is built using FastAPI[3], a modern web framework that enables efficient API development with asynchronous support. FastAPI leverages Pydantic [23] for data validation, ensuring type safety in API parameters and responses. Custom Pydantic validators and assertion checks are integrated to enhance data integrity.
The backend is responsible for:

- Managing file operations, including reading, writing, and caching QEL data.
- Performing demand analysis, lead time analysis, and service level analysis.
- Providing structured data via API endpoints for frontend visualization.
- Storing results in either an SQLite database or JSON files, depending on persistence requirements.

Frontend

The frontend is a single-page web application (SPA) built with React and TypeScript. It communicates with the backend via API requests and enables users to upload QEL files, view structured data, and interact with various analytical components.

The key frontend components include:

[1] `https://github.com/Nour125/disqver`
[2] `https://react.dev/`
[3] `https://fastapi.tiangolo.com/`

- **Upload Component**: Handles user file uploads.
- **QEL Overview**: Provides a structured summary of the loaded QEL data.
- **Interactive Graph**: Visualizes dependencies using D3.js.
- **Demand Analysis**: Computes and displays demand over time.
- **Lead Time Analysis**: Presents lead times of orders.
- **Service Level Analysis**: Evaluates fulfillment performance.

System Overview

The overall system architecture is illustrated in Figure 6.1. The frontend interacts with the backend through API calls, triggering computations and retrieving data for visualization. The backend processes requests, executes analyses, and manages data storage. The SQLite database and JSON storage serve as persistence layers for structured data.

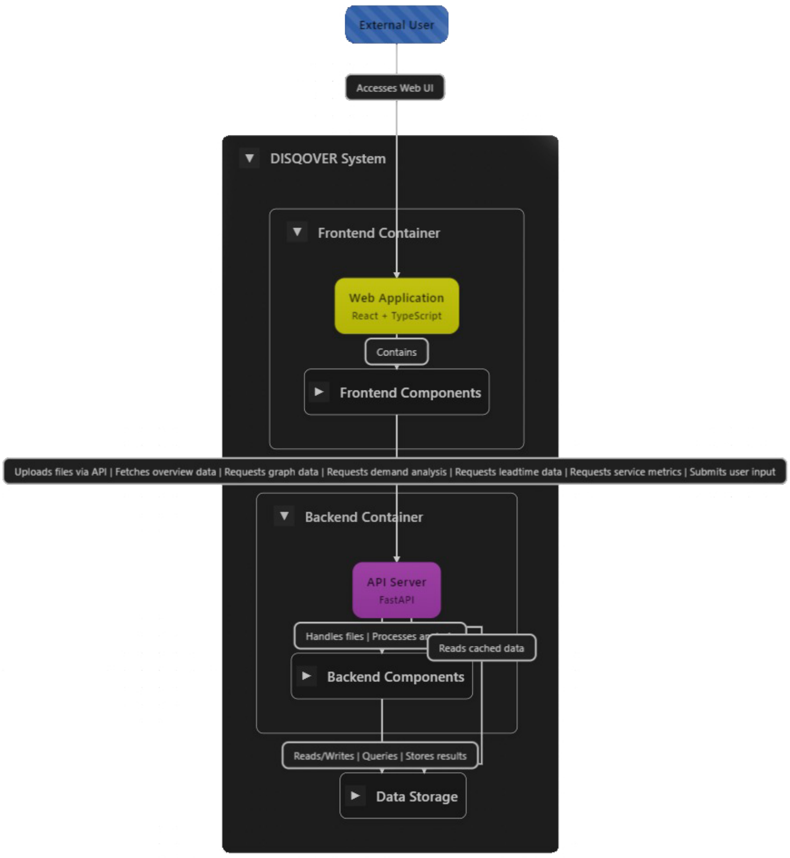


Figure 6.1: Architecture of the DISQVER application

6.2 User Journey

When opening the DISQVER application, the user is first presented with a file upload interface for event logs in the QEL SQLite format. The system requires a valid QEL file to proceed with the analysis.

Once an OCEL file is uploaded, the main dashboard (Figure 6.2) is displayed. The dashboard consists of multiple sections:

- **Overview**: Provides basic dataset information, including the number of events and objects, as well as the filename.
- **Input Information**: Requires the user to specify which object types should be considered as orders and and there placement/arrival activity.

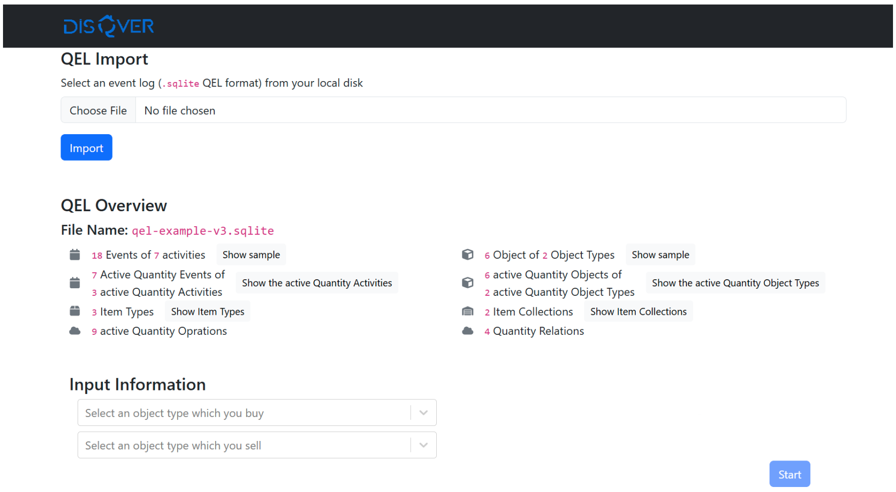


Figure 6.2: DISQVER Overview

After providing the necessary input, the user can start the analysis by clicking the *Start* button. The system then processes the data and presents the workflow as a **Quantity Net**, which visualizes material flows within the dataset.
Following the workflow visualization, the user is directed to the main analysis section (Figure 6.3), which consists of three key sections:

- **Demand Analysis**.
- **Lead Time Analysis**.
- **Service Level Analysis**.

Each section allows users to interact with the data, explore trends, and gain insights into the inventory and order management processes.

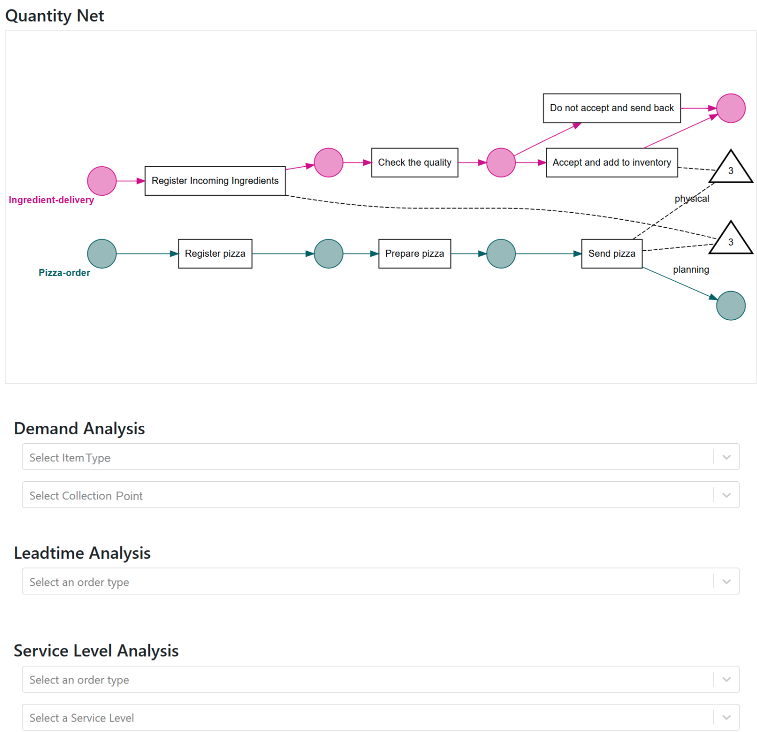


Figure 6.3: DISQVER Analysis

7 Evaluation

This chapter presents the evaluation of the methods introduced in Chapter 5 and the DISQVER application described in Chapter 6. The evaluation is structured around RQ4 and focuses on assessing the effectiveness of the calculated metrics. For the evaluation we will use a public QEL[1]. The QEL has been modified. for the modification we deleted two object types and an activity.

In Section 7.1, we introduce the QEL used for evaluation. Section 7.2 presents the visualization results and discusses the insights they provide.

7.1 Input Data

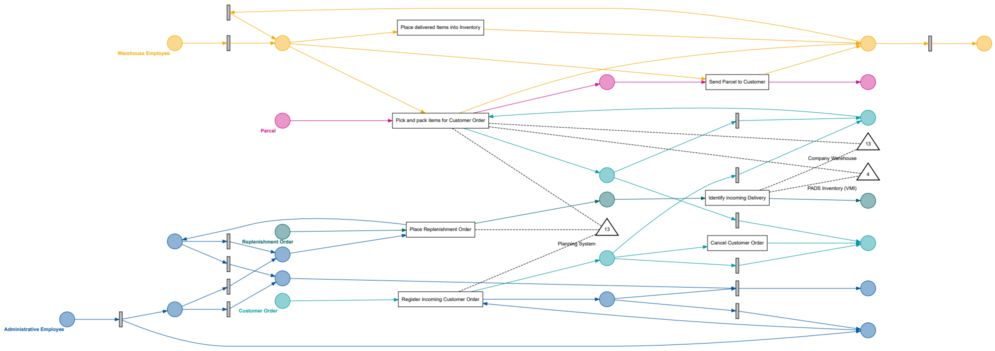


Figure 7.1: Quantity Net from the QEL used for evaluation

The input QEL consists of 7 activities and 5 object types, executed across 3,495 events and 2,308 objects. Additionally, the QEL contains 17 item types and 3 collection points. For this evaluation, we consider the two object types *Replenishment Order* and *Customer Order* as order types.

From the discovered quantity net shown in Figure 7.1, we observe that *Replenishment Orders* are placed during the *Place Replenishment Order* activity and arrive at the *Identify Incoming Delivery* activity.

[1] `https://github.com/ninagraves/qel_simulation`

For *Customer Orders*, placement occurs in the *Register Incoming Customer Order* activity, while delivery is tracked via the *Parcel* object type. Ideally, the arrival activity should be *Send Parcel to Customer*, but due to the absence of a direct mapping between *Customer Orders* and *Parcel* objects in the QEL, this approach is not feasible (more to that in chapter 8). Instead, we consider *Pick and Pack Items for Customer Order* as the arrival activity for *Customer Orders*.

Regarding collection points, the *Planning System* is responsible for placing both *Replenishment Orders* and *Customer Orders*, making it the planning collection point. The *Company Warehouse* and *PADS Inventory (VMI)* serve as the physical collection points for these orders.

The corresponding planning collection point for the *Company Warehouse* is the *Planning System*. However, no dedicated planning collection point exists for the *PADS Inventory (VMI)*. Consequently, for item types stored in the *PADS Inventory (VMI)*, service level and lead time calculations cannot be performed.

In the *Company Warehouse* are the following item types: {*Speedometer, Pedal, Brake Cable (2), Handles (2), Tube, Box, Brake Shoes (4), Saddle, Back Light, Tire, Bell, Front Light, Helmet*}. And in the *PADS Inventory (VMI)* are the following item types: {*'PADS Brake Shoes (2)', 'PADS Brake Cable (2)', 'PADS Tire', 'PADS Tube'*}

So the input in the tool will be as presented in figure 7.2.

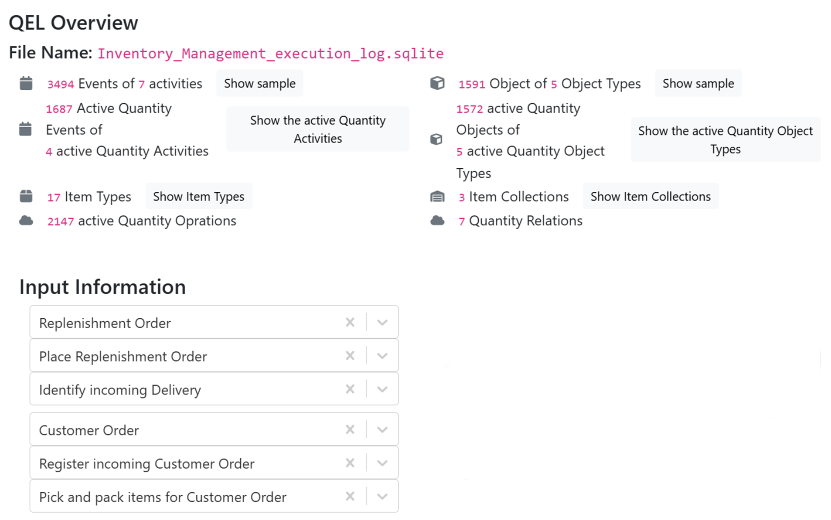


Figure 7.2: Input for the DISQVER application

7.2 Results

In this section, we present the visualizations for the calculated metrics, and discuss the insights they provide. The visualizations are generated using the DISQVER application described in Chapter 6.

7.2.1 Demand Analysis

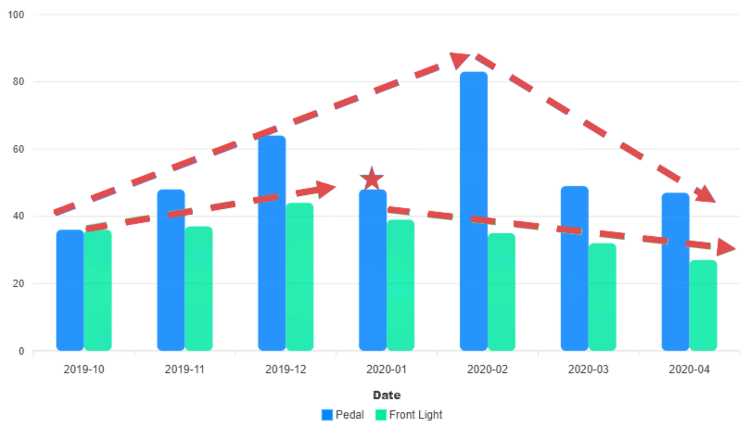


Figure 7.3: Monthly Demand (the star and arrows are not part of the demand visualization)

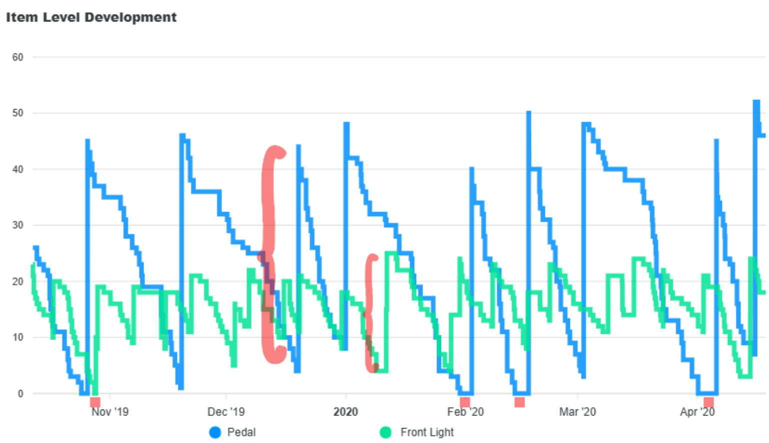


Figure 7.4: Item Level Development representing stock movements (all red representations are not part of the visualization)

For the evaluation, we focus on two selected item types: *Pedal* and *Front Light* at the *Planning system* collection point. The demand calculation is based only on items that have been removed from the collection point, without considering the activity *Cancel Customer Order*. Although cancellations also affect the demand calculation, the quantity of canceled orders is not recorded, so they cannot be incorporated. We can also observe that the quantity net contains loops, which may result in an order with an arrival event but no corresponding placement event; however, this issue is addressed by ensuring that each order has both a placement and an arrival event. Two distinct visualization techniques are employed to capture different facets of demand and inventory dynamics:

Monthly Demand: A bar chart (Figure 7.3) is used to display the monthly demand for each item type. Each bar represents the number of units sold during a particular month, which helps in identifying:

- Trends over time: The additional arrows presented in Figure 7.3 illustrate how the demand over several months increases and then decreases.
- Seasonal patterns: Although it is difficult to observe seasonal patterns in this log because it does not span several years, identifying such patterns is straightforward. By comparing the same months across different years, one can determine whether the demand in those months is also increasing.
- Sudden changes in sales: The star in Figure 7.3 indicates a sudden change in demand—where one can observe a month with high demand immediately followed by a month with a sharp decrease in demand.

Item Level Development: The quantity state at a specific collection point ($cp \in CP$) at the moment of an event ($e \in E$) is referred to as its *item level*, denoted as ($ilvl^{cp}(e) \in \mathcal{P}(I)$). We integrated this method from [6]. A line chart (Figure 7.4) illustrates the evolution of stock levels by tracking incoming and outgoing events over time. This visualization makes it possible to:

- Monitor inventory fluctuations: We can easily identify from Figure 7.4 when an order was added to stock and when it was removed, as the y-axis represents the number of available goods of that item type.
- Detect stockouts or overstock conditions: The red boxes in Figure 7.4 indicate stockouts.
- Evaluate the effectiveness of replenishment policies: The red curly bracket indicates the replenishment quantity. If an end user has a specific replenishment policy, they can assess its effectiveness by comparing the replenishment quantity to the quantity present before the replenishment occurred.
- Assess whether safety stock levels are appropriate: We can observe that the item type *Pedal* frequently experiences stockouts compared to the item type *Front Light*. This may indicate that the safety stock for *Pedal* needs to be adjusted, and corrective actions—such as increasing the safety stock level or implementing a safety stock policy—should be considered.

Insights Derived: The combination of these visualizations provides valuable insights into inventory management:

1. **Trend Analysis:** The monthly demand chart reveals overall demand trends, allowing identification of items with steady, rising, or falling demand. This is essential for recognizing seasonal effects or shifts in customer behavior.

2. **Forecasting Potential:** Historical demand data can be used to build more accurate forecasting models. A clear understanding of past trends supports better predictions of future sales, thereby optimizing inventory levels.

3. **Product Popularity:** Comparing demand across item types highlights which products are more popular, aiding decisions regarding product promotions or discontinuations.

4. **Stock Management:** The item level development chart provides a real-time view of stock variations, which is critical for detecting potential stockouts or overstocking. It also offers insights into order quantities (lot size) and the adequacy of safety stock levels.

Together, these visualizations support data-driven decision-making by offering a comprehensive view of both demand trends and inventory dynamics.

7.2.2 Lead Time Analysis

COLUMNS FILTERS DENSITY EXPORT

Order ID	Placed Time	Delivered Time	Lead Time
o-21	2019-10-12T12:43:44.891...	2019-10-16T00:06:15.680...	3 days 11 hours 22 minutes 30 seconds
o-46	2019-10-14T03:33:26.520...	2019-10-17T19:30:08.571...	3 days 15 hours 56 minutes 42 seconds
o-59	2019-10-15T14:33:52.938...	2019-10-17T14:52:17.124...	2 days 0 hours 18 minutes 24 seconds
o-89	2019-10-17T06:59:28.150...	2019-10-19T09:40:41.301...	2 days 2 hours 41 minutes 13 seconds
o-146	2019-10-19T18:02:56.542...	2019-10-24T04:58:45.974...	4 days 10 hours 55 minutes 49 seconds

1–94 of 94

Figure 7.5: Lead Time Table

In this evaluation, we focus on *Replenishment Orders* as the representative order type for assessing lead time. It is important to note that there is a small inconsistency between the formalization of lead time and its implementation. We define the placement and arrival activities as arbitrary and do not require them to be identical for a given order type. However, if there is a subset of this order type that has a different arrival activity yet still satisfies our conditions, the implementation—which selects only one placement

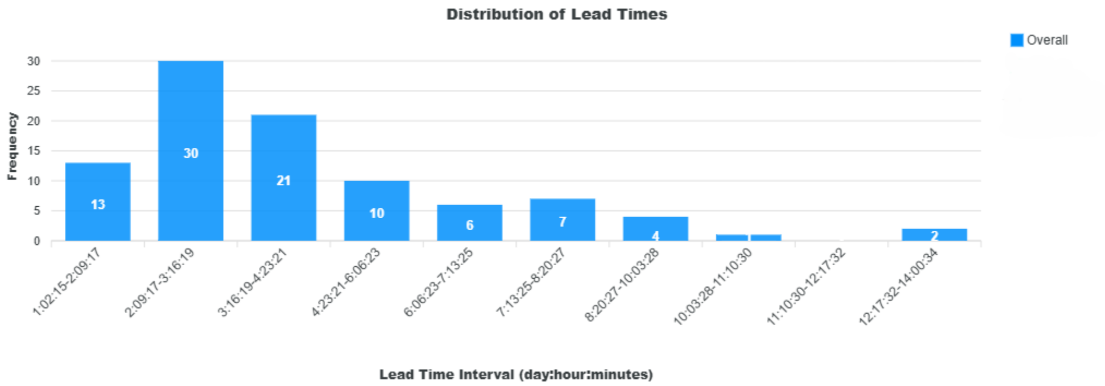


Figure 7.6: Lead Time Distribution

and one arrival per order type—will calculate lead time for only the assessed orders. To address this, the user would need to adjust the input settings to display the different lead times for these orders. Two visualization techniques are used to provide complementary perspectives on the lead time data:

1. **Tabular Representation:** The table in Figure 7.5 presents the calculated lead time for each order individually. This format allows for a detailed, order-level examination and facilitates the identification of specific instances with unusually high or low lead times. For instance, if we click on the lead time column, the data will be sorted, making it easy to identify order IDs with exceptionally high or low lead times, which can then be further investigated.

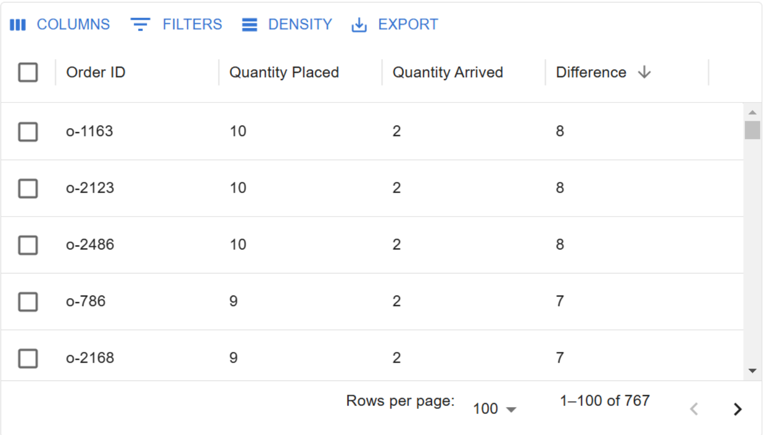
2. **Histogram:** The histogram shown in Figure 7.6 displays the overall distribution of lead times. This visualization is useful for:
   - Identifying the average lead time across orders: in this case, the histogram suggests that the average lead time is between 2 to 3 days.
   - Analyzing the variability in lead times: The histogram reveals that the number of orders with lead times exceeding the average is higher than those with lead times below the average.
   - Detecting outliers or delays in the order process: for example, Figure 7.6 shows two orders with lead times exceeding 12 days, which can be considered as outliers.
   - Determining whether the majority of orders are completed within the expected timeframe: although the QEL does not specify a timeframe for the order types, if such a timeframe were provided, it would be easy to identify orders that fall outside this range.

Together, these visualizations provide a comprehensive view of lead time performance, offering insights into both individual order performance and overall process efficiency.

7.2.3 Service Level Analysis

For the evaluation of service levels, we focus on the order type *Customer Order*. The analysis considers both the alpha and beta service levels to provide insights into the effectiveness of inventory management in meeting customer demand.

Alpha Service Level


COLUMNS FILTERS DENSITY EXPORT

Order ID	Quantity Placed	Quantity Arrived	Difference
o-1163	10	2	8
o-2123	10	2	8
o-2486	10	2	8
o-786	9	2	7
o-2168	9	2	7

Rows per page: 100 1–100 of 767

Figure 7.7: Alpha Service Level Table

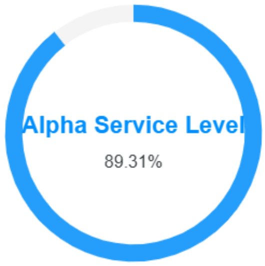


Figure 7.8: Overall Alpha Service Level

The alpha service level evaluates the frequency and extent to which customer orders can be fully fulfilled from available inventory. Two visualization techniques are used:

1. **Tabular Representation (Figure 7.7):** This table details how often customer orders could not be fully met and quantifies the shortfall for each affected order. This information is crucial for identifying the order IDs that are not fully fulfilled. The table also allows for exporting these IDs so that the end user can perform further investigation.
2. **Radial Chart (Figure 7.8):** The radial chart presents an aggregated view of the overall alpha service level. A low service level indicates frequent stockouts(as we saw for the item type *Pedal*), which can lead to customer dissatisfaction and potential loss of sales. If the observed service level is below an acceptable threshold, corrective actions—such as increasing safety stock—may be necessary.

Beta Service Level

The beta service level assesses the proportion of ordered quantities that were successfully delivered. This is visualized using two different approaches:

1. **Polar Chart (Figure 7.9):** This chart displays the fulfillment ratio for each item type, helping to identify products with low fulfillment rates that may require

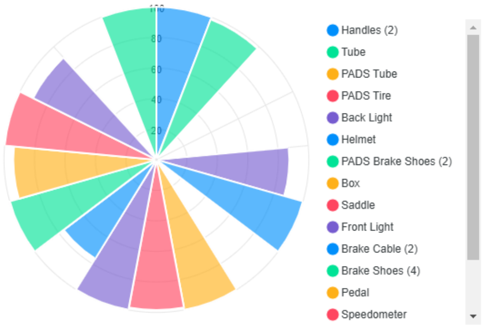


Figure 7.9: Fulfillment Ratio per Item Type

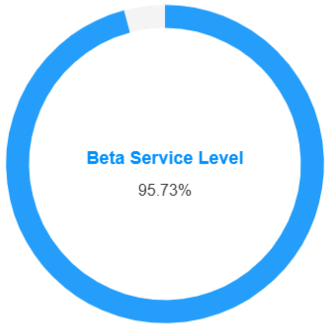


Figure 7.10: Overall Beta Service Level

further investigation. Such insights can guide improvements in procurement and stock allocation strategies.

2. **Radial Chart (Figure 7.10):** The radial chart provides an overall measure of the beta service level across all customer orders. This metric is particularly useful for evaluating the overall effectiveness of inventory management in meeting customer demand.

If the business has a specific service level target, these visualizations can help assess whether current inventory policies align with expectations. If the observed service levels fall short, adjustments—such as optimizing reorder points or improving supplier reliability—may be required.

8 Discussion

In this chapter, we discuss the benefits and limitations of our approach. to do so we will go over the method we defined in chapter 5.

As demonstrated in the evaluation, several valuable insights could be identified. For demand analysis, we were able to observe trends over time, assess forecasting potential, analyze differences in product popularity, and evaluate stock management patterns. Regarding lead time, we identified the lead time for each order object, calculated the average lead time, analyzed variability in lead times, and detected potential outliers. For service level analysis, we examined how often customer orders could not be fully met, quantified the shortfall for each affected order, and assessed the probability of frequent stockouts.

Nevertheless we face some limitations so to make use of our approach, the placement and arrival activities must include active quantity operations, meaning that they need to have an non-zero item quantity. If we take another look at the event log from the evaluation, we can see that it makes more sense to choose *Send Parcel* as the arrival activity because *Send Parcel* represents the sending of the order. However, *Send Parcel* does not have active quantity operations.

Regarding demand analysis, it is important to clarify that we only consider orders that have been removed from collection points. This approach disregards the fact that orders may be canceled (as observed in the evaluation, where a cancellation activity exists for the order type *Customer Order*) or returned (although this does not occur in the current evaluation but it is possible), both of which can impact demand calculations. Therefore, further investigation is necessary to understand how such data should be processed.

Regarding the calculation of lead time, we do not require the quantity operation. Nevertheless, we still cannot choose *Send Parcel* because there is no connection between the *Customer Order* and *Parcel* object types. This is why we have also met the assumption that orders of the same type ($ot \in OT$) should have the same placement and arrival activities. also one main aspect of calculation of lead time, is the exciting of both the placement and arrival activity. So if an order type lacks either a placement or an arrival activity, the lead time calculation fails. Also multiple placement or arrival activities can make it difficult to calculate the lead time. that is why we assume exactly one placement and one arrival activity per order type.

As with the alpha service level, our approach faces similar limitations to those observed with lead time calculations, since both rely on the placement and arrival events. Missing activities or the presence of multiple placement or arrival events can compromise the ability to accurately calculate the alpha service level. Additionally, both service levels assume that an order may have multiple deliveries. However, if the end user does not send multiple deliveries but instead waits until the order is complete before dispatching it, the service level calculation becomes ineffective. Nevertheless, since we also calculate lead time, such cases will still be noticeable in the analysis.

While our method introduces certain constraints, it offers a structured approach for deriving key inventory metrics that were previously difficult to compute using conventional process mining techniques. By leveraging QEL, we provide a **quantitative foundation for inventory analysis**, filling a critical gap in process mining applications for logistics. So our approach can still provide valuable information that supports more thoughtful, data-driven decision-making. Additionally, the framework can be extended to incorporate further functionalities, making this work a foundational step toward building an inventory management system that operates with the assistance of process mining.

9 Conclusion

This thesis has explored the challenges inherent in modern inventory management within logistics and has demonstrated how data-driven approaches can enhance operational decision-making. By making use of QELs, this work has shown that it is possible to derive key inventory metrics that provide deeper insights into process behaviors and inefficiencies.

The primary problem addressed in this thesis was the inadequacy of conventional process mining techniques to fully support the analysis of inventory management systems. The gap identified in the existing literature, where Quantity Event Logs remain underutilized, served as the motivation for the development of a new method capable of calculating meaningful metrics from such data.

To tackle this problem, a systematic approach was proposed and detailed in Chapter 5. After the meta-review that identified the most relevant performance indicators namely (lead time, demand, and service level). Essential data requirements and event log structures were established, ensuring that the derived metrics capture the complexity of inventory processes.

The implementation of the method is realized in the DISQVER web application, as described in Chapter 6. This tool integrates the derived metrics into a platform that supports detailed visualization and analysis. The application's architecture, based on a FastAPI backend and a React-based frontend, facilitates efficient data processing.

The evaluation presented in Chapter 7 shows the practical benefits of the proposed approach. An analysis performed on a QEL dataset has demonstrated that the method not only enhances the detection of process inefficiencies but also supports more informed, data-driven decision-making in inventory management.

overall this thesis provides a structured framework for the derivation of key inventory metrics from Quantity Event Logs, thereby filling an important gap in process mining research and implementation of the DISQVER application illustrates how these insights can be transformed into actionable tools for inventory control.

9.1 Future Work

There are several promising avenues for future work that could enhance the functionality and applicability of the DISQVER tool. One valuable improvement would be to enable users to upload an OCEL along with a quantity operation table, so that the tool can automatically generate a complete QEL. Since many operations already maintain an OCEL, requiring only the additional quantity operation table would streamline the data preparation process and broaden the tool's usability.

Furthermore, as shown in Table 4.2, a wide range of metrics relevant to inventory management have been identified. Due to the scope of this thesis, only a subset of these metrics has been implemented. Future work should focus on integrating the remaining metrics, which would allow for a more comprehensive evaluation and deeper insights into inventory performance.

In addition to expanding the metric set, further improvements can be made by addressing the limitations discussed in Chapter 8. Enhancing the tool in this way would not only improve its overall functionality and reliability but also adapt it to more complex real-world scenarios.

Another potential extension is to incorporate a quantity operation table for objects, as two of the metrics are currently calculated at the object level. Providing objects with their own quantity operation table would enable more detailed tracking and analysis, thereby increasing the precision of the metrics.

Finally, future work could also explore the integration of additional object attributes into the metric calculations. For example, if a replenishment order includes supplier information, the tool could be extended to calculate the lead time for each supplier individually. This enhancement would provide more granular insights and support targeted decision-making in inventory management.

Bibliography

[1] J. W. Toomey. *Inventory management: principles, concepts and techniques.* Vol. 12. Springer Science & Business Media, 2000.

[2] R. D. Reid and N. R. Sanders. *Operations management: an integrated approach.* John Wiley & Sons, 2019.

[3] G. van Heck. "INVENTORY MANAGEMENT." In: *Delft University of Technology* (2009).

[4] W. Van Der Aalst. "Process mining: Overview and opportunities." In: *ACM Transactions on Management Information Systems (TMIS)* 3.2 (2012), pp. 1–17.

[5] C. dos Santos Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, and E. E. Scalabrin. "Process mining techniques and applications–A systematic mapping study." In: *Expert Systems with Applications* 133 (2019), pp. 260–295.

[6] N. Graves, T. Brockhoff, I. Koren, and W. M. van der Aalst. "Extending Process Intelligence with Quantity-Related Process Mining." In: ().

[7] A. Berti, I. Koren, J. N. Adams, B. Knopp, M. Rafiei, N. Graves, G. Park, Y. Zhang, L. Tacke genannt Unterberg, V. Peeva, M. Pegoraro, and W. M. P. van der Aalst. "OCEL (Object-Centric Event Logs) 2.0 Specification." 2023.

[8] F. H. Staudt, G. Alpan, M. Di Mascolo, and C. M. T. Rodriguez. "Warehouse performance measurement: a literature review." In: *International Journal of Production Research* 53.18 (2015), pp. 5524–5544.

[9] D. Kretzschmann, A. Berti, and W. M. van der Aalst. "Optimizing Inventory Management using Object-Centric Process Mining." In: ().

[10] D. Knoll, G. Reinhart, and M. Prüglmeier. "Enabling value stream mapping for internal logistics using multidimensional process mining." In: *Expert Systems with Applications* 124 (2019), pp. 130–142.

[11] A. Rzad, J. Wojnecka, M. Rutkowski, and M. Guliński. "Investigating Purchase-to-Pay process using Process Mining in a multinational corporation." In: *Business Process Intelligence Challenge 2019* (2019).

[12] A. Berti, U. Jessen, G. Park, M. Rafiei, and W. M. van der Aalst. "Analyzing interconnected processes: using object-centric process mining to analyze procurement processes." In: *International Journal of Data Science and Analytics* (2023), pp. 1–23.

[13] L. Terlouw. “Optimization of Logistics Processes by Mining Business Transactions and Determining the Optimal Inventory Level.” In: *Proceedings of the EEWC Forum*. 2017.

[14] J. P. S. Piest, J. A. Cutinha, R. H. Bemthuis, and F. A. Bukhsh. “Evaluating the use of the open trip model for process mining: An informal conceptual mapping study in logistics.” In: *23rd International Conference on Enterprise Information Systems, ICEIS 2021*. SCITEPRESS. 2021, pp. 290–296.

[15] T. De Kok, C. Grob, M. Laumanns, S. Minner, J. Rambau, and K. Schade. “A typology and literature review on stochastic multi-echelon inventory models.” In: *European Journal of Operational Research* 269.3 (2018), pp. 955–983.

[16] A. S. Eruguz, E. Sahin, Z. Jemai, and Y. Dallery. “A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization.” In: *International Journal of Production Economics* 172 (2016), pp. 110–125.

[17] L. N. K. Duong, L. C. Wood, and W. Y. Wang. “A review and reflection on inventory management of perishable products in a single-echelon model.” In: *International Journal of Operational Research* 31.3 (2018), pp. 313–329.

[18] S. Lutz, H. Löedding, and H.-P. Wiendahl. “Logistics-oriented inventory analysis.” In: *International Journal of Production Economics* 85.2 (2003), pp. 217–231.

[19] S. Nallusamy. “Performance measurement on inventory management and logistics through various forecasting techniques.” In: *International Journal of Performability Engineering* 17.2 (2021), p. 216.

[20] A. Salam, F. Panahifar, and P. J. Byrne. “Retail supply chain service levels: the role of inventory storage.” In: *Journal of Enterprise Information Management* 29.6 (2016), pp. 887–902.

[21] H. Tempelmeier. *Bestandsmanagement in supply chains*. BoD–Books on Demand, 2005.

[22] S. Axsäter. *Inventory control*. Vol. 225. Springer, 2015.

[23] S. Colvin, E. Jolibois, H. Ramezani, A. Garcia Badaracco, T. Dorsey, D. Montague, S. Matveenko, M. Trylesinski, S. Runkle, D. Hewitt, and A. Hall. *Pydantic*. 2024. URL: `https://docs.pydantic.dev/latest/`.

Additionally, to enhance grammar and language, DeepL Write and ChatGPT are used.