Norges miljø- og
biovitenskapelige
universitet

Master Thesis 2020 30 ECTS
School of Economics and Business
Supervisor: Jens Bengtsson

Process Mining: Construction of an Event Log and Process Discovery within a Return-Order Process

Miriam Magnusson Touiti & Håvard Kopland Sand
Master of Science in Business Administration with Major in Business Analytics
School of Economics and Business

Abstract

In recent years, organizations have expressed a rapidly growing interest in improving their end-to-end processes by using the powerful tool of Process Mining, taking advantage of data in order to discover their actual business processes. Currently, poor data quality costs around $3 trillion per year and only 3% of firm's data meets basic quality standards. Consequently, businesses have acknowledged the potential of utilizing unstructured raw data, transforming it into an event log, thereby enabling improvement of their operational processes.

In context of the Supply Chain Management process of return orders in SAP, this thesis emphasizes on developing a step-by-step guide for the construction of an event log, in order to enable Process Mining and subsequently evaluating the Discovered process model.

Through an analysis of a 2019 SAP-data extraction of a company in the car parts business, this study develops a *six-step guide* towards a complete *event log* aimed at visualization and analysis of the return-process of sales orders. The analysis describes an approach to identifying and separating process instances, order events and construct timestamps, extract activities, in addition to extracting and enriching event data to form the event log.

Process analysis in the form of Process Discovery is made possible by utilizing the steps of the developed guide. Furthermore, the quality of the resulting process model including the representative behavior seen in the event log is evaluated by applying a four-dimensional framework. The dimensions *Replay Fitness*, *Simplicity* and *Precision* is characterized with a plus-symbol (+), whereas the dimension of *Generalization* is characterized by a minus-symbol (-).

The approach to construct activities in the event log is highlighted as a likely root cause of the process model's low score on Generalization. Furthermore, the current method of evaluating the quality of process models is considered to be lacking proper scaling capabilities, and further research on the topic is advised. After the evaluation, a supplementary case study utilizes the step-by-step guide on the extracted SAP-data, in order to illustrate the possible business insights that Process Mining may extract from the constructed return-order event log.

In closing, the thesis sums up the step-by-step guide, subsequently concluding that all steps are considered essential and that the resulting process model is of medium-plus (+) quality.

Sammendrag

De siste årene har sett en markant økning i etterspørselen etter å utnytte Process Mining til å forbedre eksisterende forretningsprosesser. Process Mining muliggjør at selskaper kan unytte store mengder data til å analysere *de facto* modeller, ved å konstruere høy-ytelses *hendelseslogger*. Totalt, er det estimert at lav datakvalitet medfører årlige kostnader på 3 billioner dollar, der 3% av selskapenes data er av holdbar kvalitet. Dette kan ha en årsakssammenheng med selskapers økte fokus på datadrevne kvalitetsforbedringer for å ta i bruk Process Mining.

Målet med denne gradsoppgaven er å utvikle en steg-for-steg guide for konstruksjon av en hendelseslogg, i konteksten *retur-ordre* i ERP-systemet SAP, for dermed å muliggjøre Process Mining. Deretter skal prosess-modellen evalueres for å avgjøre ytelsen.

Gjennom en analyse av et uttrekk av SAP-data fra 2019 av et selskap i bransjen for bilrekvisita, utvikler denne studien en *seks-stegs* guide mot en komplett hendelseslogg rettet mot å visualisere og analysere retur-prosessen for salgsordre. Analysen beskriver avgjørende fremgangsmåte for å identifisere og skille prosess-instanser fra hverandre, sortere og tidsbestemme hendelser, hente ut aktiviteter, samt knytte sammen og berike hendelsesloggen. Ved å benytte stegene i guiden muliggjøres prosessanalyse gjennom Process Discovery, samt påfølgende evaluering av prosess-modellen. Evalueringen ser på hendelsesloggens ytelse gjennom å bedømme prosess-modellen på de fire kvalitets-dimensjonene *Tilpasning*, *Enkelhet*, *Presisjon* og *Generalisering,* hvorpå de første tre oppnår høy score, mens sistnevnte scorer lavere.

I oppgavens diskusjon trekkes blant annet fremgangsmåten for konstruksjon av aktiviteter i hendelsesloggen frem som en hovedårsak til prosess-modellens lave score på dimensjonen generalisering. Videre identifiseres et behov for utvikling av en mer utdypende metode for evaluering av prosessmodeller innen Process Mining-disiplinen. Oppgaven presenterer dernest hvilke analytiske innsikter Process Mining kan bidra med gjennom en egen *case-studie* som er basert på den resulterende hendelsesloggen der oppgavens guide er anvendt.

Avslutningsvis summeres oppgavens seks-stegs guide, hvorpå det konkluderes med at samtlige steg er av avgjørende betydning, samt at den resulterende prosess-modellen er av medium-pluss (+) kvalitet.

Acknowledgements

This Master's Thesis has been written during the spring of 2020, as part of our degree; Master of Science in Business Administration, with major in Business Analytics.

We started off by setting a goal of discovering pecuniary values by constructing an event log and extract value from given data, within a company's supply chain process. With a strong idealistic conviction that this was indeed possible during a short period of time, we managed to provide further insight into a very interesting area of study – the return-order process. The research has been a wonderful experience and learning journey, which has increased the desire to continue learning about the interface between business and technology.

Throughout this thesis, we have been lucky to receive generous help and guidance from several stakeholders. First of all, we would like to thank Jo Sigurd Pedersen, Tom Einar Nyberg and KPMG Norway for providing us access to data, databases and guidance in order to make this collaboration possible. Thank you for giving us first-hand experience on applying Process Mining on a business - the process has been challenging, educational and interesting all at the same time. In addition, we would like to express our gratitude to KPMG's client for sharing data to scientific research in Process Mining. Furthermore, we would sincerely like to thank our supervisor and mentor, Associate Professor Jens Bengtsson for guiding us through this project during the extraordinary circumstances caused by the Covid-19 pandemic. Additionally, we are deeply grateful for the contributing inputs and guidance you have provided us with. Moreover, we also would like to express a thank you to family, friends and neighbours for their encouragement.

Lastly, we would like to express our gratitude to each other for supporting one another throughout the whole process of writing the Master Thesis. Thanks for all support through this stressful period, at the same time as we had to deal with a new everyday life due to the circumstances of Covid-19 pandemic, after we arrived Norway from exchange in Siena, Italy in February of 2020.

Håvard Kopland Sand and Miriam Magnusson Touiti

TABLE OF CONTENTS

LIST of TABLES and FIGURES

List of Figures

1 INTRODUCTION

This master thesis is concerned with organizations rapidly growing interest of improving end-to-end processes, by using the powerful tool of process mining to exploit the availability of data. Specifically, it examines the relationship between a company’s actual processes and recorded data on one hand, and process models on the other hand. By that, we mean that we want to provide insight into a company's existing process by using historical data. Moreover, we wish to get an in-depth understanding of given dataset based on the descriptive information it contains, in contrast to the normative perspective of process-mining literature.

Process mining is an emerging focus area that is impossible without proper event logs. The challenge is to extract process-related information from a variety of data sources, e.g., various tables, databases, files and logs (Aalst, 2016). Like many other data-driven approaches of analysis, such as Machine learning and Business Intelligence, Process Mining needs to deal with data quality problems. According to Harvard Business Review, only 3% of firm’s data meets basic quality standards (Nagle, Redman & Sammon, 2017). While a company’s salesforce waste time dealing with erred prospect data, vendors waste time correcting purchasing orders received from sale. IT-sections spend a great amount of time building system integration between interconnected networks that “don’t communicate”. Furthermore, data scientists work hours after hours cleaning data. These *hidden data factories* are time-consuming, expensive and form the basis for IBM’s estimation of poor data quality costs. In US alone, the expenses are around $3 trillion per year, in 2016 (Redman, 2016). Therefore, it is essential that companies have high-quality information systems to preserve data with good quality.

So far, the introduction has presented what process mining is, and the importance of good data-quality, in order to construct a proper event log. Next section will go deeper into what it takes for a company to construct an event log to become a digital master.

In an ideal situation, all businesses should solve the problem of constructing an event log by using a modern table builder in their information systems to detect and connect activities from tables, such as EVS Model Builder (Ingvaldsen & Gulla, 2008). This could be used to create an event-log, without accessing and manipulating data by using a programming language, such as SQL, Python or R. In this master thesis, we need to create an event-log by using a programming language. De Murillas, Reijers & van der Aalst (2018) explains that *event log extraction* is increasingly time-consuming and is barely supported. Usually, event logs are data assembled through an enormous number of tables, which need a complex combination of queries to extract activity logs. However, we aim to use the event log to map out and analyze an organization's existing business process, more specifically, the return-process. This thesis will not focus on a company as a whole, but mainly use the data extracted from their ERP-system to contribute, in order to turn the company into a digital master. American studies report that *Digital Masters*, i.e., companies that use digital technologies to drive significantly higher profit, productivity and performance exists, but are rare (Westerman, Bonnet & McAfee, 2014). The same studies show that these companies are 26 percent more profitable compared to their competitors. One of the key features and characteristics, together with other factors, is the individual company's ability to build digital capabilities by rethinking and improving their existing business processes.

Given these empirical studies, there is an increasing demand of enhancing the query building experience in a system, allowing for a more natural and user-friendly way (de Murillas, Reijers & van der Aalst, 2018, p.1239) for connecting data in databases with process mining. Process mining is a relatively young research discipline (Aalst, 2016, p. 31), so most findings of this study, are due to the steps of constructing the event log. However, various factors and skills are involved to enhance the query building experience, and these likely influences one or another. We take advantage of rich technological tools, where the data on sales- and distribution processes has not been used for KPMG's particular client, known as A-store. The client wishes to stay anonymous and is not a focus area here, because their shared data will only contribute to research. Even though the system experts at KPMG has provided us with relevant tables and data from their client, the construction and the designed event log is autonomically conducted by modelling with SQL queries. Furthermore, this thesis will provide a structural approach that businesses and practitioners can use, in order to complete their end-to-end project by using process-mining. To do so, this research seeks to:

Develop a step-by-step guide to construct an event log in the context of the Supply Chain Management process of return orders in SAP, aimed at Process mining, subsequently evaluating the Discovered Model

The purpose is to show how data can be transformed from raw data to an event log, without any component that monitors events, and creates logs automatically. These events represent a set of operations or actions the company's goods have processed through. The main domain is activities associated with return-processes of goods, after one of KPMG's attendees predicted this area of study as a potential area for improvement.

This thesis will split the main approach into two research sub-questions, in order to accomplish the goal. Each question is an essential part of the project, where the outcome will be utilized as input for the process analysis in the subsequent case study of A-store. We can define the questions as following:

1. What are the crucial steps in pre- processing data from SAP, with the purpose of constructing a return order event log aimed at Process mining?
2. Using the resulting event log in Process Discovery: What level of quality characterize the discovered model?

The research of Van der Aalst (2016) and Piessens (2011) is the fundamental for solving the first question. Moreover, the academic literature provides the latest contribution for the direction of practicing business process mining. Today's massive data volumes need to be put into a broader process context, as there is a growing demand in helping organizations to improve their operational processes (Aalst, 2016).

In order to answer the second question, the result of the first question must be solved. Furthermore, the extracted knowledge will hopefully provide the academic literature with empirical evidence within the field of logistics. Especially, within the area of return-processes by using data from the ERP System, SAP, to apply as input for Process Mining. Anyhow, next part of the introduction will provide an overview of the thesis structure:

The second section describes the literature of Process Mining to provide guidance towards answering which steps of constructing an event log are crucial. Given that this area of study is a relatively young research discipline, the theory enriches the analysis and enables an interesting discussion.

In the third part, the methodology chapter will provide insight into two parts. First of all, the database structure that the event-log is built upon is presented. The second part considers methods of evaluation, in terms of validating the goodness of the discovered model.

In Section 4, the analysis provides the approach to answering the research questions. Here, our findings are presented in a systematic manner, as we present our step-by-step guide. In other words, the theory is put into a context.

Section 5 provides a discussion and reflections upon answering the research questions, as well as shedding light on potential areas for improvement, based on the analysis. In addition to adding recommendations for further research, carried out from this area of study.

The following part, section 6, provides valuable insight in the extracted information and value from data stored in our event log, in the context of business optimization. More specifically, one may see the most frequent variants, to show what insights Process Discovery may give. Here, we also discuss the case study, in addition to come up with recommendations for potential improvements aimed towards the case company.
Lastly, section 7 briefly summarize and conclude the most important highlights of the thesis, particularly based on section 4 and section 5.

2 LITERATURE REVIEW

2.1 Process Mining

This chapter will provide a broader understanding of how we can systematically collect relevant information, needed to transform unclear data into valuable insight. The first part will use the literature of process mining to introduce the refined Process Mining framework of Van der Aalst (2016). This literature will provide guidance towards answering which steps of constructing an event log are crucial. In the following section, Van der Aalst will provide insight into limitations of modeling considering SAP. The last section includes Van der Aalst's four competing quality criteria to evaluate the quality of the discovered model, aimed at assisting us in solving the second research sub-question.

2.1.1 Operational Support - Refined Process Mining Framework

We assume the "world" consists of people, business processes, different organizations, documents etc. that has an information system that collect, record and support this "world". Looking from the perspective of auditing, it is important to ensure that event logs cannot be tampered with (Aalst, 2016, p.302) so no-one can influence the cases. Most of the process-mining techniques are working on *post-mortem* (historical) event data, which mean that the analyzed events are based on cases that have already been completed. This is known as *Business Process Provenance* and can be used for process improvement and auditing (Aalst, 2016). *Pre-mortem* (current) refers to event data with ongoing cases that are still running.

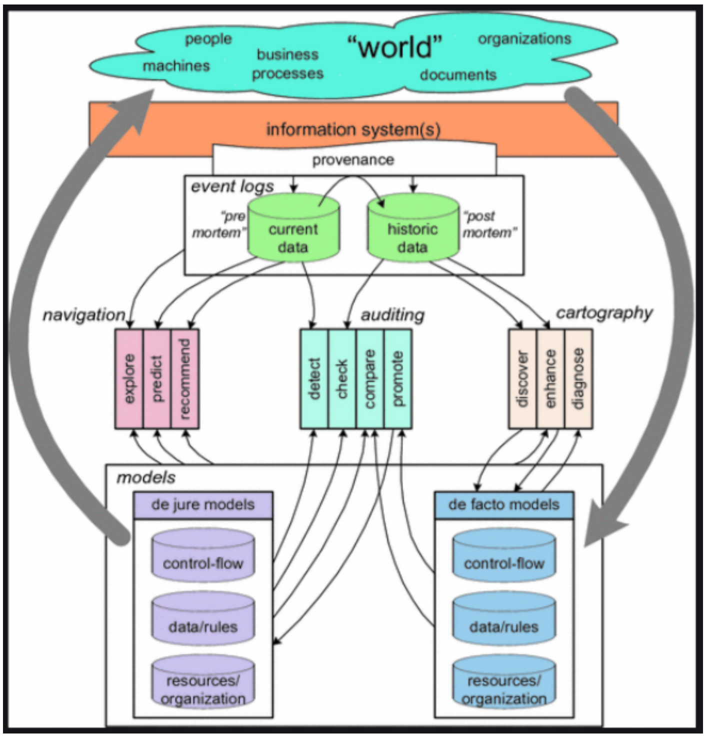


Figure 2.1: The Refined Process Mining Framework

The framework distinguishes between *de jure models* and *de facto models.* The *de jure models* are normative and include how things should be performed, comparing to *de facto models* which are descriptive and aim to reflect the reality. The two large arrows illustrate that de facto models arrives from the real world, and vice versa for de jure models. After refining the company's event logs into categories from above, we can identify ten process mining related activities. These are grouped into three categories: *cartography, auditing* and *navigation*.

2.1.2 Limitations of Modeling

The goal of process mining is to use event data to extract process-related information to discover a process, based on recorded data (Aalst, 2016). However, a model's value is limited if the business is paying too little attention to an idealized model that hides the reality. A system that is implemented in a business on a basis of illustrating an idealized business model, is likely to hide the reality of real processes. For example, it's waste of time modelling a log if the behaviour seen in the event-log don't reflect the real process. A nice illustration is the limited quality of most *reference models* (Aalst, 2016, p. 30). Such reference models are most likely to be used in large enterprise systems, such as SAP. The SAP reference model has little to do with the real processes supported by SAP. As much as 20% of the SAP models contain serious errors (Aalst, 2016). Given these considerations, the actual process related to event data, should be discovered, evaluated, adjusted and improved.

2.1.3 Process Discovery: Four Competing Quality Criteria

Process Discovery is one of three types within the field of Process Mining. The other two types are *Conformance* and *Enhancement*. A discovery technique uses an event log to construct a model by not using any a-priori, i.e. earlier information. The α-algorithm extracts an event log and construct process models in different forms, such as Petri nets, which explain the related information in the event log. This is an automatic construction.

The discovery can for instance show how many cases that are missing an invoice or show how many days it takes to deliver an item from one department to another. However, it is difficult to evaluate the quality of a Process Mining result. Van der Aalst (2016) divides the quality of the discovered model into four dimensions: *Replay fitness*, *Simplicity*, *Precision* and *Generalization*. All four dimensions have to be balanced, to achieve high-quality models.

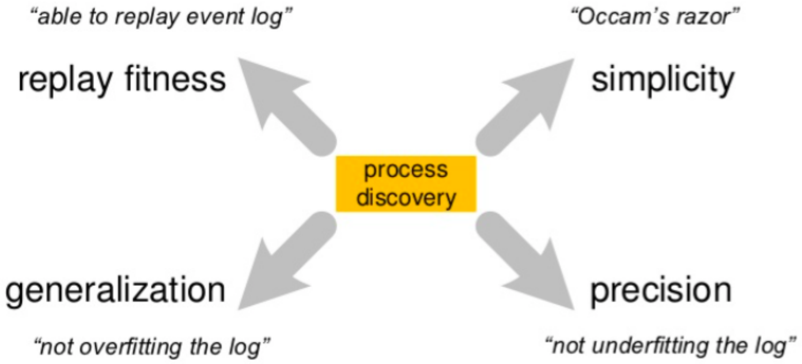
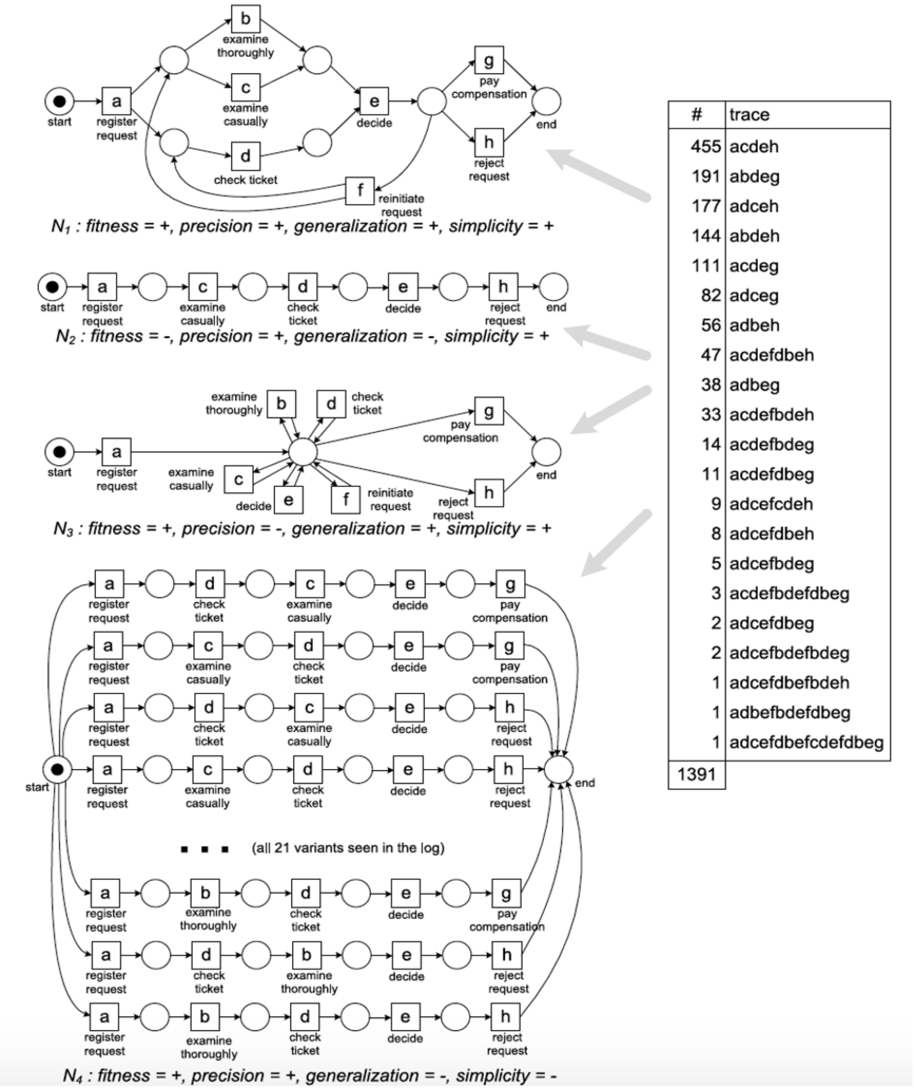


Figure 2.2: The Balance of Four Quality Dimensions with High-level Characterizations

Replay Fitness refers to a model that allows us to see the connected interaction seen in the event log. A model with perfect fitness can follow up all the traces from the beginning to the end of the process. This can be defined at a *case level*, e.g., the fraction of traces in the log that can be fully replayed, or at the *event level*, e.g., the fraction of events in the model (Aalst, 2016, p. 189). The defining question of determining whether it is a good fit or not, could be decided through answering: What are the consequences of skipping a step?

The *Simplicity* dimension in the context of process discovery, is associated with the term *Occam's Razor*. This indicates that the best model is the simplest model, as it can explain the behavior seen in the log. The model's complexity can be measured by defining the number of attributes and arcs. A *Precise* model refers to a model that does not allow "too much" behavior. By that, it means that a model which is not precise, is *underfitting*. On the other hand, a model that is overfitting, is a model that does not *Generalize* and only allows for the exact behavior stored in the log. A model should generalize and not reflect behavior. If the model is not generalized, it is *overfitting* and only illustrate the most common behavior to be recorded in the log (Aalst, 2016). The next paragraph will describe heterogeneous ways of characterizing the four quality dimensions.

The model, N_1 (See figure 2.3), is characterized as a good, compared to the other three models. It has a balance between overfitting and underfitting, in addition to being simple and having a good fitness. The model, N_2 is characterized as overfitting, as it only illustrates the most frequent traces and allows us to see only certain sequences (*a, c, d, e*, *h*). More specific, it shows 1391- 455 = 936 traces that fits. The third model, N_3 is underfitting. This can be explained as the behavior within the event-logs seems to be very difficult to trace. On the other hand, this model is not overfitting, but simple and has a fitness which is characterized as good. The last model, N_4 shows only 4 out of 21 different traces. The model shown in figure 2.3 illustrates a highly complex and overfitting model, with lack of simplification and a minimal structure. Depending on a company's purpose and goal of a process discovery analysis, it is possible to balance and operationalize the four quality dimensions heterogenous (Aalst, 2016).


start
a
register request
b
examine thoroughly
c
examine casually
d
check ticket
e
decide
f
reinitiate request
g
pay compensation
h
reject request
end
N₁ : fitness = +, precision = +, generalization = +, simplicity = +
N₂ : fitness = -, precision = +, generalization = -, simplicity = +
N₃ : fitness = +, precision = -, generalization = +, simplicity = +
(all 21 variants seen in the log)
N₄ : fitness = +, precision = +, generalization = -, simplicity = -

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Figure 2.3: Four Alternative Models from the Same Event Log (Aalst, 2016)

2.2 Event Log Design

This section of the thesis will present the current theories and practical guidelines concerning the design of an event log for Process Mining. This will include the critical components and structural elements, put differently, what the event log should look like. Further on, the most important challenges regarding the construction of an event log will be introduced, before focusing especially on the concept of convergence and divergence. Finally, this section will conclude with the six guiding principles for extracting data as put forth by the IEEE *Task Force on Process Mining.*

2.2.1 The Event log

To be able to run Process Mining analysis on our data, we are dependent on creating an event log. These logs contain data of the events related to the specific process, that is the target for Process Mining analysis, and structured in a certain way. To better illustrate which step in the Process Mining workflow that relates to the event log, we present fig. 2.4 derived from Van der Aalst (2016). The workflow illustrates that we start off with raw data. The data could be gathered from numerous data sources, such as Excel spreadsheets, simple flat files or database tables, which is how data is stored in ERP systems like SAP. The extraction phase is where the raw data is transformed into structured logs that relates to specific events that occur within the process which is target for the Process Mining analysis.

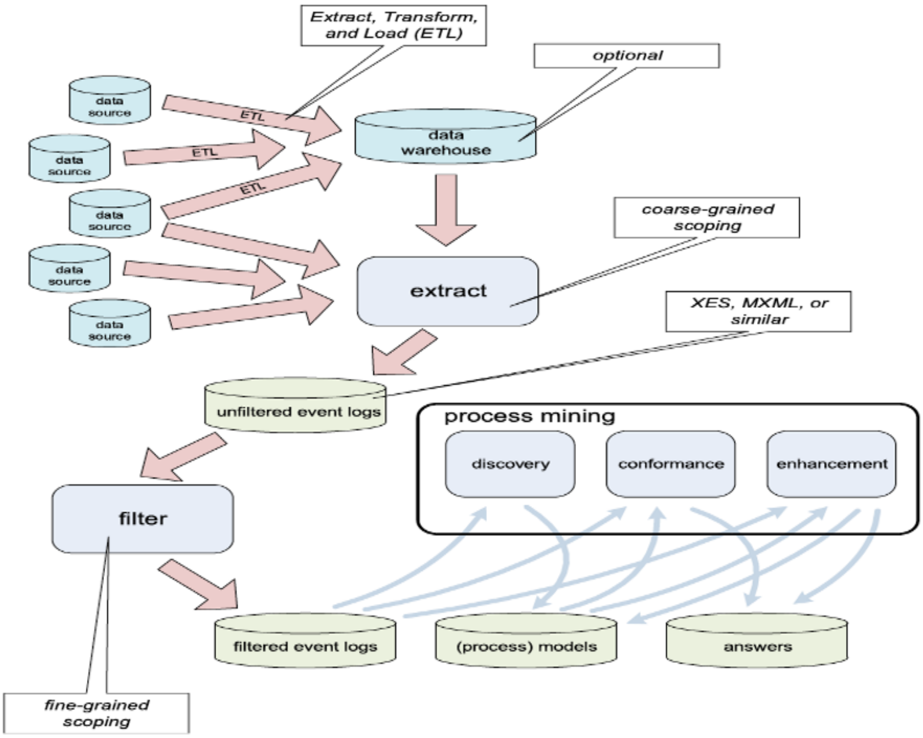
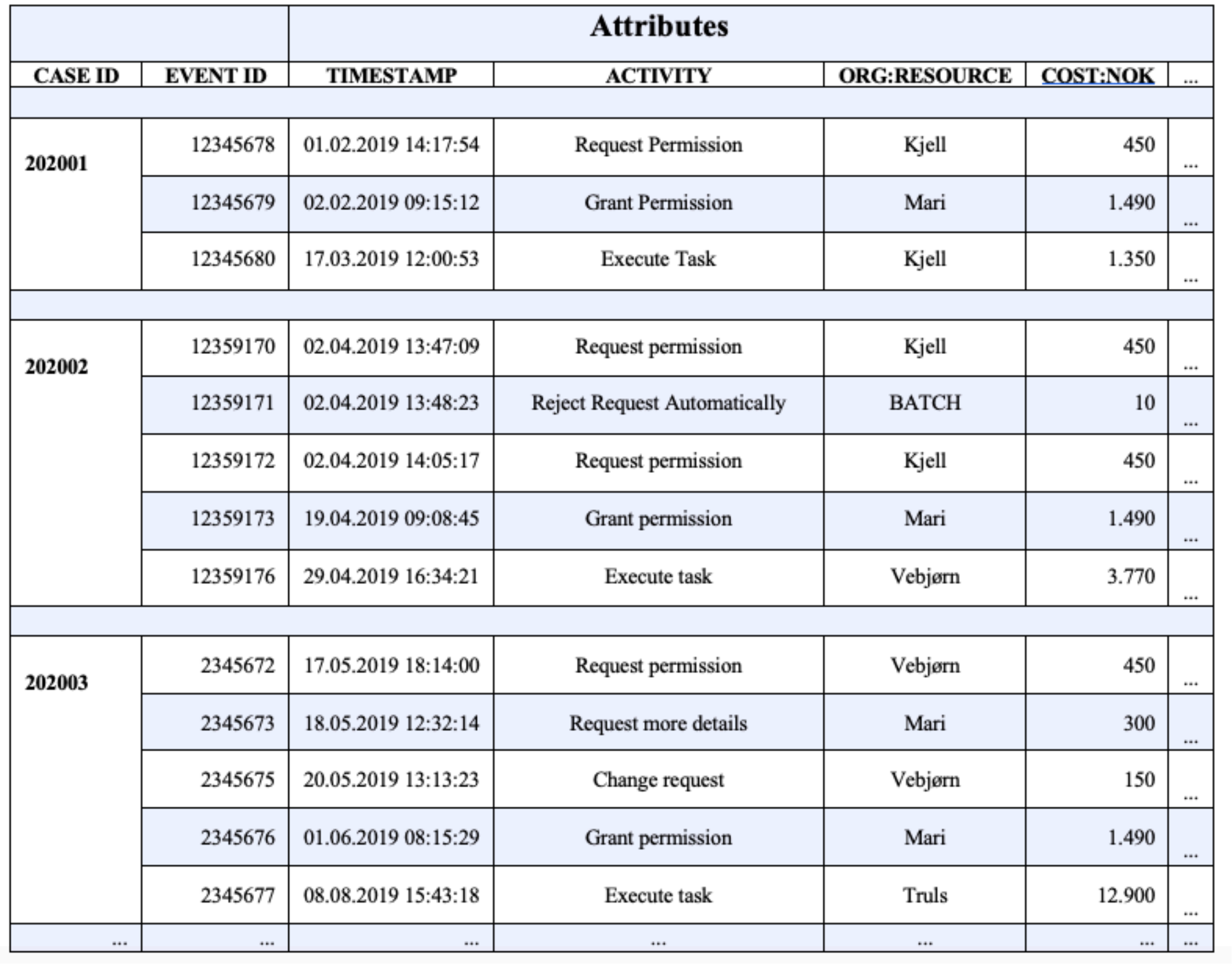


Figure 2.4: Process Mining WorkFlow – From raw Data to Results (Aalst, 2016)

In what is referred to in the figure as *Coarse-Grained Scoping*, the raw data is extracted from the source and transformed and structured into meaningful logs suitable for Process Mining and adapted to serve the objective of the analysis. An important bit to notice here, is that depending on the viewpoint and the questions in need of answers, different event logs may be extracted from the same data set (Aalst, 2016). In this, we find that the creation of an event log requires both extensive domain knowledge of the process to be analyzed, as well as a good apprehension of what data to extract. For instance, which ones of the thousands of SAP-tables holds data relevant to the Purchase-to-Pay, PTP or the Order-to-Cash, OTC process? In coarse-grained scoping, the aim is to make sure that all event data relates to a single process only, i.e. the process that is target for the subsequent analysis. The next paragraph will present the structural requirements of an event log, and subsequently, a fictive log will illustrate what a complete event log may look like.

The basic requirements of an event log are, primarily, that all the events should relate to the target process. Secondly, each event in the log has to refer to a single process instance, referred to as a *Case* (Aalst, 2016). A case will represent one execution of the target process, either partially or complete from start to finish. Cases could for instance be given a label with a number from *1-to-n,* or it could refer to a specific document number as long as it is unique. Thirdly, an event has to relate to an *Activity*, for instance, the creation of a document or adding a part to a product in a production line. In addition, the events within a case has to be ordered to be able to analyze causal dependencies in process models (Aalst, 2016). Each event inside the different cases are identified by a unique ID, meaning that each event can only relate to one case. The different events can, however, relate to the same type of an activity. As an example, the activity "Create Purchase Order" would typically be observed in most cases of a Purchase to Pay process, but it must be identified which instance it belongs to, hence the need for unique *Event ID*s.

Typically, an event log will also include one or more *attributes,* such as date and/or time related to each event in the form of a *Timestamp*, e.g. "24.12.2019 17:30:14". A timestamp will make it possible to analyze which activities that consume most time, or the waiting time between activities. Other common attributes are *organizational resource*, *Price/Cost*, *Customer* and/or *Vendor.* Below, we have included an illustration of a fictive event log that includes mandatory components and some additional attributes.



		Attributes				
CASE ID	**EVENT ID**	**TIMESTAMP**	**ACTIVITY**	**ORG:RESOURCE**	**COST:NOK**	...
202001	12345678	01.02.2019 14:17:54	Request Permission	Kjell	450	...
	12345679	02.02.2019 09:15:12	Grant Permission	Mari	1.490	...
	12345680	17.03.2019 12:00:53	Execute Task	Kjell	1.350	...
202002	12359170	02.04.2019 13:47:09	Request permission	Kjell	450	...
	12359171	02.04.2019 13:48:23	Reject Request Automatically	BATCH	10	...
	12359172	02.04.2019 14:05:17	Request permission	Kjell	450	...
	12359173	19.04.2019 09:08:45	Grant permission	Mari	1.490	...
	12359176	29.04.2019 16:34:21	Execute task	Vebjørn	3.770	...
202003	2345672	17.05.2019 18:14:00	Request permission	Vebjørn	450	...
	2345673	18.05.2019 12:32:14	Request more details	Mari	300	...
	2345675	20.05.2019 13:13:23	Change request	Vebjørn	150	...
	2345676	01.06.2019 08:15:29	Grant permission	Mari	1.490	...
	2345677	08.08.2019 15:43:18	Execute task	Truls	12.900	...
...	...	...	...	...	...	...

Table 2.5: Example of an Event Log where each line Represents an Event

2.2.2 Challenges

This section will describe what Van der Aalst (2016) hold as the five most important challenges related to extracting event logs.

C1: *Correlation*

This challenge relates to the requirement that events in an event log have to be grouped per case, i.e. event correlation. The core of the problem is that it is not straight forward to identify events and corresponding cases when event data can be scattered over several tables, as is the case with data from ERP-systems such as SAP.

C2: *Timestamps*

Though timestamps are not required to run Process Mining, the events have to be ordered per case. However, timestamps are typically what makes it possible to sort events, in order of occurrence when merging data from various sources. The core of this challenge has to do with the fact that different information systems store timestamps in different formats, from the coarser end of the scale with only a date, to the fine-grained timestamps which include milliseconds. It can be nearly impossible to reconstruct the order of events in a process, if multiple events happen on the same day and the information system only records date. In addition, delayed recordings and multiple clocks may pose a challenge, leading to an unreliable event log, e.g., an employee waiting until the end of the day to punch data into the system instead of doing it right away.

C3: *Snapshots*

This challenge refers to the problem that some cases have started before the recording period, whilst others are still running after the recording period has ended. This is usually solved by filtering out unfinished cases, given that the initial and final activities are known. If, however, the average duration of a case is close to the span of the recording, it is harder to discover end-to-end processes.

C4: *Scoping*

Determining the scope of an event log poses the fourth problem. ERP-systems, such as SAP may potentially hold thousands of tables of business data. The tables that are needed will depend on the questions that one seeks answers to, as well as the available data. Scoping and deciding which relevant tables to extract requires domain knowledge.

C5: *Granularity*
The final challenge has to do with the granularity of the event log, i.e. how detailed the activities are specified. Some logs may contain low-level events that are too detailed to serve the purpose of the end-users. As an example, the creation of a sales order in SAP that may have several individual order lines can be seen as one activity by the end user, although it in fact comprise of a number of activities, namely the creation of each order item. As we may derive from this, it is possible to abstract low-level activities into higher level abstractions.

2.2.3 Convergence and Divergence

In addition to the five main challenges presented above, there are several other problems that may occur in the process of extracting event logs. Two of these common errors are referred to as *Convergence* and *Divergence* of the event log, which is tightly connected to *C1*, correlation and *C5*, granularity. At the center of the problem regarding convergence and divergence, lies the fact that there may not always be a straightforward one-to-one relationship between documents and events in reality. This is certainly the case for ERP-systems like SAP. We may observe both *one-to-many* relationships as well as *many-to-one* relationships.

When an event in the event log is related to many cases, we get the effect named convergence (Aalst, 2019). In Process Mining, this occurrence will typically look as if the same activity was performed on several cases, but in reality, it was an event performed just once (Selig, 2017). To exemplify using SAP, one billing document may comprise of several sales orders, creating an illusion that a billing document was created for each distinct sales order, although it was in fact created only once in the ERP system.

The other effect, divergence, gives name to the situation where there are numerous instances of the same activity within a single case (Aalst, 2019). This effect may be visual in the form of loops in a Process Mining analysis, even though they are in fact related to different documents. By once again using SAP as an example, we may observe the activity "Create

Sales Order" multiple times for the same case, while in reality all the occurrences refer to the creation of individual order lines within the case itself.

As we may find, the concept of convergence and divergence is not easily migrated, but it can be adjusted by deciding on higher or lower levels of abstraction, i.e. the granulation of the events. For instance, it is possible to use the individual order line item as the case identifier, equally it is possible to aggregate all the order line items into a single sales order and use this as the case id.

The next chapter will introduce the guiding principles to overcome obvious mistakes easily made when applying Process Mining in real-life settings, focusing especially on data extraction as proposed, by leading scholars and practitioners (Aalst et al., 2012) of the Process Mining field.

2.2.4 Guiding Principles for event log extraction

We conclude this section on event log design by presenting the six guiding principles to process Mining, as stated by the "The Process Mining Manifesto" (Aalst et al., 2012) of the IEEE Task Force on Process Mining. These principles provide guidelines that have reached consensus throughout the Process Mining community. As this portion of the thesis is devoted to event log design, we will focus on the three principles that relates to event logs, only briefly touching the other three guidelines. The propositions will assist us in revealing the most important steps involved in event log construction, as they are prerequisites for ensuring sufficient quality of the final log, and thus also increasing the subsequent model quality.
The six guiding principles are as follows:

GP1: Event Data Should Be Treated as First-Class Citizens
GP2: Log Extraction Should Be Driven by Questions
GP3: Concurrency, Choice and Other Basic Control-Flow Constructs Should be Supported
GP4: Events Should Be Related to Model Elements
GP5: Models Should Be Treated as Purposeful Abstractions of Reality
GP6: Process Mining Should Be a Continuous Process

The only part of GP3 that holds any relevance to log extraction is dealing with concurrency, which indicates the need for timestamps. The rest relates to requirements of the mining tool. GP5 is unrelated to event log design, as its focus is on interpretation of results. GP6 holds some relevance to event logs, but as our thesis focuses on the use of post-mortem data only as opposed to pre-mortem data, it is considered less important for our work at this point.

GP1 concerns the event data quality, and states that among other criteria, it should be trustworthy and complete. The point is to treat the event data not just as a by-product but prioritize and ensure that it is of the best quality as possible. It should be safe to assume that the events recorded in the log actually has occurred, and give the full picture, i.e. not leave certain parts of the process out.

To be able to extract meaningful event data to form an event log, it is crucial that one or more concrete questions are in focus, which is at the core of GP2. If not, it would become nearly impossible to know what data is relevant when thinking of for instance the thousands of tables in an ERP system, such as SAP.

GP4 relates to the event log in several ways. First of all, it must be made certain that the events recorded do belong to process instances, i.e. *event correlation*. Secondly, an event can belong to one activity only. Thirdly, the granularity of the log should match the granularity of the process model or it would not make sense. Finally, as pointed out by Selig (2017), the events must refer to business activities and not technical activities performed for instance in the SAP system.

2.3 Event Log Extraction from SAP

This section will provide operational support to guide us in collecting relevant information about SAP. Moreover, we will construct an event log, based on Piessens (2011) contribution to the field of extracting an event log from SAP. By using information from this research, we will be able to have more insight of SAP's structure, in order to answer the first research sub-question.

SAP ERP provides a set of best practices, which firms can use as a reference model, in order to shape their own processes (Piessens, 2011). The “Purchase to Pay (PTP)” and “Order to Cash (OTC)” are processes focusing on the entire chain in a typical procurement cycle and a typical sales process with a customer. As the OTC represents the entire process of sales and invoice to customers, the Purchase to Pay- process refers to purchase and payment of invoices from suppliers. A customer’s return of a good or a service begins each time at the OTC-process. Therefore, it is possible to use this process to find all the return-orders within the entire order processing system. The business process covers typical insight in various activities, such as:

- Quotation
- Creating the Sales Order
- Packing Item
- Picking up Goods
- Confirm Delivery
- Return of Goods
- Closed Payment
- Open Payment
- Change Table Activities

The research of Piessens (2011) states 27 OTC- activities, considering extracting data from SAP. Moreover, these activities are recorded in specific tables where some tables overlap with the Purchase to Pay process, and are characterized as:

Order to Cash Tables								
CDHDR	LTAP	MSEG	VBAP	VBEP	VBUK	VBRK	VTTK	LIKP
MKPF	VBAK	VBFA	VBUP	VBRP	VTTP	LIPS	CDPOS	LTAK

Figure 2.6: Order to Cash table Characteristics. Inspired by Piessens (2011)

These tables are used to collect information about activities within *Sales Orders* and *Goods Movements*; in the SAP system’s SD (Sales and Distribution) and WM (Warehouse Management) modules. Piessens (2011) presents an overview of relevant activities considering the OTC- activities, and are characterized as:

Order to Cash Activities	
Create Sales Inquiry	Change Sales Inquiry
Create Sales Quotation	Change Sales Quotation
Create Standard Sales Order	Change Standard Sales Order
Post Goods Issue	Create Outbound Delivery (TO)
Create Shipment	Change Shipment
Confirm Delivery	Cancel Transfer Order
Packing	Goods Movement
Goods Movement (Documentation)	Billing the Sales Order
Change Billing Document	Invoice Cancelation
Intercompany Invoice	Pro Forma Invoice
Returns	Debit Memo
Debit Memo Request	Create Purchase Order
Create Contract	Credit Memo Request
Returns Delivery for Order	

Table 2.7: Order to Cash Activities. Source: Piessens (2011, p. 91)

The *Change Tables* have most records, comparing to the other OTC-tables, as a result of containing all the changes stored in the information system. These are important for the extraction of event logs. Moreover, extracting an event log is stated to be a crucial step in the Process Mining project (Piessens, 2011). One of several important steps, is to determine the purpose of the project, in order to make it easier for the analysis, and avoid problems later. There are especially five important things to know, for creation of an event log: (1) Relevant Activities, (2) How to recognize occurrences of activities, (3) Relevant attributes, (4) The cases that determines the scopes and (5) The output format that leads to an event log.

An *occurrence of an activity* refers indirectly to an event. In the context of Process Mining, this means that we need to answer: what and when did the activity occur, and who was involved? By using SQL-queries, it is possible to select the information that affects the case. The output format can be determined by a process analysis tool. Another important thing is to check whether each of the determined activities can be mapped to an artefact. For the Purchase to Pay process, the following artefacts would be identified:

1) Purchase Requisition
2) Delivery
3) Invoice
4) Payment

3 METHODOLOGY

The methodology chapter will show what we have done to increase the validity and reliability of the analysis and explain why. It is divided into two parts. First, we present database structure that the event-log is built upon. The second part considers methods of evaluation, in terms of validity. In this part, we also discuss the characteristics of our event log that is important to take into considerations for evaluating the quality of the discovered model.

3.1 Database structure

This section of the thesis is concerned with the process of extracting data from SAP, and the structure of the extracted data. It is essential to gain an understanding of the data at hand, before commencing the task of constructing event logs, needed to perform analysis with Process Mining.

We will start off by describing the structure of the raw data provided to us before defining the relevant tables for our analysis. A brief presentation of the way in which data is connected through the concept of primary and foreign keys will directly follow. These are essential concepts for our work on this project, as data from numerous tables have to be connected in order to construct the event log. Lastly, chapter 3.1.2 provides a general overview of the Return Order Process that has served as a reference for building the event log in the analysis and results section of this thesis.

3.1.1 Relational Databases, Tables and Transactions

The case-company's available dataset is an extraction of their SAP ERP-system, with entries confined between 01.01.2019 and 31.10.2019. The extraction contains data from most modules and areas within the ERP system, e.g. Financial Accounting, Materials Management and Sales & Distribution. Some lookup tables are missing or not included in the data extraction, meaning that a lot of research effort have to be put into making sense of the table contents. In other words, the tables themselves do not provide interpretations of their content, so secondary sources will have to be consulted in order to make the data understandable. The main challenge, however, is not with regards to missing tables, but rather the opposite, namely the sheer size of the dataset and the vast number of tables. Our database consists of 233 tables, and the total size of the extraction is 14,5 gigabytes. The data is saved as .txt-files and accessed through a dedicated server which contains a duplicate of the original data extraction that was made for auditing purposes.

The standard SAP tables contain information related to specific documents, transactions, resources and such, but the task of extracting event logs from them is still not trivial. Each table has its own specific codename, consisting of letters and numbers that does not make much sense without the right degree of domain knowledge. Luckily, SAP being one of the most widely used ERP systems around, there exists official online resources, such as table dictionaries and best practice guides for common processes. As well as informal communities dedicated to knowledge transfer between professionals and users. These are all helpful in gaining the sufficient degree of knowledge about the most common, standard SAP tables and their interconnections.
Though some studies, like *de Murillas, Reijers & van der Aalst* (2018) have been aimed at automating the extraction of event logs from database systems, the most general is to obtain events manually from the tables in the database. Next, we will look at the most relevant tables containing information essential to our analysis including their basic structure.

The most important standard SAP tables that contain information we need to reconstruct the return order process flow, is found within the Sales and Distribution area. These are related to the Sales Order Document, the Shipping/Delivery Document and the Billing Document.

Information concerning the Sales Orders are stored in the VBAK and VBAP tables, where the former is the header data at the sales order abstraction level, whereas the latter contains entries at the sales order line level, i.e. information about heterogeneous items in a specific sales order.

As for the Delivery Document, we need the LIKP and LIPS tables, which are the header table and the item table, respectively. Information about the Billing Document is found in the header table VBRK and the adjoining item data table, VBRP. In addition, there are a few other critical tables we have to identify in order to construct the return order event log, especially the VBFA and the MKPF tables.

The VBFA table contains the sales document flow, which is essential for this thesis, as it stores data about documents following each other. These documents contain certain activities that are essential components of the return-order process. As for the MKPF table, this stores header data about the material document.

Lastly, the CDHDR and CDPOS tables hold some importance. These are significant because they store most creation and change events on documents, where CDHDR is the Change Document Header, and CDPOS is the Change Document Position. These two tables can be a bit challenging to interpret because each transaction is represented by codes. Luckily, the SAP tables TSTC and TSTCT contain a description of each transaction, and is therefore, crucial in providing business understanding to the somewhat cryptic transaction codes. However, we note that the change document tables and the sales document flow table overlap in the sense that the same activity can be extracted from both CDPOS and VBFA.
It is expected that some of the activities of the OTC process will occur in the return order process, as returns are a special case that is initiated through the same leading document as the OTC process, namely the sales order document. Some activities that are linked with an Order-to-cash process is presented in table 3.1, which have been derived from the TSTCT table in SAP.

TCODE	**TTEXT**	**TCODE**	**TTEXT**
VA11	Create Sales Inquiry	VA12	Change Sales Inquiry
VA21	Create Sales Quotation	VA22	Change Sales Quotation
VA01	Create Standard Sales Order	Vl01N	Create Outbound Delivery (TO)
VT01N	Create Shipment	VT02N	Change Shipment
04H1	Confirm Delivery	LT15	Cancel Transfer Order
Pl00	Packing	VF01	Change Billing Document
MIGO	Goods Movement	MR8M	Invoice Cancelation
MBRL	Returns	FPY1	Debit Memo
VL01N	Returns Delivery for Order	FD32	Credit Memo Request

Table 3.1: Example of Activities often Observed in an Order to Cash process

Next, we will look at how the tables can be connected to each other in order to provide us with the chain of events needed for the construction of the event log and subsequent analysis.

A database is comprised of columns and rows, where columns refer to attributes whilst rows contain entries. Each entry in a table is identified by a unique identifier, called a primary key, usually a in the form of a single numerical field or a combination of several numerical fields within the table. In a relational database, the tables are connected via foreign keys, which are references to primary keys in other tables. These relationships make it possible for the staff working with the database to save both time and space because it is not necessary to enter every detail related to for instance the relevant vendor or customer when creating a new sales order. By storing some data in master tables, redundancy is thereby avoided. The different tables contained in the database from A-Store's ERP-extraction is connected in the same way. Discovering these connections are key to our work on this thesis to construct the event-log.

The tables we have presented, each contain a set of keys, making it possible to identify unique entries. As an example, the VBAK table has a primary key consisting of two columns; MANDT, which is the client number and VBELN, which is the Sales Order Document number. The other header tables have a similar structure, whereas the item level tables also have a third component that makes it possible to uniquely identify each item that makes up an entire order. In the table VBAP, the Sales order item table, the third column of the identifier has the name POSNR. Identifiers like these, may act as foreign keys in other tables, which makes it possible to connect data through this reference, for instance a specific delivery item to a corresponding order item. In order to make searching through tables for relevant data easier, we made several lookup tables containing SAP codes and their corresponding descriptions, e.g. warehouse movement types.

3.1.2 The structure of the return order process

This chapter will give a brief explanation of how the return order process is performed within the SAP Sales and Distribution module. This will serve as a base reference for the thesis, as it reveals some critical activities to look for in the dataset.

The return order process is initiated when a customer informs the company that they want to return one, several or all items of an order. The reasons for return could be many, including wrong item(s), damaged goods or other complaints.

The company's first step is to create a return sales order, where data of the returned item(s) are stored. The next step is for logistical staff to create the return delivery, where the goods are returned from the customer and sent back to a shipping point that belongs to the company. When the return order arrives at the shipping point, details of the goods movement will then be documented by staff receiving the return delivery. There may be several types of goods movement, e.g. moving the goods to scrap if damaged, moving the goods to unrestricted stock for re-use or moving it to blocked stock if not acceptable for re-use. After the returned goods have been handled and moved to the designated warehouse area, it may be returned to the vendor, if for instance, the return order is related to the quality of the product.

The final step in the return order process has to do with billing. A decision is made of whether or not the return is justified to be followed up as a credit memo. If so, a billing block will be lifted, in order to make a credit memo to the customer account, based on the sales return. Put plainly, the customer gets money back for the returned unit, or if a reference invoice of the original order exists, the invoice is reduced accordingly. Figure 3.2 below illustrates the return order process flow as performed in SAP SD.

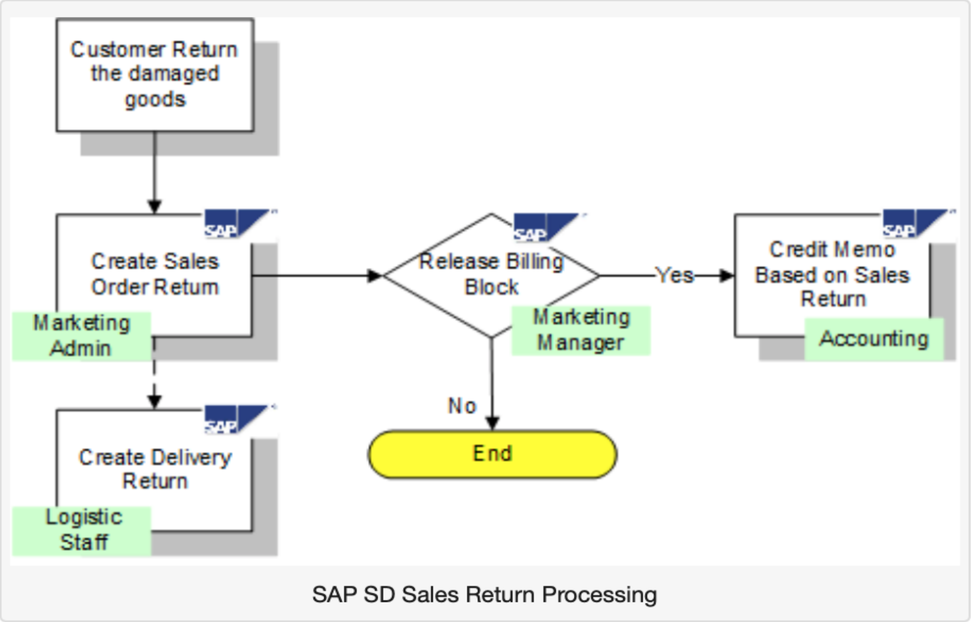

Customer Return the damaged goods
Create Sales Order Return
Marketing Admin
Create Delivery Return
Logistic Staff
Release Billing Block
Marketing Manager
Yes
No
End
Credit Memo Based on Sales Return
Accounting
SAP SD Sales Return Processing

Figure 3.2: Return order Process Flow. Illustration by Gea (2018)

3.2 Evaluation and Validation

In this section, we start off by presenting and then, discussing the underlying process that can be adequately described by using a WorkFlow Net in context of our discovered model, cf. section 3.2.3 in Van der Aalst, 2016 for more information regarding WorkFlow Nets. As we are modeling business processes in terms of a net with a dedicated source place, a WorkFlow net shows all nodes on a path, from source, i.e. process start, to sink, i.e. a place where the process ends. Secondly, we will consider our event-log upon a representative sample of behavior. In addition, we will show how to minimize noise and incompleteness, to discover a suitable process model.

3.2.1 Representational Bias

The representational bias helps us to limiting the search space of possible candidate models. This can make discovery algorithms in itself more efficient. For instance, our event log is assumed to be similar to the underlying process of Figure 3.3 (b). This figure shows a discovered model produced by the α-algorithm; due to the representational bias, the algorithm is destined to use two α transitions and no τ labels, cf. section six in Aalst, 2016 for an in-depth explanation of the parameters. The τ transitions are sensible and not recorded in the event-log, as the algorithm would have problems reconstructing their behavior if it was present. When illustrating the underlying processes to be described by a WF-net, we assume that the underlying process can be explained by a WorkFlow-net where each transition has a unique and visible label. Ideally, one would like to discover a WorkFlow-model as figure 3.3 (c), by reproducing the trace (a, b, c) and not (a, c). In this case, the model is produced by the α-algorithm. We can see that the algorithm is destined to fail for that log, because of the representational bias.

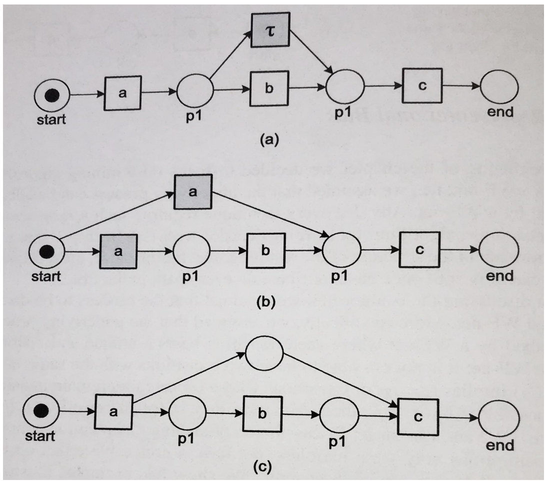


Figure 3.3: Three WorkFlow-nets for an Event Log (Aalst, 2016)

3.2.2 Noise and Incompleteness

To discover a suitable process model, it is assumed that the event-log has representative sample of behavior. By this, we mean that there are two related phenomena that make our event log less representative for the return-process being studied:

- *Noise* refers to rare and infrequent behavior in the event log, that is not representative for the typical behavior of the discovered process.
- *Incompleteness* refers to an event log containing too few events, such as activities, cases and timestamps to be discovered in the underlying control-flow structure.

Seen in context of Process Mining, we assume that the information within the event log are the most frequent events that reflects what really happens in the return process. Looking at it from one perspective, it is quite certain that the reliability is consistent. We expect the same events and throughput times if we repeat the measurements and adjust the event log because it does not describe rare behavior. However, whereas noise refers to the problem of having "too much data", completeness illustrates the problem of having "too little data". The α-algorithm uses under five activities and corresponds to approximately 147.705 fully processed cases, out of approximately 153 800 cases, which are assumed to be relatively weak notions of completeness to avoid this problem. Given that we have constructed tables and merged tables based on online-information and limited knowledge about the client's real-process, it is unrealistic to assume that every single trace is presented in the event log. To show the relevance of completeness to our log, consider the process consisting of five activities that correspond to a log that contains information about 147.705 cases. Seen from another perspective, the total number of possible interleavings in the model with five activities in the model, is 5! = 120. Hence, it is quite realistic that each interleaving is present in a log, as there are more cases (147.705) than potential traces (120). Even though if there are 120 traces in the log, it is extremely unlikely that all possible variations are present, due to poor data quality.

4 ANALYSIS and interpretation of RESULTS

4.1 Extracting Data from SAP

The extracted SAP tables have unstructured raw data in their information systems. Therefore, we start off by using an analytical tool, *Microsoft SQL Server Tool 18* to connect certain tables with each other, to track and trace return-orders. Given this basic requirement (Aalst, 2016), it is a demanding need to construct new tables in the database, to extract value from tables, due to missing lookup tables. By relating existing and new tables together, it enables us to structure and transform raw data into meaningful logs for Process Mining, also known as *coarse-grained scoping*. Anyhow, when it comes to modeling the event log, this master thesis has chosen to exclude the details about the SQL-query steps as they are purely technical. They also do not provide any valuable information regarding why these particular tables are suitable data for constructing an event log.

Note that this master thesis uses online information, e.g. *leanx* and *support.sap.com,* to quickly increase the SAP knowledge on commonly available information. This helps us to accumulate our domain knowledge of the company's process and the extracted data. In addition, it provides us an increasingly understanding of the data quality problems we are dealing with, in our dataset. Unfortunately, we are dealing with missing metadata that makes it the construction of the event-log harder. However, we solve this by answering ongoing questions about the return-process. This enables us to decide the number of tables to extract and construct. Further on, please note that all the return-orders are related to the OTC-process, as all event data needs to be related to a single process only. In other words, the source of the extracted data comes from the *Sales and Distribution (SD) module*, the *Financial Accounting (FI)* module and the *Materials Management (MM*). These modules concern processes within logistics.

4.2 From Raw Data to an Event Log

4.2.1 Preparation Phase

This part of the analysis will use certain SAP tables that provide columns with valuable information. We use a preparation phase to collect all the SAP specific details needed to extract an event log containing relevant information. The most significant columns from several tables are merged into a new constructed spreadsheet in Excel, by using the *Internet* to extract information about actual activities of the return-process. We are mainly dealing with challenges related to transforming unstructured logs to meaningful logs, suitable for Process Mining. Moreover, the steps are driven by the questions:

(1) How are the available SAP-tables connected to each other, in order to follow the return-orders from start to end?

(2) Are we dealing with extracted data that can map out the return-process, that can provide valuable information about observed activities? If yes/no, what is the next step to solve barriers?

By answering the above questions, we are able to develop a step-by-step guide to construct an event log in the context of the SCM process of return orders in SAP, aimed at Process Mining. For instance, most of the timestamps determines the position in a process flow. Moreover, finding the relevant columns that indicates the activities for a specific case, will increase the precision in the Process-Discovery.

Step 1: *Constructing CaseID*

Our approach to select a suitable label for each unique process instance, i.e. case, is based on detecting what can be referred to as the *leading document* in the flow of documents in SAP.

As a starting point, we knew that the first document to be created after a customer notifies the company of a desire to return one or more items, is the sales order document. This document is found in the VBAK table in SAP, which contains the header data, and VBAP table which contains the item level data. An employee will create this document first, and it will serve as a reference for subsequent documents in the return order process flow. Based on this, we selected the document number of the sales order document as the most important component of our CaseID. This number can be extracted from the VBELN column in the VBAK/VBAP tables. As these tables contain details concerning all sales orders and their associated line items, it is important to filter out the cases that do not refer to return orders. This is done by only extracting the rows in VBAK where the document type is labeled "Return". In SAP, this involves selecting all rows where the column VBTYP has the value "H", i.e. the document category is set to return.

As *GP2* of Van der Aalst et al. (2012) implies, it is important to choose the type of cases to be analyzed before applying a Process Mining technique. When constructing the CaseID for this project, the lifecycle of the individual order lines was considered to be of particular interest within the return process. To be able to analyze the return order process at the item level, the order document number is not sufficient on its own. Each return order may consist of more than one item, and this eventuality calls for a way of identifying each of these order lines. In the SAP table VBAP, each line item of the sales orders is identified by the column POSNR. On its own, this column is not unique, but if combined with the VBELN column it is possible to label each returned item correctly. On one hand, including the POSNR in the CaseID will migrate divergence by reducing the possibility of rework loops as suggested in section 2.2.3 of the theory section. On the other hand, this will lead to convergence because some activities performed only once at the return order abstraction level, is now duplicated. As may be observed, there is no "free lunch" when balancing convergence and divergence. However, divergence is preferable over convergence in the context of return orders, to facilitate constructing an event log where analyzing the lifecycle of return items should be possible.

The final piece in the CaseID is the column MANDT. This is the client field, and always the first field in every database table that contain application data. The combination of MANDT, VBELN and POSNR is observed both in some change tables and as part of the primary key in VBAP, making this a logical CaseID for the event log.

Structure of the CaseID: ***MANDT*** *+* ***VBELN*** *+* ***POSNR***

To summarize, this step towards a complete event log has involved the construction of a CASEID suitable to correctly identify each unique process instance of the return process from customer in SAP. This sorting column is made from the combination of the three columns MANDT, VBELN and POSNR, which all can be found in the VBAP table.

Step 2: *Constructing Activities*
We used the column *VBTYP* in the table VBAK to find all the document types, e.g., C, H, K and L, related to return-orders, to create an event log with relevant return-orders activities. In addition, we constructed a table called *VBTYP_TEXT*, to transform the VBTYP-values into meaningful information by using common internet information about SAP. Moreover, the first activity *Create Sales Order Returns* was constructed, by adding a description of this in the existing table. Thereby, we could immediately count the numbers of return-orders. Note that the analysis was applied upon duplicated tables, in order to keep the originals untouched.

Further on, we used the approach of utilizing VBTYP to classify different types of documents in order to create activities. Given that the tables VBAP, LIKP, LIPS, VBRP and VBRK had the column VBTYP, we would trace and track different events with heterogeneous timestamps. The timestamp was used as an indicator to determine whether the activities had occurred at a unique point in time.

Like the theory, we believed that it was safe to assume that the events recorded in the tables actually occurred and gave the full picture. Therefore, we moved on to add the description of the activities, recorded in these tables. Now, there were another four activities in the log, known as *Goods Movement GD returns Unrestricted*, *Returns Delivery for Order*, *Cancel Goods Issue* and *Credit Memo*. Unlike the theory, we did not assume that the events recorded gave the full picture, after all these activities were visualized from above tables. More specifically, the constructed lookup tables did not provide any valuable information related to the activity *Release Billing Block.* This activity is considered an essential part of the return-process, in the context of supply-chain-management process set upon the structure of a typical return-order process in SAP, cf. section 3.1.2.

Step 3: *Constructing Timestamp*
The construction of timestamps for each activity in the event log is for the most part based on two essential columns that can be found in VBAK, VBAP and VBFA under the name ERDAT and ERZET.

In VBAK and VBAP, the ERDAT column refers to the date on which the document was created, i.e. the first document in the return order process flow. ERZET on the other hand, refers to the specific time of creation. There is a slight difference between the time recorded in VBAK and that recorded in VBAP, since the former refers to the creation of the order document, while the latter is the time the individual order lines where generated. For all practical considerations, the difference between these two times are so small that it does not matter much which of the two is extracted. However, we chose the time attached to the different line items, since it most correctly represents the process at an item level cf. the CaseID. In the VBFA table, we can equally extract the time and date from the ERDAT and ERZET columns. In this case they represent the date and time of when the subsequent documents in the sales document flow were created. As presented in section 2.2.2 under *C2* regarding timestamps, it is challenging to ensure that the data format and level of granularity is suitable for Process Mining. To attack this issue, we made sure that the occurrence of each event was pinpointed down to the second, thereby facilitating chronological ordering of events on the same day. The next paragraph shows how we tackled the formatting, which made it possible to deal with concurrency like *GP3* of section 2.2.3 implies is essential in order to construct an event log.

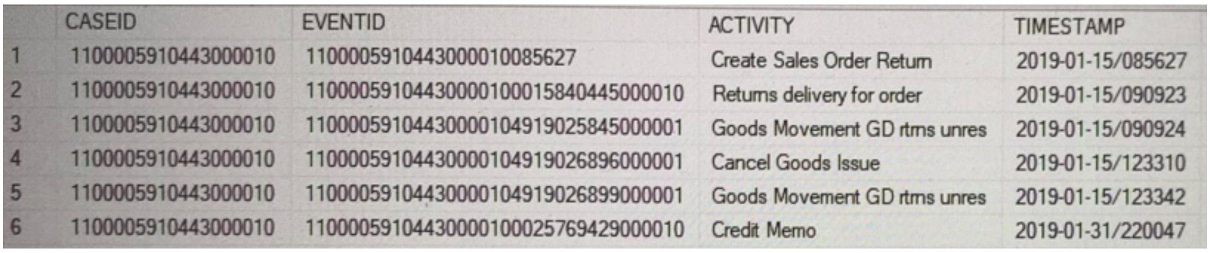
To make sure that the timestamp fulfills the requirements of the Process Mining tool, the two columns have to be combined into one that holds information about both date and time. To get to this structure, we had to convert the column ERZET from its original data type format of string, i.e. text, into the required format of date. During the datatype conversion, we also added colon-separators between the hour and minute positions and between the minute and second positions. The result is a time-column in the ***HH:mm:SS*** -format.
Having formatted the ERZET column, it was possible to combine it with ERDAT into a proper timestamp column with the following structure:

Timestamp format: ***YYYY.mm.DD HH:mm:SS***

In short, this step towards a complete event log has involved the construction of a timestamp column of a suitable data format, making it possible to sort activities in their correct order. The timestamps are a combination of the ERDAT and ERZET columns found in the VBAK/VBAP and VBFA tables in SAP.

Step 4: *Constructing an EventID*

The creation of an EventID is a crucial step of the return order process. Since the cases are identified by a unique ID, we also have to identify which events belong to which cases, cf. Van der Aalst, 2016 (pp. 125-162). The table VBFA, known as the Sales Document Flow, has four different columns that have been merged into one column, named EventID. The four columns are known as MANDT, VBELV, POSNV, VBELN and POSNN.



	CASEID	EVENTID	ACTIVITY	TIMESTAMP
1	1100005910443000010	1100005910443000010085627	Create Sales Order Return	2019-01-15/085627
2	1100005910443000010	11000059104430000100015840445000010	Returns delivery for order	2019-01-15/090923
3	1100005910443000010	11000059104430000104919025845000001	Goods Movement GD rtrns unres	2019-01-15/090924
4	1100005910443000010	11000059104430000104919026896000001	Cancel Goods Issue	2019-01-15/123310
5	1100005910443000010	11000059104430000104919026899000001	Goods Movement GD rtrns unres	2019-01-15/123342
6	1100005910443000010	11000059104430000100025769429000010	Credit Memo	2019-01-31/220047

Figure 4.1: The Constructed EventID as a result of merging four columns from VBFA

The particular columns are a combination of client code, MANDT, originating document number, VBELV and its specific position, POSNV, plus the document number, VBELN and its position, POSNN. In addition, the first activity is constructed by adding the column ERZET, which indicates the time. We use different values to construct the EventID, as a result of having different unique keys available in the tables. Like the theory, a combination of these columns sets us in a position to avoid divergence, because we limit the number of the same activity to the same CaseID. Looking from a descriptive approach, the reality shows that there is only one particular activity that is created upon a case. In addition, the tables VBAK and VBAP needs to be merged, in order to extract sales document position, which we need to create the unique ID. In other words, we connected VBAK and VBAP with LIPS and LIKP, before linking them up with VBRP and VBRK, see details in section 4.2.3, figure 4.2. Moreover, it is possible to trace and track events, such as activities and timestamp when the unique EventID is constructed, in addition to adding other attributes that can provide relevant information.

4.2.2 Construction results overview

This section of the thesis has presented our approach to construct the most important identifiers and attributes of the event log, namely the CaseID, activities, timestamps and EventID. The resulting CaseID is composed of the client code, document and item numbers of the leading document, i.e. the sales order document. This construction mainly solves the issue of linking each event to a single process instance, referring to one of the requirements of the event log presented in section 2.2.1, but also sets the granularity at order line level.

The activities, essential to discovering the *de facto* process model for the return process, was constructed by labeling the category codes of the documents in the SAP document flow.

Timestamps were constructed to ensure correct chronological ordering of events. They were made by combining separate time and date columns into the suitable format YYYY-mm-DD HH:mm:SS.

Finally, the EventID was put together using several original SAP columns, making it possible to identify each unique entry in the event log, and limiting divergence that could otherwise appear as rework-loops in the subsequent Process Mining analysis.

4.2.3 Extraction Phase

The preceding section has covered the crucial steps involved in the preparation phase of the event log extraction. This section will turn the focus over to the extraction phase, where each of the required and optional attributes are sorted in a table where each entry corresponds to a unique event. The decision of which additional attributes to include is driven by the scope of our work, namely developing a guideline to extracting an event log of the return order process from customers for use with Process Mining analysis. In this context, there are a few attributes that if included in the event log, may improve the overall quality of the subsequent

Process Mining analysis. A selection of such attributes will be the topic of step number six, which we have called *Enriching Logs*.

Before we explore this step, we will spend time on explaining our approach to extract the event data into an event log. Unlike the steps in section 4.2.1, this fifth step in our guidelines will be of a more elevated character, focusing on how the data is extracted and put together to form the event log.

Step 5: *Extracting Event Data*
The steps taken to construct the essential components of the event log during the preparation phase now has to be combined into a meaningful input for process discovery analysis.

Using the *guiding principles for event log extraction* as a basis for our approach, we had to make sure that the events recorded in our log would be an accurate representation as possible of the real-life process. As seen in section 3.1.2, the SAP-system does have a template of how the return order process should be performed. We have also had to assume that the recorded events in SAP are trustworthy on the basis that even though they are scattered across numerous tables, the events are recorded automatically, and must be assumed to be correct. For instance, it should be safe to assume that a return order recorded by the system actually exists and equally that an actual return order is recorded into the system.
Even though the events in our data extraction should be seen as trustworthy, they are still unlikely to be complete in the sense that we cannot guarantee that every event related to the return process is captured in our log. This has to do with the fact that every related event may not be recorded by the ERP-system itself, but also the limitations posed by the enormous task of sifting through all related tables and not missing for instance a rarely performed activity. Like (Aalst et al., 2018, p. 7), neither we can guarantee the completeness of the log stemming from the ERP-system, nevertheless, we can rely on its correctness, i.e. we know that the recorded activities were performed in reality.

There are many questions that could potentially be answered using Process Mining. When deciding on the scope of this thesis in general, and in particular the event log extraction, we have had to be efficient when narrowing down the amount of data needed. This scoping process, however challenging, has been aided by focusing on the research questions we set out to answer during this project, as *GP2* in section 2.2.4 suggests. When setting out on the mission to reveal the crucial steps involved in constructing the event log for the return order process, we have had to emphasize the most important steps, knowing very well that for instance some infrequent activities may not be caught by our approach. As we set out to discover the most important steps, it was possible to minimize the number of tables needed, thus narrowing the scope of our extraction quite substantially. Figure 4.2 below illustrates the source of the event data extraction including their connections.

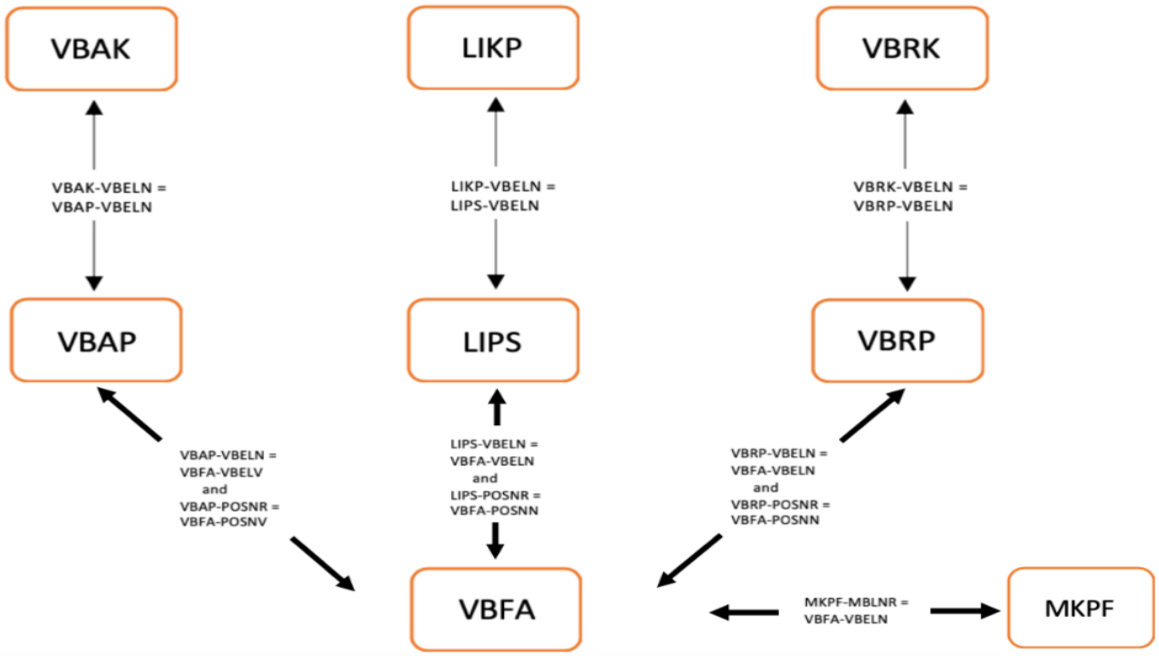

VBAK
LIKP
VBRK
VBAK-VBELN = VBAP-VBELN
LIKP-VBELN = LIPS-VBELN
VBRK-VBELN = VBRP-VBELN
VBAP
LIPS
VBRP
VBAP-VBELN = VBFA-VBELV and VBAP-POSNR = VBFA-POSNV
LIPS-VBELN = VBFA-VBELN and LIPS-POSNR = VBFA-POSNN
VBRP-VBELN = VBFA-VBELN and VBRP-POSNR = VBFA-POSNN
VBFA
MKPF-MBLNR = VBFA-VBELN
MKPF

Figure 4.2: SAP tables Target of the Event data Extraction.
Header tables on top, item tables below and the document history at the bottom.

When narrowing down the number of relevant tables for our extraction, we also had to consider what columns within the tables that contain the data we needed. The most important columns, i.e. those needed for the different constructions of the preparation phase is presented in table 4.3 below. The table also includes the columns that contain username connected to the different events. Though not mandatory in order to perform Process Mining, it may enrich the log, making it possible to for instance reveal which users are affiliated with which events.

Enriching of the event log will be the topic of the next, sixth and final step of event log construction.

VBFA	VBAK	VBAP	LIKP	LIPS	VBRK	VBRP	MKPF
*MANDT **	*MANDT **	*MANDT **	*MANDT **	*MANDT **	*MANDT **	*MANDT **	*MANDT **
*VBELV **	*VBELN **	*VBELN **	*VBELN **	*VBELN **	*VBELN **	*VBELN **	*MBLNR **
*POSNV **	VBTYP	*POSNR **		*POSNR **		*POSNR **	USNAM
*VBELN **		ERNAM		ERNAM		ERNAM	
*POSNN **		ERDAT					
*VBTYP_N **		ERZET					
VBTYP_V							
ERDAT							
ERZET							
BWART							

Table 4.3: *The essential columns (names) sorted under their respective table (names) * denotes primary key component*

Step 6: *Enriching Logs*

Enriching logs is all about adding meaningful information to the basic event data extracted in the preceding step. As *GP1* of event log extraction dictates, organizations should aim at constructing event logs at the highest possible level in order to benefit from Process Mining (Aalst et al., 2018, p. 7). Enabling the log to answer more questions by enriching it with more attributes is also in accordance with *GP2*, as presented in section 2.2.3 of this thesis. The close connection with these two *guiding principles* is the foundation of claiming this sixth step of the event log construction to be of high importance and thus part of this step-by-step guide.

The event data was extracted to enable the reconstruction of the process flow, to discover possible bottlenecks and compare throughput time between the different process instances. The information added to the log when enriching it, makes it possible to answer questions such as “What item is most frequently appearing in a return order?” or “Is the return process performed differently in department A as opposed to department B?”. These questions, along with several others, may be of particular interest to the end user. To be able to provide such detailed information to the event log, certain additional variables had to be extracted, which we will present in this, our sixth step of the event log construction.

In Step 5, we briefly introduced the columns containing usernames connected to the activities performed in the return process. We constructed a column in our final event log called *ORG:RESOURCE*, into which we extracted the information regarding who performed which activity in the process. For analytical purposes, this column is interesting because we are able to map out part of the "social network" in the target organization. It is possible to compare for instance the degree of efficiency between all users that perform the same tasks across the business. Information like this could lead to the company discovering that one or more employees have a more efficient way of performing their role, than company guidelines dictates, and may thus lead to new policies or instructions.

When attempting to analyze the return order process, several other attributes may be of interest. For obvious reasons, it is interesting to analyze the items being returned from the customer. More specific, answering questions such as "Are there specific items within a return order that contribute in delaying parts of the process?" or "Which items are most frequently returned by the customer?" are likely to be of value to the end user. We have therefore included the attribute *VARE* into our event log, which was extracted from the column ARKTX in table VBAP.

A third factor that may contribute to a richer analysis of the return order process is the customer itself. By extracting the customer number, identified by the column KUNNR in table VBAK, it is possible to analyze which customers return the most items, and whether or not the particular customer contributes to longer processing times. Because of this, we argue that including some customer data will enrich the final event log. It would be possible to also include addresses and names of customers by connecting to the customer master table, but as this raises privacy concerns, we have not included these data into our event log.

Finally, to analyze whether or not the return process is handled differently or systematically taking longer time to complete in certain departments, the attribute WERKS from the table VBAP should be included. This column denotes which department each return order has been processed in. It should be included to further increase the quality of the final event log.

This section has presented four optional attributes that is extracted in order to enrich the final event log by providing additional information. These attributes are related to the return orders and include the customer number, the plant/department number, the organizational resource and description of the return order items.

4.2.4 Results: An Event Log

Through the six previous steps we have gone from raw data extracted from SAP, to a complete event log aimed at modeling the return order process. The first four steps have involved the construction of important identifiers and properties of the event log, while the two last steps concerns knitting all the event data together to form the event log.

Step one in the preparation phase saw the construction of the CaseID, an identifier that deals with the challenges of granularity and correlation of the event log. The CaseID makes it possible to separate process instances from each other, grouping events per case. Our chosen granularity level for the final log is set at the item level, minimizing divergence.

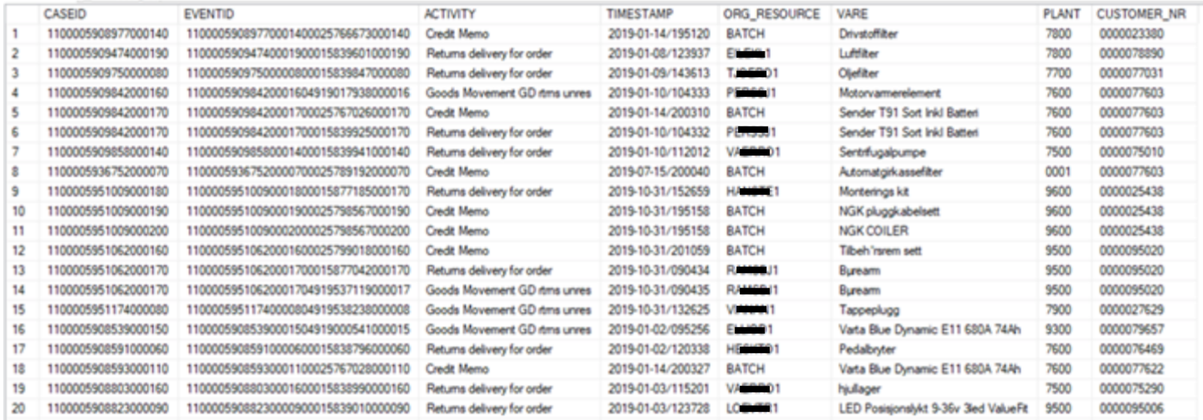
Step two concerned the construction of activities. This step gave name to the different tasks performed in the return process, from its creation to the issuing of a credit memo to the customer based on the returned items. The challenge of *scoping* was evident during this phase, dealing with vast amounts of SAP-tables that could potentially contain activities. Knowing very well that some activities were likely not captured in our log, we had to keep focusing on the most common activities performed in return order processing.

Step three was dedicated to the construction of timestamps. These timestamps are a way of ordering events in a chronological order, and crucial for calculating throughput time of the different cases in the log. This step also handled the challenge with timestamps, namely granularity and formatting.

Step four in the preparation phase dealt once again with correlation between events and cases, through the construction of EventID. This column in the event log ensures a proper connection between events and their respective cases.

Moving on to the fifth step, we illustrated which columns from which tables held the essential event data for the log construction. The connections between the tables were also illustrated, showing how the all the elements of the resulting log came together in order to reconstruct the return order process for Process Mining purposes.

The sixth and final step towards a complete event log involved enriching the log by adding attributes that may provide additional information of great value to the end user of the Process Mining analysis. An excerpt of the final event log can be viewed in figure 4.4.



	CASEID	EVENTID	ACTIVITY	TIMESTAMP	ORG_RESOURCE	VARE	PLANT	CUSTOMER_NR
1	1100005908977000140	11000059089770001400025766673000140	Credit Memo	2019-01-14/195120	BATCH	Drivstoffilter	7800	0000023380
2	1100005909474000190	11000059094740001900015839601000190	Returns delivery for order	2019-01-08/123937	[redacted]	Luftfilter	7800	0000078890
3	1100005909750000080	11000059097500000800015839847000080	Returns delivery for order	2019-01-09/143613	[redacted]	Oljefilter	7700	0000077031
4	1100005909842000160	11000059098420001604919017938000016	Goods Movement GD rtrns unres	2019-01-10/104333	[redacted]	Motorvarmerelement	7600	0000077603
5	1100005909842000170	11000059098420001700025767026000170	Credit Memo	2019-01-14/200310	BATCH	Sender T91 Sort Inkl Batteri	7600	0000077603
6	1100005909842000170	11000059098420001700015839925000170	Returns delivery for order	2019-01-10/104332	[redacted]	Sender T91 Sort Inkl Batteri	7600	0000077603
7	1100005909858000140	11000059098580001400015839941000140	Returns delivery for order	2019-01-10/112012	[redacted]	Sentrifugalpumpe	7500	0000075010
8	1100005936752000070	11000059367520000700025789192000070	Credit Memo	2019-07-15/200040	BATCH	Automatgirkassefilter	0001	0000077603
9	1100005951009000180	11000059510090001800015877185000170	Returns delivery for order	2019-10-31/152659	[redacted]	Monterings kit	9600	0000025438
10	1100005951009000190	11000059510090001900025798567000190	Credit Memo	2019-10-31/195158	BATCH	NGK pluggkabelsett	9600	0000025438
11	1100005951009000200	11000059510090002000025798567000200	Credit Memo	2019-10-31/195158	BATCH	NGK COILER	9600	0000025438
12	1100005951062000160	11000059510620001600025799018000160	Credit Memo	2019-10-31/201059	BATCH	Tilbeh'rsrem sett	9500	0000095020
13	1100005951062000170	11000059510620001700015877042000170	Returns delivery for order	2019-10-31/090434	[redacted]	Bjuream	9500	0000095020
14	1100005951062000170	11000059510620001704919537119000017	Goods Movement GD rtrns unres	2019-10-31/090435	[redacted]	Bjuream	9500	0000095020
15	1100005951174000080	11000059511740000804919538238000008	Goods Movement GD rtrns unres	2019-10-31/132625	[redacted]	Tappeplugg	7900	0000027629
16	1100005908539000150	11000059085390001504919000541000015	Goods Movement GD rtrns unres	2019-01-02/095256	[redacted]	Varta Blue Dynamic E11 680A 74Ah	9300	0000079657
17	1100005908591000060	11000059085910000600015838796000060	Returns delivery for order	2019-01-02/120338	[redacted]	Pedalbryter	7600	0000076469
18	1100005908593000110	11000059085930001100025767028000110	Credit Memo	2019-01-14/200327	BATCH	Varta Blue Dynamic E11 680A 74Ah	7600	0000077622
19	1100005908803000160	11000059088030001600015838990000160	Returns delivery for order	2019-01-03/115201	[redacted]	hjullager	7500	0000075290
20	1100005908823000090	11000059088230000900015839010000090	Returns delivery for order	2019-01-03/123728	[redacted]	LED Posisjonslykt 9-36v 3led ValueFit	9500	0000095006

Figure 4.4: Excerpt of the final Event Log ready to Feed the Process Mining tool - Usernames censored

The analysis thus far has detailed the step-by-step guide towards a complete event log aimed at using Process Mining to analyze the return order process in a company using the ERP-system SAP. The next portion of the analysis is dedicated to using the event log for process discovery in order to evaluate the model stemming from the event log that were created following the six steps on data from the company *A-store*.

4.3 Evaluating the Process Discovery Model

This part of the master thesis will evaluate the quality of the discovered model, within the field of Process Mining to answering whether it is good or not. KPMG provides us with access to the powerful tool of Celonis to visualize the process easily and get insight in process variations, in order to evaluate the quality. To run the Process Mining analysis on our Event Data, we inserted the event log in Celonis by using an Excel-file. The tool provides a function that creates relationship between the columns in the excel-file. These steps are not crucial, which is why we jump straight to the illustration of the model we are evaluating. Like the theory, the evaluation is based on four dimensions: *Replay Fitness*, *Simplicity*, *Precision* and *Generalization*.

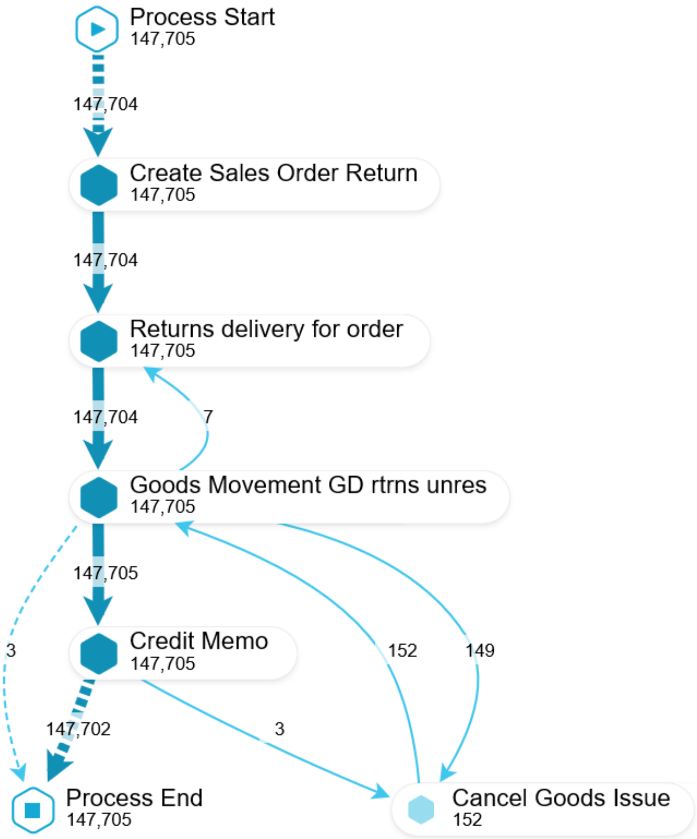


Figure 4.5: The Discovered Model of the Return Process

In the next section, we will use the above model to describe whether it is good and operationalize the four quality dimensions in different ways.

4.3.1 Replay Fitness

The model allows us to see the connected interactions between the data in the event log. Looking from the theoretical approach, it is difficult to say whether we can see certain sequences or not, since the quality of the event log has not been evaluated. The illustration only shows the most frequent traces and has minimized the connections to different paths. Like the theory, we can trace all the fractions of traces, a total of 147.705 events in the log. Given these, it is also hard to tell if we've followed up all the traces within the process, from start to end because the event-log doesn't reflect the number of events. Only after we have fed the event-log in Celonis, then we are able to see all the traces. Unlike the theory, we are not able to answer if we have skipped a step within the return process or not because we have not evaluated the quality of the event log. This leads us to new questions: How is it possible to evaluate if we have optimal results? What indicates the result of using a plus (+)? What defines "accurate reproduction"? How do we evaluate the quality of an event log?

However, we could characterize the replay fitness as medium-plus good, which gives us the result of plus (+), as a result of observing behavior outside the main path. The following illustration shows a model that allows us to see behavior in the event log.

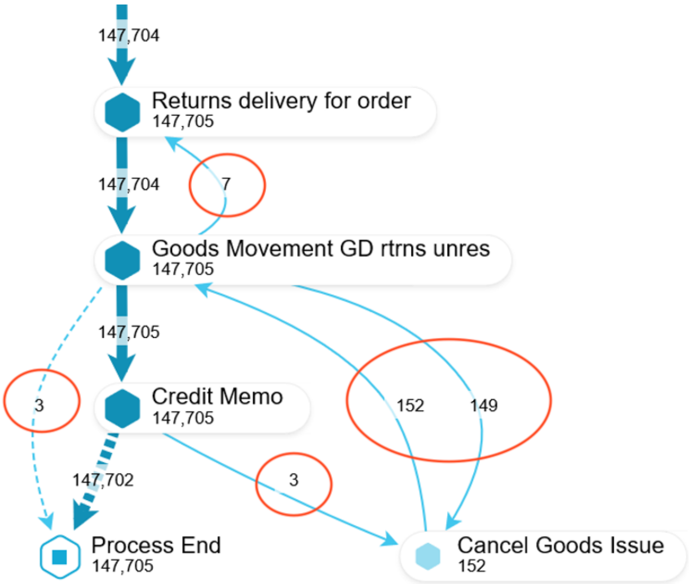


Figure 4.6: The Red highlights indicate Behavior seen in the Event Log

4.3.2 Simplicity

In the context of Process Mining literature, a high-quality model with a simple dimension is considered to be positive if it describes the flow of activities. The discovered model we have developed, is characterized as simple because it reflects the behavior seen in the event log. However, we don’t get insight in all the combinations of the variants, such as loops and details about the order of each activity. Looking from one perspective, it provides an overview of all the happenings, because we understand the combinations of activities on a basic level, compared to a complex model, cf. model N_4 in section 3.1.3. On the other hand, it is difficult to evaluate whether the discovered model has a simple visual image of the actual return processes in the company. More specifically, the model does not show deviations from the event log, such as the behavior of skipping a step or illustration of a blocked payment.

Unlike the theory, the description of behavior seen in the event log is not specified with any details. One can imagine that the consequence of missing a step increases the simplicity because the number of activities decreases, and we get an unrealistic view of behavior in an event log. In other words, the event log does not reflect infrequent “noise” that is observed. Given these considerations, the evaluation can mislead us into an inconsistent conclusion based on an event log with poor quality. Anyhow, we evaluate this model to be simple, with a medium-plus quality and the symbol of plus (+), as a result of comparing our models with the theory, cf. figure 2.3 in section 2.1.3. In the next section, the thesis will evaluate whether the model is characterized as *Precise*.

4.3.3. Precision

The discovered model is characterized as a high-quality model with a precise dimension, without “too much” behavior. In specific, this includes a variety of activities that are visualized clearly, accurate and understandable. There are no attributes that form a “flower model” (see N_3 at section 2.1.3). In other words, the discovered model is quite precise and not underfitting. The problem of underfitting occurs when the model allows for many heterogeneous behaviorists, and it is difficult to distinguish the varieties from each other.

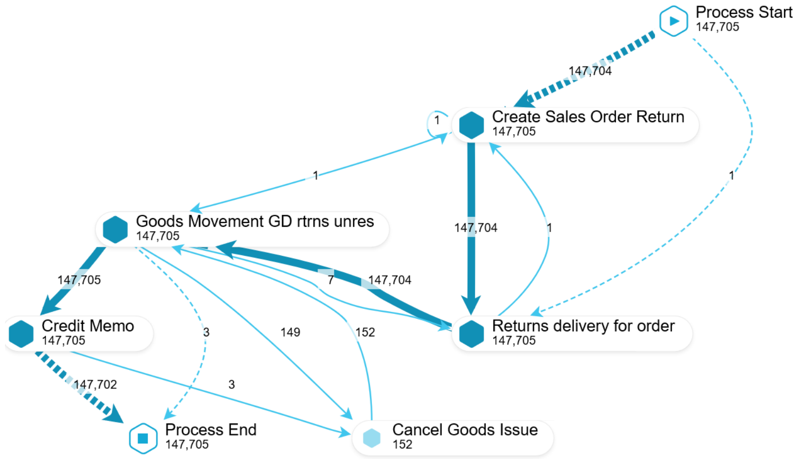
Looking at it from one perspective, one can imagine that our model is not precise, because the activity “Release Billing Block” is not a part of the event- log, and thereby the return process, based on the table- structure in SAP, cf. section 2.1.3. The weakness of the theory lies in answering the consequence of skipping a process-step. An ongoing argument could state that the theory needs more scientific evidence to evaluate the discovered model. However, if we tried to reconstruct the event-log by including more attributes, e.g., AUART, VGART and table SE11, we might not receive the same form of observational results, as the current event log. Looking at another aspect, the discovered model is considered as over-generalized, due to the behavior seen in the event-log. By that, we mean that it’s hard to tell if we answer the question: *How can we evaluate if the behavior seen in the event-log reflects our observational review in the discovered model?* Like theory, the thesis is driven by questions. The advantage lies in people’s opportunity to be curious and autonomous, to solve sub-questions. Despite seeking answers to the above question, our discovered model sticks to a medium-high quality with the plus-symbol (+).

4.3.4. Generalization

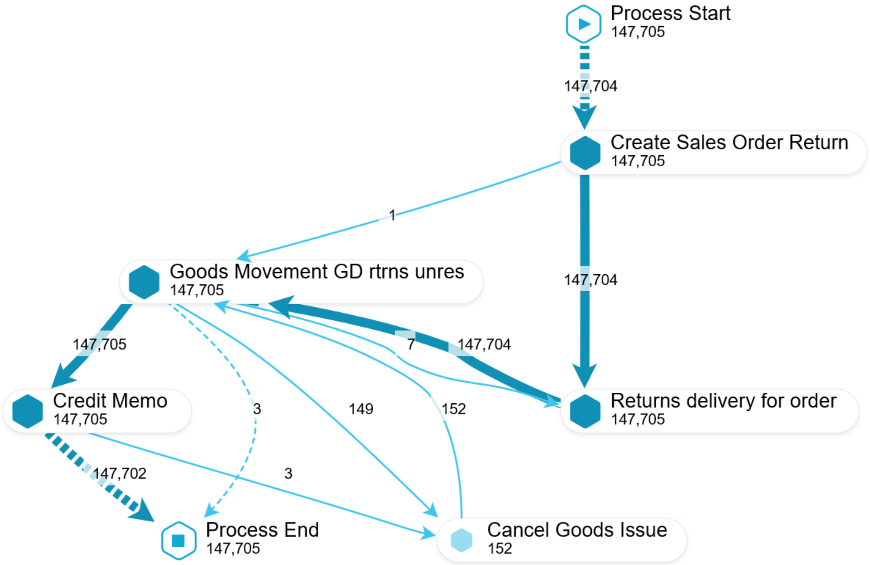
In the context of Process Mining, our discovered model is considered to be *overfitting* because the behavior recorded in the event log, produce a very specific model with few behaviors. One can imagine that the behavior of this process doesn’t have any other patterns than the ones we observe, which gives a skewed distribution of the contents of the event log. We assume that there are several patterns and activities within the dataset, that has not been included in the construction of the event log, due to poor data-quality and limited time to model. In addition, the disadvantage of evaluating the generalization in the event log, is due to the balance between *overfitting* and *underfitting*.

Like the theory suggests, the Process Mining algorithms in Celonis strikes a balance between overfitting and underfitting that enables us to adjust the level of insight into the process. However, if we adjust the connections to 100% and the activities to 100%, the generalization strikes a poor balance between the indicators, and vice versa.

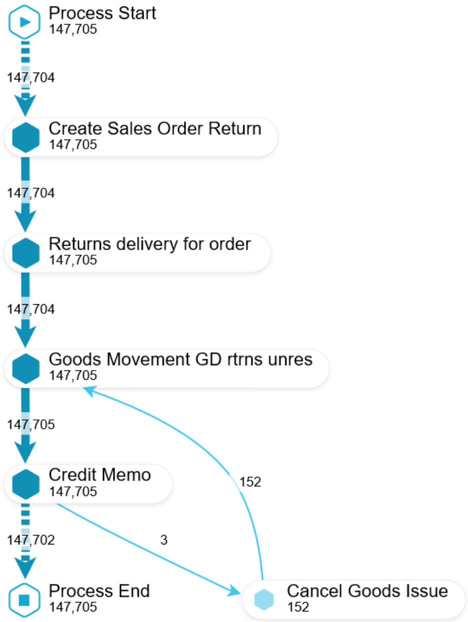
To clarify, adjusting the connection level in the context of process discovery is a method of filtering out the infrequent “paths” which some cases may take, cf. *fine-grained scoping* in figure 2.3. A connection level of 100% means that every possible path which a case in the event log may follow is included in visualization of the process model. Equally, a connection level of 0% hides all paths but the most frequent one. If we decrease the level of connections whereas the level of the activities stays unchanged, the discovered model ends up being unbalanced. Therefore, the generalization ends up with the "grade" minus (-). To solve this, we've tried to balance the two forces by adjusting the connections, while the level of activities stays unchanged because they are few. The figures below illustrate a recent attempt to balance between overfitting and underfitting, respectively 100%, 50% and 0%. Unfortunately, this has not led to better generalization and thus, no better quality of the discovered model. As a result, we conclude that it's difficult to balance between being too general and too specific, whereas our model is characterized as *overfitting* (-).



The figure illustrates the Discovered Model with a connection level of 100%. Source: Screenshot in Celonis



The Figure illustrates the Discovered Model with a connection level of 50%. Source: Screenshot in Celonis



The Figure illustrates the Discovered Model with a connection level of 0%. Source: Screenshot in Celonis

4.3.5 Conclusion: The Four Quality Dimensions

The discovered model in figure 2.3 can reproduce almost all the traces seen in the event-log. Overall, there are 38.260 return-orders within the return-process that can be observed. Even though the SAP activity “Release Billing Block” has not been identified during the construction of the event log, the quality of the Process Mining result has been possible to evaluate, looking at the traces seen in the event log. In addition, there is a high probability that there are several activities that have not been included, due to limited insight into certain tables, cf. section 4.2.3). By that, we conclude that the discovered model has a high-level characterization of three dimensions. More specifically, replay fitness, precision and simplicity. One dimension is characterized with poor quality, due to the assumption of skipping the step “Release Billing Block”, which over-generalizes the behavior seen in the event log. Set up against the parallels of theory, the discovered model is overfitting because the corresponding mining technique assumes a quite strong notion of completeness (Aalst, 2016, p.190). For the other three dimensions, we conclude that they have a high-quality characterization. Overall, the discovered model is good, and our results are:

Figure 3.4: ***Fitness*** = +, ***Precision*** = +, ***Generalization*** = **-** , ***Simplicity*** = +

5 DISCUSSION

This section will discuss how our findings contribute to answering the research questions, as well as potential areas for improvement based on the analysis. Additionally, spending time on this project has led to uncover some areas that should be further investigated, in order to strengthen the research on Process Mining in the SAP-environment, as well as enhancing user friendliness for practitioners. As the case study in section 6 of this thesis is regarded as supplemental and not critical for answering the research questions, we have chosen to exclude the findings from it in the following discussion. This further implies that the case study is a supplemental illustration of our work and of the valuable insight that Process Mining may provide, cf. Section 6.3 for a discussion of the case study results.

We embarked on this project with the goal of making Process Mining analysis possible on the return order process in SAP, focusing on the construction of a suitable event log for the purpose. On this journey, uncovering the crucial elements or steps towards the complete event log were key to make the subsequent process discovery of an acceptable quality. Determining the quality of the discovered model was done using Van der Aalst's (2016) four quality dimensions *Replay fitness, Simplicity, Precision* and *Generalization.*

5.1 Discussion of Results

Previous research reveals that Process Mining has mainly been applied in the areas of the software development process, online education and healthcare (Becker & Intoyoad, 2017). The same study shows that the opportunity to mine logistic processes depend on the ability to handle heterogeneity. If we view this thesis approach to constructing a return order event log, the perfect result would be to extract data that reflects valuable information about returned items, by tracking and tracing every item through each step in the process, as recorded by the SAP system.

Constructing a CaseID which uniquely identifies each returned item, our first step towards the complete event log, could be said to support an ideal result as stated above. This is partly because we can be certain that all the returned items exist in the real world, and all items have to be recorded into SAP through the *sales order document*. This ensures completeness in the sense that our approach has captured all the return order items.

Although our method of choosing the case identifier supports the notion of an ideal event log, our step involving the construction of activities might be considered a tad too simple. Our approach of extracting the lion's share of the activities through the *sales document flow* table makes the step overly simplistic because some less frequently performed activities may be left out of the final log. Another alternative approach would have been to extract activities from the *change document* tables, but since these tables contain all changes recorded in the SAP system, such an approach might again have been too detailed. Considerations regarding level of detail aside, we can be sure that the activities extracted using our method are present in the actual return order process, as they match the activities of the flow chart presented in chapter 3.1.2.

Moving on to the next step in our step-by-step guide, we turn our focus towards constructing the timestamp. There is no doubt that this step is crucial to be able to perform Process Mining analysis, as it orders the events in chronological order. What could have been done differently is choosing another granularity level, for instance only date or date, hour and minute, excluding seconds. However, as several of the events are performed in rapid succession, no coarser granularity would have been able to order the events correctly, equally no finer granularity is available in the SAP system. Put differently, our approach is sufficient to deal with concurrency in the data set, as pointed out in section 4.2.1.

Constructing the EventID was done using a combination of several SAP table columns to ensure that each recorded event could be uniquely identified and connected to their respective process instance, i.e. their corresponding case. As long as these two demands are met, the chosen style of the identifier is not considered crucial. There are possibly several other methods of constructing such an event, e.g. by giving each event a unique, random number. Our approach of combining the CaseID with subsequent document numbers and positions made the EventID disproportionally large. On the other hand, it is never unclear which events belongs to which cases, since the CaseID is part of the EventID.

On extracting the event data, our fifth step towards the complete event log, a major part was to decide what tables to extract data from and tying the log together. This task is quite substantial and time-consuming, so like the theory, we narrowed down the event data to a minimum, in conjunction with our goal of uncovering the most important steps in the pre-processing phase, i.e. constructing the event log. Once again, the completeness of the extraction is somewhat questionable, as it is not likely to capture all events related to the return order process, since the chosen tables for the extraction are so few. Missing out on some events will inevitably mean that the process model constructed by a Process Mining tool will not be able to capture the real-life process in its complete form, thus lowering the quality of the subsequent analysis.

As for the sixth and final step identified and performed in this project, one could possibly criticize it for not being crucial to be able to perform Process Mining on the event log. This may be true; it is possible to at least perform Process Discovery and to get statistics of the different cases etc. However, this would have very limited use for the practitioner because it cannot answer questions like “Why is case 1 slower than case 2?” or “Is department B conducting their return process differently than department A?”. Which attributes to include may vary, the ones we chose to include can be viewed as suggestions. As *GP2* of section 2.2.4 suggests, the extraction should be driven by the questions one seeks answers to, thus constructing a “one size fits all” type of event log for the return order process is not possible. Additionally, it is worth mentioning that some of the additional attributes, e.g. usernames, may give rise to privacy concerns. For instance, is it ethically justifiable to perform an analysis of which users can be considered bottlenecks in the return process because they handle cases slower than the average? If results of the Process Mining analysis were to be used as arguments for layoffs rather than to improve the return process within a company, we are facing serious threats to privacy and job security.

The second research question of our thesis is dedicated to determining how good the process model resulting from the constructed event log is. Of the four quality dimensions, it was clear that *Generalization* scored lower than the other three. Specifically, we observed that the model could be characterized as *overfitting* in the sense that the event log is too close to the SAP template in section 3.1.2. By this, we deduce that the possible heterogeneity of the real-life process might not have been captured by the event log, thus resulting in a lower score on the quality dimension of Generalization. This is likely due to the possibility of the event log lacking some rare events in the form of infrequent activities as discussed earlier in this chapter. Generalization aside, the model did good in terms of *Replay fitness, Simplicity* and *Precision* – but it is hard to decide just how good using the model of Van der Aalst presented in chapter 2.1.3. This will be further discussed in the next section of the thesis, concerning recommendations for further research.

5.2 Recommendations for further research

Like the subsequent chapter, 5.1, this chapter is aimed at enriching academic literature, based on the work put down to answer the problem statement and the research questions. The recommendations presented in this section is not to be confused with the recommendations of chapter 6.4, which is related to the supplemental case study.

Three areas in need of strengthening and further research have been identified while working on this project. The first two relates to the construction of event logs, whereas the third has to do with evaluating the quality of process models, a discussion briefly brought up in the preceding chapter. We present our suggestions for further research in the form of bullet points below, including a short elaboration of each point.

- *Standardized coding* for event log construction. Constructing event logs do inevitably require a certain level of knowledge of database structures and coding languages, such as SQL. Though, some processes may be very heterogeneous, there will usually exist a core that is more or less the same for typical processes in the SAP environment. End users could possibly save a lot of time if standardized coding and procedures for creating event logs were more available and easier to access, e.g. essential SQL-queries. Additionally, this can contribute to lowering the threshold of technical competence needed to perform Process Mining.

- *Reference book of standard SAP processes* and their essential tables and columns. This could seriously limit the time spent on researching procedures and tables in SAP, decreasing the need for extensive domain knowledge. Mapping out the most essential tables, columns and connections related to the most common SAP processes relevant for Process Mining would make event log construction faster and easier.

- *Evaluating process models.* Using the four quality dimensions of Van der Aalst (2016) to evaluate process models should be extended in order to provide a better understanding of just how good a model performs. We suggest developing a scale within each of the four dimensions, complete with associated *scaling statements* and examples. Only being able to give a quality dimension either a *plus* (+) or a *minus* (-) is too coarse to properly evaluate models or comparing them. A possibility is to develop a five-point scale of each of the four quality dimensions, much like previous research of Van der Aalst et. al. (2012) has done regarding maturity level of event logs. The aim must be to make assessing and ranking process models more detailed.

6 CASE STUDY

This section of the thesis will serve as an example of how the constructed return-order event log may be used to provide valuable insight from raw data provided and extracted by a case company. By applying the steps towards a complete event log identified in our project thus far, we will use a Process Mining tool to analyze the return order process from different angles, culminating in a few recommendations to the case company based on our findings. First, this section starts off with a brief introduction of the firm, before presenting the analysis

The case company, referred to as "A-Store", is an international firm operating in the car-parts industry, with subdivisions located throughout Norway. The company is an existing client of KPMG, and an extraction of data from their ERP-system was already available from the audit department of KPMG Norway. In the context of Process Mining, we chose to limit the number of variables in our event log to avoid noise and incompleteness. Therefore, the first section will show an overview of data we're dealing with. Secondly, we will illustrate and describe extracted information and value from data stored in our event log. We narrow the scope by describing the most frequent variants, to show what Process Discovery can provide of valuable information. The next part will discuss our evidence, in the context of business optimization. Lastly, the case study provides 3 recommendations for potential improvements.

6.1 Table Characteristics

Firstly, we started off by gathering information about number of records in each table we were dealing with. This gives an image of the entire OTC process; Table 6.1 presents this overview. Even though there are certain tables that reflect more important information about the return-process, there are common information, steps and activities within the OTC-process needed to be extracted, in order to gain valuable insight.

Name	Table	#Records	Name	Table	#Records
Changes Header	CDHDR	2.334.661	Changes Lines	CDPOS	12.258.976
Document Segment Material	MSEG	2.188.319	Material Doc. Table Header	MKPF	578.114
Sales Doc: Item data	VBAP	1.094.314	Sales Document: Header Data	VBAK	487.770
Open Payment Doc.	BSID	37.134	Sales Document Flow	VBFA	4.027.519
Sales Document: Delivery Document	LIKP	538.745	Sales document: Delivery item line	LIPS	1.508.868
Invoice Document	VBRK	195.736	Invoice Lines	VBRP	1.092.806
Payment documents	BKPF	1.190.376	Closed Payment Doc	BSAD	800.215

Table 6.1: An Overview of SAP Tables and Number of Records utilized during modeling in SQL Server

6.2 Process Discovery

6.2.1 Metrics

Based on the behavior seen in the event log, the majority of the return orders have an average duration of 0-2 days from process start to process end, with a total of 59.697 cases (relative amount: 40%). The slowest return-orders are completed between 18-139 days and makes up a total of 2.246 cases (relative amount: 2%). Overall, the average duration for a return-order to be completed is 4 days. See *Appendix A* for detailed illustration of figure 6.2.

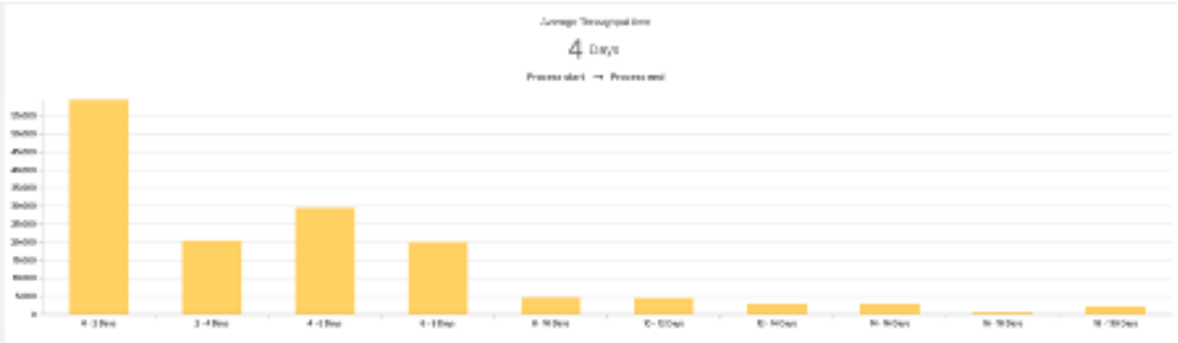


Figure 6.2: Number of Cases and Throughput time for 38.230 Return- Orders

Equally important, the development of cases per day seems to depend on seasonality, see the fluctuations in figure 6.2. The majority of return orders are handled from March 2019 to June 2019. The biggest drop in numbers is in August, whereas the sloping tendency starts in the beginning of June and lasts until the rise in August. In total, there are 678 cases per day that processes from start to end, whereas the number of events per day is 2.712. Given that the extreme outliers are not taken into consideration, the average case duration is 3 days from process start to process end. A total of 99.89% of cases flow from most frequent starting activity to the most frequent ending activity, known as the *Happy Path*, and adds up to 147.705 cases in total. The following four activities occur most frequently:

Create Sales Order Return - Returns Delivery order - Goods Movement Rrtns. unres – Credit Memo

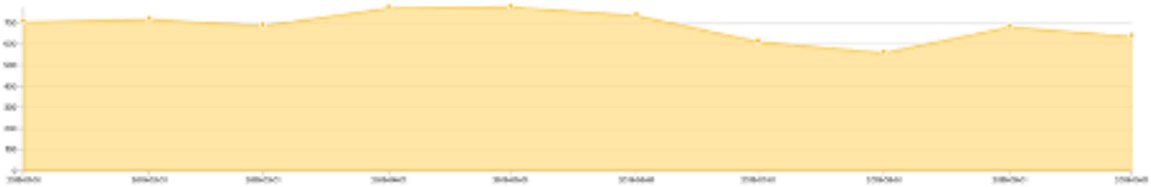


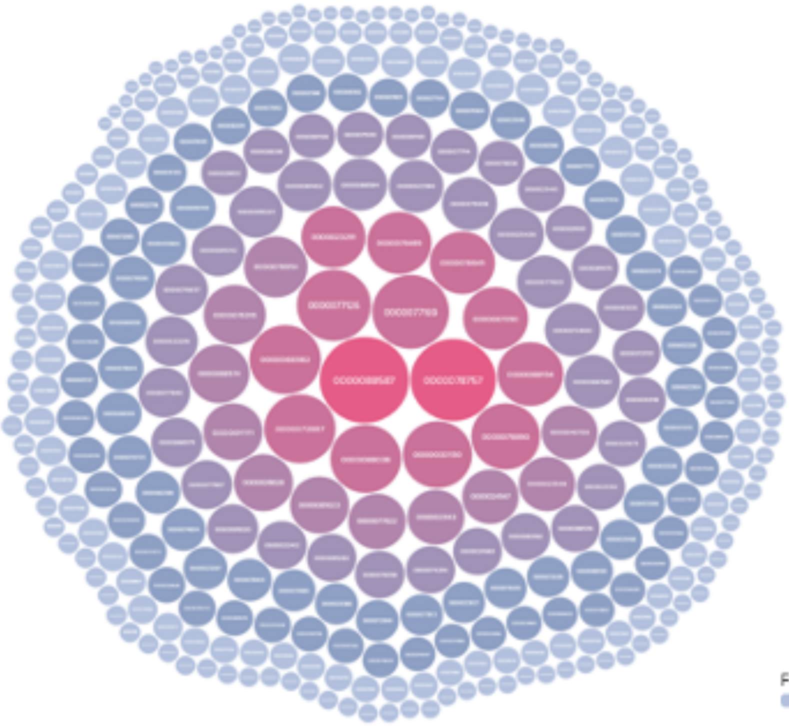
Figure 6.3: Shows the Process Discovery - Development of Cases per Day
(see *Appendix B* for detailed illustration of the figure)

6.2.2 Analysis Units

This chapter will introduce four analysis units, which reflects the behavior seen in the event log: *Customers, Usernames, Returnable Products* and *Storage Locations*. In general, the biggest circles, representing variables have a higher importance in relation to the return process, compared to smaller circles. The color-codes indicates variables causing inefficiencies. Yellow reflects quickly resolved return-orders, while pink indicates slowly resolved cases. The nuanced colors reflect velocity somewhere in-between. In addition, we use color-codes to count number of units shipped in return, where purple indicates few products and pink reflects many units. The next part will give a short summary of units that may lead to inefficiencies in the process.

6.2.2.1 Customers

The top five frequent customers that return the most products, ends on: 88587, 78757, 77169, 77126 and 78890. The number of events counted for the most frequent customer 88587 is 9.308, whereas the average number of different activities per day for that particular customer is 4 per day. The average end-to-end throughput time is 110.2 hours, whereas the average number of events per day is 404. More information about the most frequent customer can be found in *Appendix C*.



The figure on the left-hand side shows the total number of customers in the discovered process model.

6.2.2.2 Usernames

The top five frequent usernames that handle return orders are: *BATCH, NAESST1, EILEKL1, TAJEBE1* and *CHRIRO1*. Based on online research, we assume BATCH is an automated SAP system-user that have authorization to do modifications in the database and is not a real person because the total number of events are increasingly high; a total of 137.042 events. NAESST1 handles in total 55.863 events, 358 events per day, and the average number of heterogeneous activities are 3 per day. In addition, average end-to-end throughput time for this particular user is 99.1 hours. Moreover, 100% of all the cases that NAESST1 handles, is channeled to BATCH, BAKKBJ1 and FREIEV1. See *Appendix D* for more information.

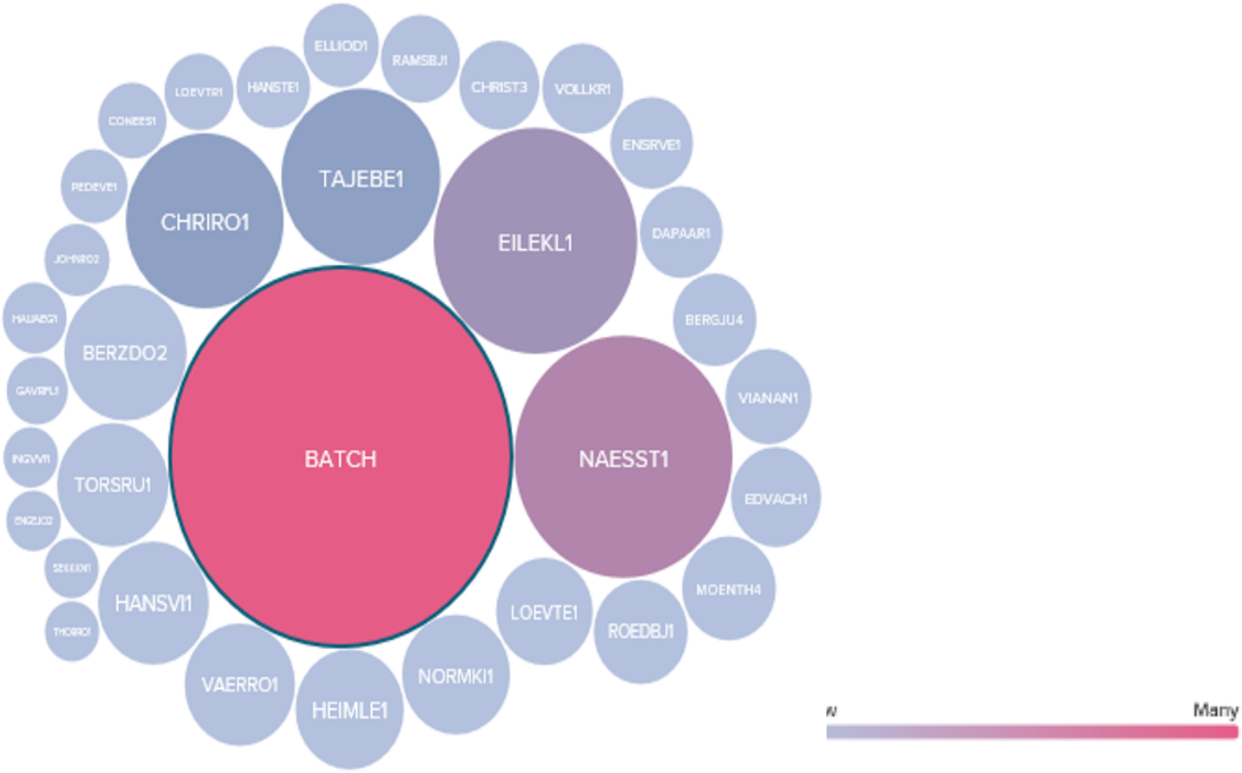


Figure 6.4: The relative Number of cases each User handles. Source: Screenshot from Celonis

6.2.2.3 Returnable Products

The most frequent products that have been sent in return are: *Caliper, Drivstoffilter, Oljefilter, Bremseklosser Foran* and *Luftfilter*. The product Caliper has a total number of events adding up to 63.320, compared to next-largest event, which is *Drivstoffilter* with a total of 23.814 events. The average number of events for Caliper is 292 per day, and the average start-to-end throughput time is 106.8 hours. The average number of different activities are 4 per day. We assume that the reason this particular product is returned, is due to wrong component size or other quality discrepancies. After a quick search online, one can imagine that a *caliper* is custom made for specific vehicles and is in short supply, or out of production for certain vehicles. In other words, car repair shops face a challenge in repairing vehicles, due to obtaining the right caliper to repair for example older vehicles. Again, these can affect A-stores liquidity negative, by increasing costs due to for example transaction costs, transportation costs and current assets. One can also imagine that capital in the form of current assets, is tied up in inventory. See *Appendix E* for details about the Caliper.

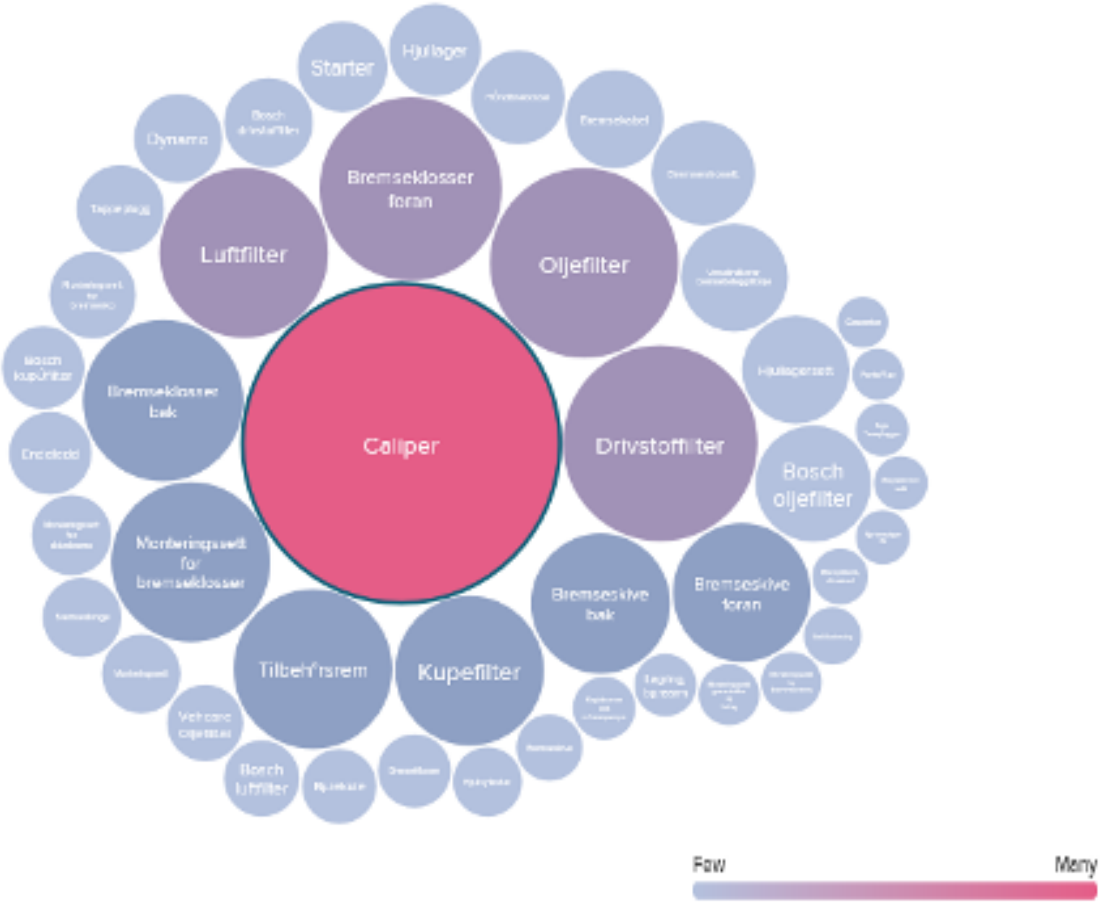


Figure 6.5: The most frequently returned products. Source: Screenshot from Celonis

6.2.2.4 Storage Locations

The top four most frequent storage locations are assumed to receive the majority of returned products, and are: *0001, 7800, 1001* and *8700.* Storage location 0001 has an average number of events equal to 460 per day, whereas the average number of heterogeneous activities are 3 per day. 70% of all the cases comes from 7600, 7100 and 9400, and ends up at storage location 0001, whereas the average end-to-end throughput time is 102.4 hours. This could possibly indicate that 70% of returned products arrives repositories before shipping to 0001. However, the storage location that we assume causes most inefficacy is 9000 and 8100. The storage location 9000 has an average throughput time of 193.5 hours and solves an average of 160 events per day through three activities per day, in average. Moreover, 64% of all the cases comes from 7700, 8800 and 7500, and in 99% of cases, they continue to flow to 1001, 7900 and 0001. Again, one can imagine that the returned products arrive repositories before continued shipping in the process-flow. Or, it is possible to assume that most frequent returned products may exist, due to A-stores policy of having available spare parts and meets the necessary quality guarantee requirements, just in case of increased demand. See *Appendix F* for more details regarding storage location 9000.

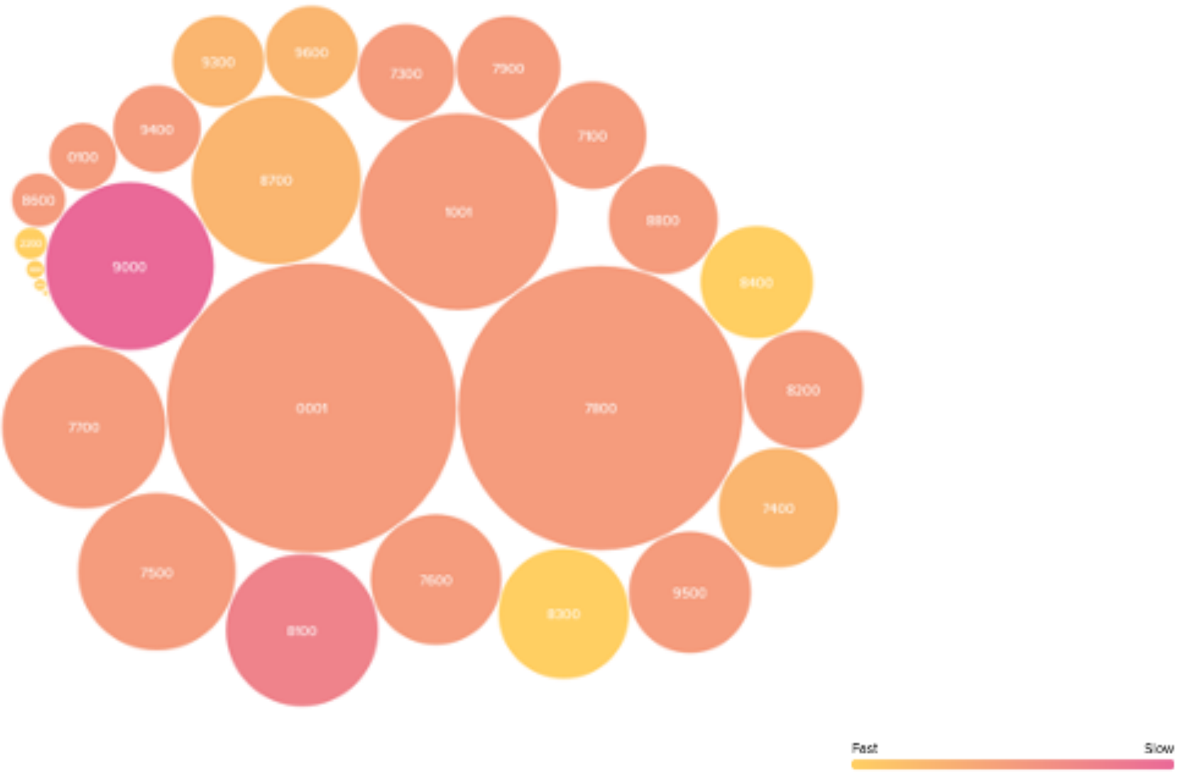


Figure 6.6: Illustration of throughput time for cases handled at the heterogenous storage locations

Source: Screenshot from Celonis

6.3 Discussion

Introducing the analysis units, aids in preventing companies from using unnecessary resources in their business and contributes to maintain efficient return-process within the supply-chain. The objective of this section is to provide valuable insight in the process-flow.

This end-to-end process has illustrated various variables that may be the cause of inefficiency and extra costs, within A-store's return-process. Looking at it from one point of view, it is conceivable that the majority of return-orders go straight to stock after the customer has returned the item. Given that the traceability stops after the activity "Goods Movement GD rtrns unres", this might indicate that the company does not possess data about the process-flow, after the item is in stock. One can imagine that the company chooses to keep the returned product, rather than returning it to its suppliers. This makes the traceability inadequate between stock and supplier. Given that the company choose to keep products in its inventory, capital is tied up to the storage location, that could otherwise generate income. However, we assume that there may be a causal relationship between the variables *Customers* and *Returned Products*, as customers are likely to affect the frequency of returned products. Based on our starting point, it may require more dynamic follow-up activities along customer 88587, as we assume, he/she is likely to influence the increased frequency of returned items. The advantage lies in the fact that this comprehensive process overview provides an objective, though limited insight into the entire process-flow.

From another point of view, it is assumed that there is a causal relationship between the throughput time of different users and the speed of inventory. Like we observe in figure 6.3, there are fewer return-orders being managed during the summer period. Therefore, it is possible that the decline of handling return-orders in July and August is due to increased inventory in the summer. This might be costly, as it could tie up capital in the inventory during low seasons. For remaining seasons, one can imagine that high-performance users, e.g., users that possess high-volumes of return orders, may increase the productivity, as high working capacity leads to faster shipments to storage locations. Given that this is a causal link, the business can eliminate waste and narrow the return-process, by saving resources and avoiding unnecessary bottlenecks.

However, even though this Master thesis has provided insight into how Process Mining can be applied on a return-order process in the context of supply chain management, it is just a starting point. There are multiple arenas where this type of analysis can be applied. We have gained some additional perspectives of the possibilities that Process Mining might give stakeholders, such as firms, customers and employers. There are valuable insights to be discovered in other types of processes, such as *Project Planning* and *Customer-Relations-Management (CRS)*. The next section will line out some reflections, in light of our experience of working on this project.

As previously stated, the construction of this thesis' event log, enables for example businesses to get a better understanding of CRS -processes. Whilst reflecting on our experiences writing this thesis, it might be possible to extract data from data warehouses, where firms can improve customer satisfaction by extracting data from different communication channels, e.g. phone support, online chat and email support.
Moreover, it is possible to discover throughput time spent on responding customers inquiries; for instance, 5-10 days or 10-15 days. The possible value to be extracted, could facilitate customized support, via for instance chatbots or other automated solutions. Anyhow, in light of our learning experience, it came to our attention that A-Store could utilize their data to apply Process Mining on other business areas. Suggestions of candidate operations include but are not limited to the Purchase-to-Pay and Order-to-Cash processes. Additionally, upon analyzing their return order process, we found that there are a number of interesting attributes that could provide additional insights. For instance, an inclusion of the pecuniary values of both return-orders and the individual order lines may be of interest to the company in mind. There might be a relationship between the value of orders and the throughput time of each case, as well as other factors yet to be discovered.

Essentially, the areas of applicability for Process Mining are many. The main challenge, however, is to ensure that the raw data and resulting event logs are of sufficient quality for stakeholders to discover the real potential.

6.4 Recommendations

These follow-up activities are predicted to increase the efficiency of the return-process, both in the long and short term:

- Improve historical data-quality by continually increasing the quality of the data warehouse to simplify traceability.
- Invest in training, competence and skills – research shows that efficiency enhances growth and increases the need for employees in companies (Førde, 2017).
- Open up for visible leadership and transparent communication, as the management should pursue a long-term, step by step change for the follow-up activities of the processes. It contributes to greater ownership of lean (Rolfsen, 2014). Lean is based on the idea of continuous improvements, values and flows in the organization. (Maurya, 2012).

Additionally, we recommend that A-Store consider utilizing the potential of their existing data to discover, analyze and possibly improve other processes, such as the OTC and PTP.

7 CONCLUSIONS

With the rising interest in applying Process Mining to business processes by companies world-wide in mind, this thesis set out to tackle one of the main barriers between raw data and meaningful mining results - a proper event log. The aim of the project has been to contribute with purposeful empirical findings to this young research discipline, as well as to provide businesses and practitioners with a structural approach to constructing high quality event logs of the return order process on data from SAP. This goal led to the following problem statement:

"Develop a step-by-step guide to construct an event log in the context of the Supply Chain Management process of return orders in SAP, aimed at Process Mining, subsequently evaluating the Discovered Model"

The problem statement was further broken down into two main research questions, where the first one dealt with identifying the crucial steps involved in pre-processing raw data from SAP into an event log of the return order process. To answer this question, the *guiding principles* to event log extraction (Aalst et al., 2012) and earlier research on Process Mining and SAP data by Piessens (2011) were central in supporting our approach. The second question concerned determining the quality of a process model after applying *Process Discovery* to the resulting event log. Evaluating the quality was done using Van der Aalst's (2016) four quality dimensions.

By analyzing a 2019 SAP-data extraction from the company referred to as "*A-Store"* it was possible to identify *six steps* that may be considered of utmost importance for constructing and extracting the return order event log. The first step involved constructing a *CaseID* to label the process instances was done to correctly identify each unique order line item for tracking purposes. The next step concerned identifying the different *activities* involved in return order processing. The approach managed to extract the most frequent activities, but as brought up by the discussion of results, the list of activities is likely incomplete. The third step deals with the construction of timestamps of a suitable format for the events in the data set. These *timestamps* are crucial to establishing correct ordering of the events within each case. The fourth step involved constructing an *EventID*, which made connecting events to their respective cases possible. As for the fifth step, the event log was tied together by connecting and *extracting event data* from a set of tables in SAP that holds the most relevant data of the recorded return order process. The sixth and final step wraps up the *step-by-step guide* that was the main goal of this thesis, by including four attributes that contributed to *enriching the event log*. Whether or not this step can be considered *crucial* was discussed in section 5.1. As not including the step would mean that the log cannot serve the analytical purposes of most practitioners and businesses, we conclude that step is in fact crucial to the return order event log.

When evaluating the discovered process model, we conclude that the overall level of quality is medium plus (+), based on high-quality characterization scores on *Replay Fitness*, *Simplicity* and *Precision*, and a low-level characterization score on *Generalization*. The latter is due to over-generalization, which can likely be explained by less frequent activities missing in the event log. Determining the quality level is difficult and unreliable without a comprehensive method of ranking models within each of the four dimensions, thereby addressing a need for further research on the topic.

REFERENCES

Aalst, W. M. P., (2019). Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data. In Ölveczky, P. C. & Salaün, G. (eds.), *Software Engineering and Formal Methods* (pp. 3-25). DOI: 10.1007/978-3-030-30446-1_1

Aalst, W. M. P., (2016). *Process Mining: Data Science in Action.* (Second Edition). Berlin Heidelberg New York Dordrecht London: Springer.

Aalst, W. M. P., Adriansyah, A., Alves de Medeiros, A. K., Arcieri, F., Baier, T., Blickle, T., ... Wynn, M. (2012). *Process Mining Manifesto*. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.), Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing, vol. 99. (pp. 169-194). Berlin Heidelberg: Springer. DOI: 10.1007/978-3-642-28108-2_19

Førde, H. (ed.). (2017, April 6th). Norges Forskningsråd: Effektivisering gir flere arbeidsplasser. *Logistikk & Ledelse.* Source: https://www.tungt.no/logistikk/internlogistikk/norges-forskningsrad-effektivisering-gir-flere-arbeidsplasser-2003513?fbclid=IwAR0gfAk7z7zYW4ye_X9EXZQ1ZxQ2SsQr-%20J9eA7L6tBQ2iW004_Ug4uSgvmo

Gea, F. (2018, January 16th). SAP SD Return Order Process. *ERProof.* Source: https://erproof.com/sd/free-training/sap-sd-return-order-process/

González Lopêz de Murillas, E., Reijers, H. A. & van der Aalst, W. M. P. (2018). *Connecting databases with Process Mining: a meta model and toolset.* Software & Systems Modeling, 18, 1209-1247 (2019). DOI: 10.1007/s10270-018-0664-7

Ingvaldsen, J. E. & Gulla, J. A. (2008). *Preprocessing Support for Large Scale Process Mining of SAP Transactions.* IN: ter Hofstede, A., Benatallah, B., Paik, H. Y. (eds.), Business Process Management Workshops. BPM 2007. Lecture Notes in Computer Science, vol. 4928. Berlin, Heidelberg: Springer.

Maurya, A. 2012. *Running Lean: Iterate from Plan A to a Plan That Works*. Beijing, Boston, Farnham, Sebastopol, Tokyo: O,Reilly.

Nagle, T., Redman, T. C. & Sammon, D. (2017, September 11th). Only 3% of Companies' Data Meets Basic Quality Standards. *Harvard Business Review.* Source: https://hbr.org/2017/09/only-3-of-companies-data-meets-basic-quality-standards

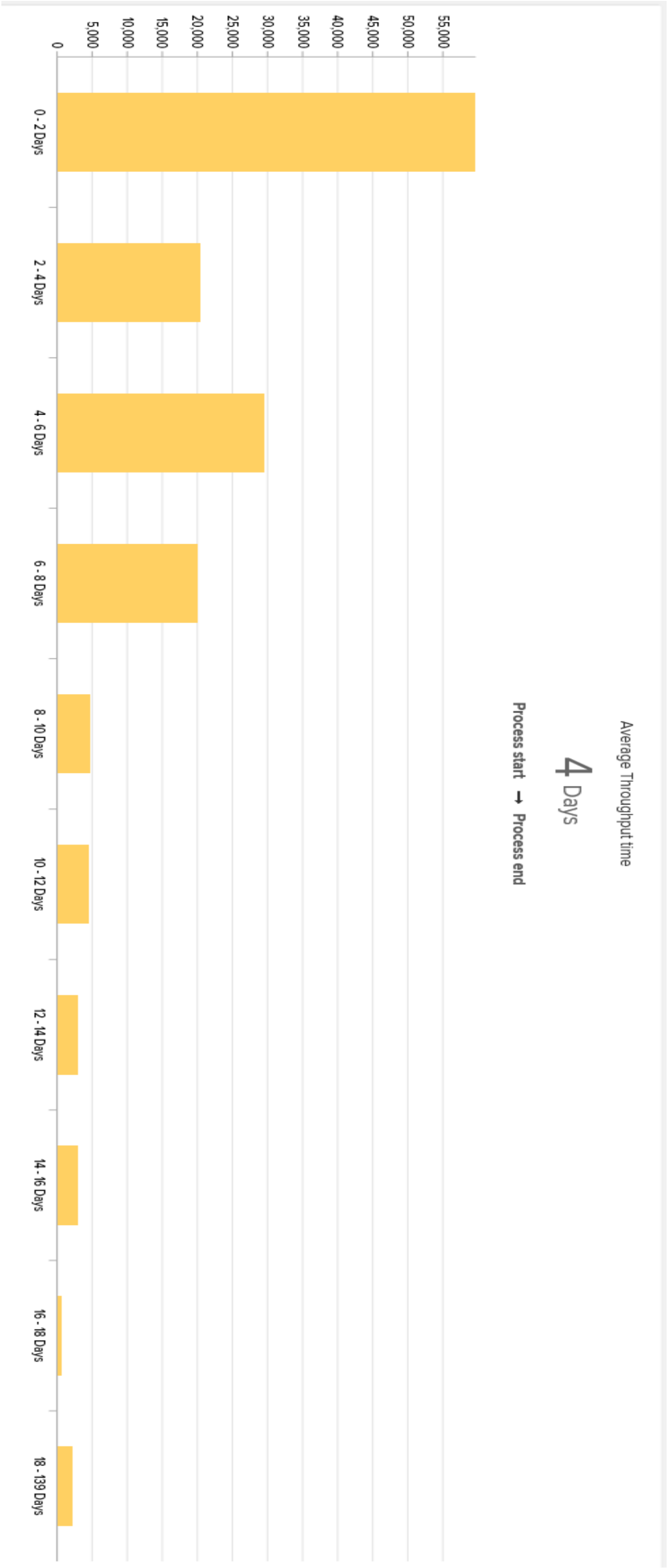
Piessens, D. A. M. (2011). Master thesis: *Event log extraction from SAP ECC 6.0.* Eindhoven: Technische Universiteit. Source: https://pure.tue.nl/ws/portalfiles/portal/47013562/712711-1.pdf

Rolfsen, M. (2014). *Lean blir norsk*: *Lean i den norske samarbeidsmodellen.* Bergen: Fagbokforlaget Vigmostad og Bjørke AS.

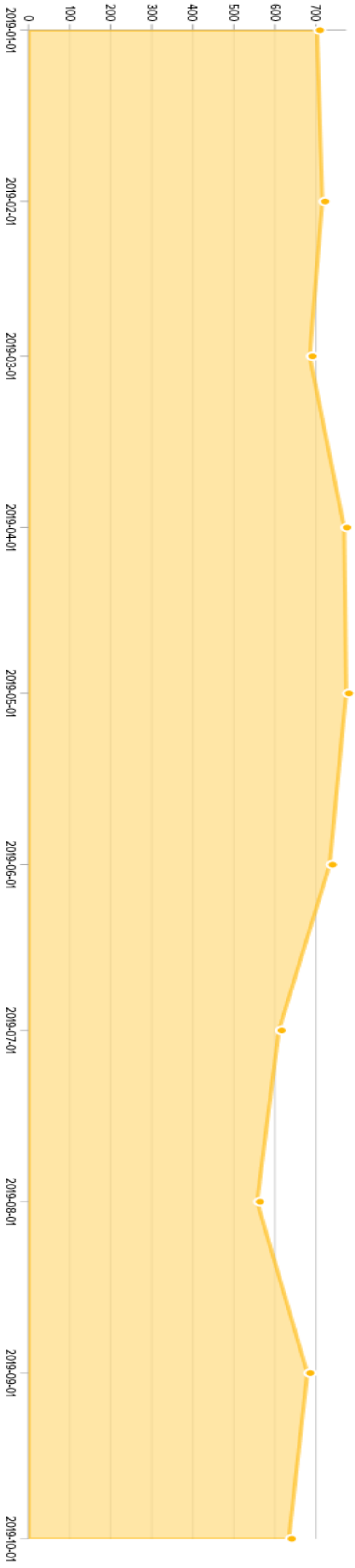
Selig, H. (2017). *Continuous Event Log Extraction for Process Mining.* Degree Project in Information and Communication Technology, Second Cycle. Stockholm: KTH Royal Institute of Technology, School of Information and Communication Technology.

Westerman, G., Bonnet, D. & McAfee, A. (2014). *Leading Digital: Turning Technology into Business Transformation.* Boston: Harvard Business Review Press.

Appendix A

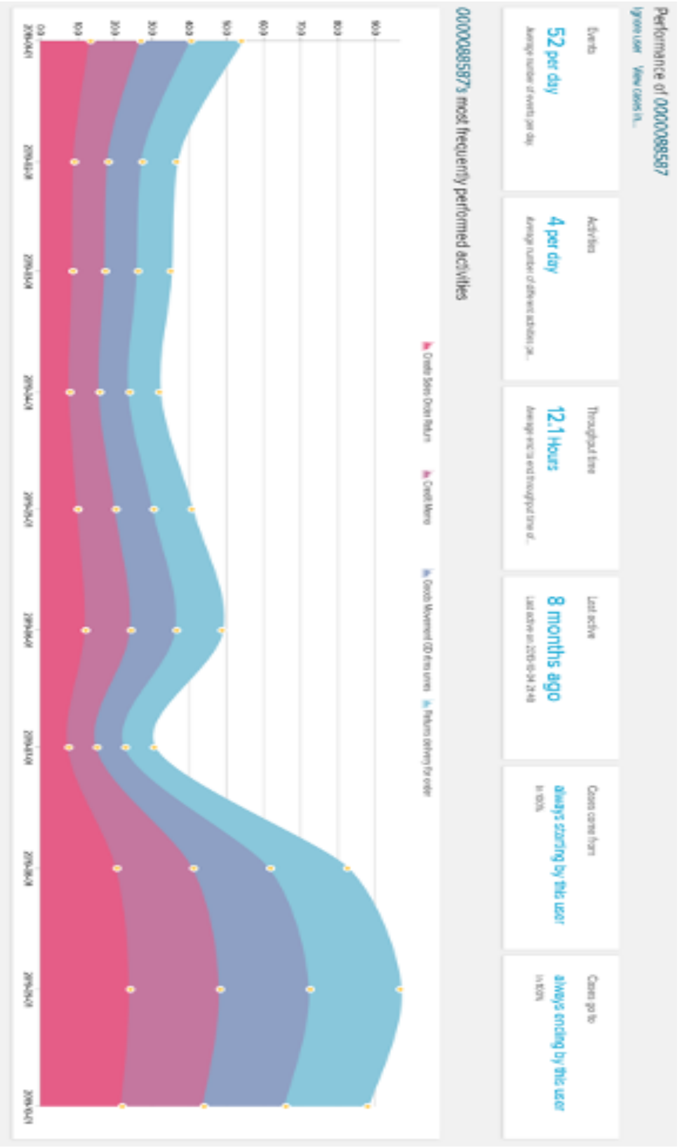

Average Throughput time
4 Days
Process start → Process end
0
5,000
10,000
15,000
20,000
25,000
30,000
35,000
40,000
45,000
50,000
55,000
0 - 2 Days
2 - 4 Days
4 - 6 Days
6 - 8 Days
8 - 10 Days
10 - 12 Days
12 - 14 Days
14 - 16 Days
16 - 18 Days
18 - 139 Days

Appendix B


0
100
200
300
400
500
600
700
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01

Appendix C

The figures provide details about the most frequent customer (no. 000088587) that return the most items. Overall, the customer seems to hand in items most frequently in September 2019. Low season is in July 2019, and here it is possible to set parallels to the summer holiday. It is likely to believe that the customers need items at their storage location in the holiday, as most employees are on vacation and it's not possible to get deliveries from suppliers, to meet demand. In addition, it is reasonable to believe that the customer handles return-orders between 06:00-08:00 at most, and 08:00-10:00 during the day. It is also reasonable to assume that Credit Memo is generated by the system between 18:00-22:00 at the evening, as regularly working day in Norway is between 08:00-16:00.



Performance of 000088587

Ignore user View cases in...

Events
52 per day
Average number of events per day

Activities
4 per day
Average number of different activities pe...

Throughput time
12.1 Hours
Average end to end throughput time of...

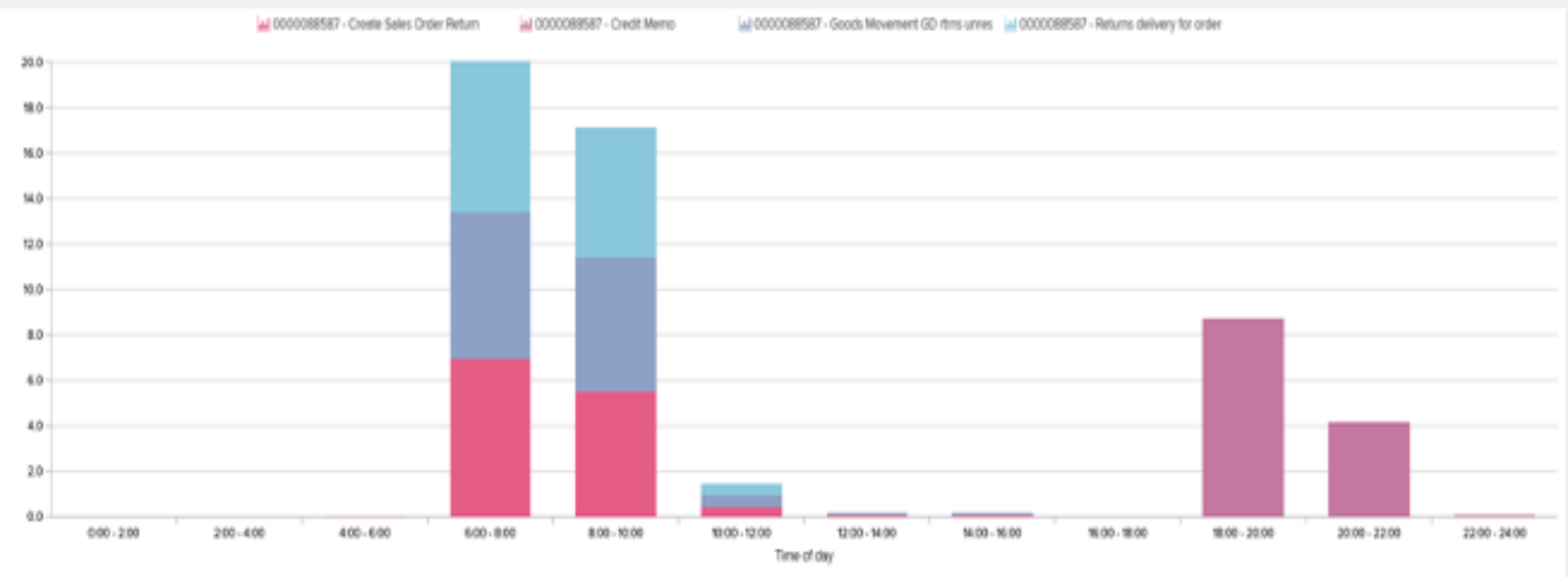
Last active
8 months ago
Last active on 2019-10-04 21:48

Cases come from
always starting by this user
in 100%

Cases go to
always ending by this user
in 100%

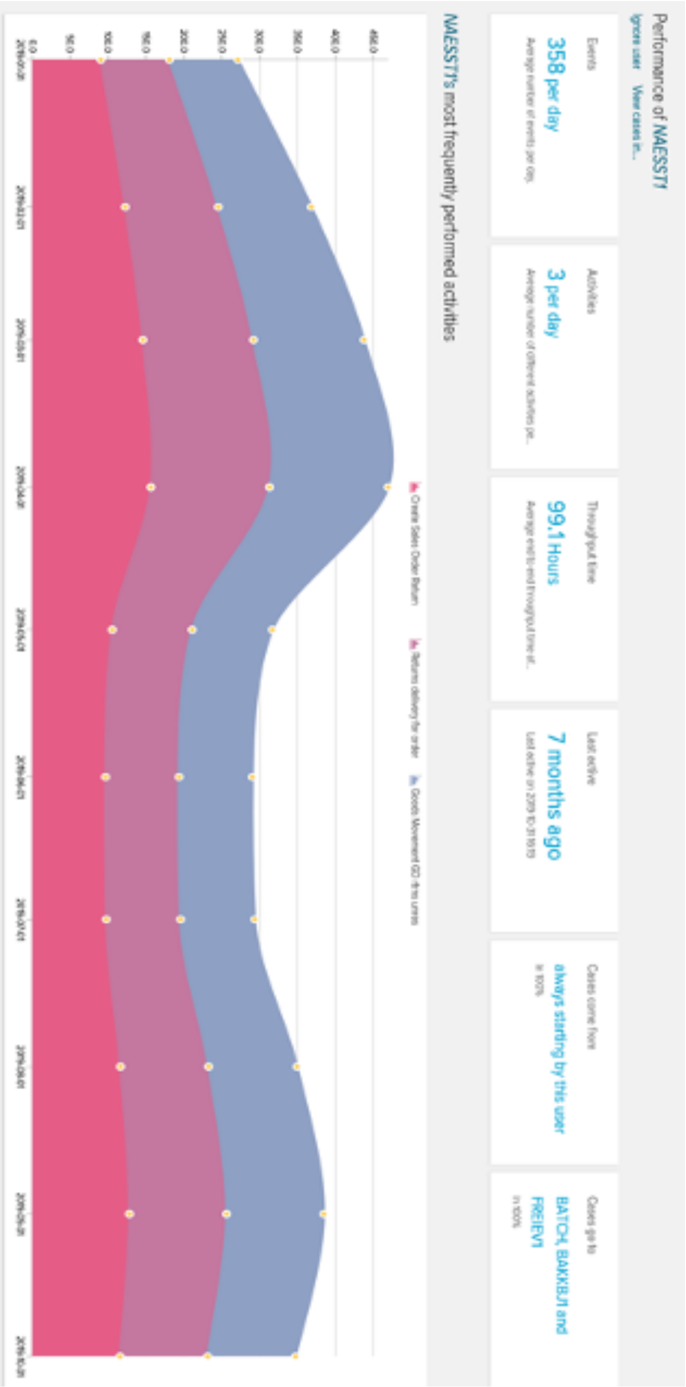
000088587's most frequently performed activities

Credit Sales Order Return
Credit Memo
Goods Movement GD this unrest
Returns delivery for order


0000088587 - Create Sales Order Return
0000088587 - Credit Memo
0000088587 - Goods Movement GD rtrns unres
0000088587 - Returns delivery for order
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0:00 - 2:00
2:00 - 4:00
4:00 - 6:00
6:00 - 8:00
8:00 - 10:00
10:00 - 12:00
12:00 - 14:00
14:00 - 16:00
16:00 - 18:00
18:00 - 20:00
20:00 - 22:00
22:00 - 24:00
Time of day

Appendix D

The figures provide details about the most frequent user (NAESST1) that handles return-orders. Overall, the user solves in average 358 events per day. The productivity is on it highest in March 2019 and lowest in June 2019, due to for example summer holiday. It is also reasonable to consider that the user has most to do at work between 14:00-16:00 during the day; resolve number of return-orders. In addition, one can imagine that the working day duration is between 08:00- 16:00 since the number of events are closed to zero after 16:00.



Performance of NAESST1

Ignore user View cases in...

Events
358 per day
Average number of events per day

Activities
3 per day
Average number of different activities pe...

Throughput time
99.1 Hours
Average end-to-end throughput time of...

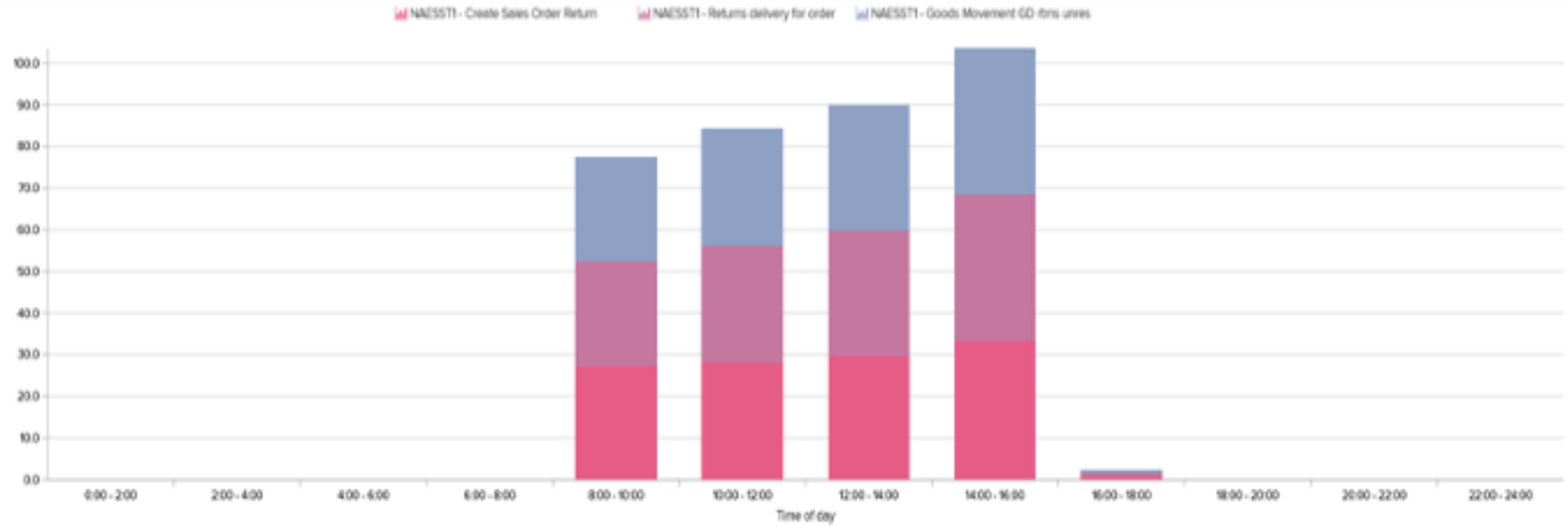
Last active
7 months ago
Last active on 2019-10-31 16:13

Cases come from
always starting by this user
in 100%

Cases go to
BATCH, BAKKBJ1 and FREIEV1
in 100%

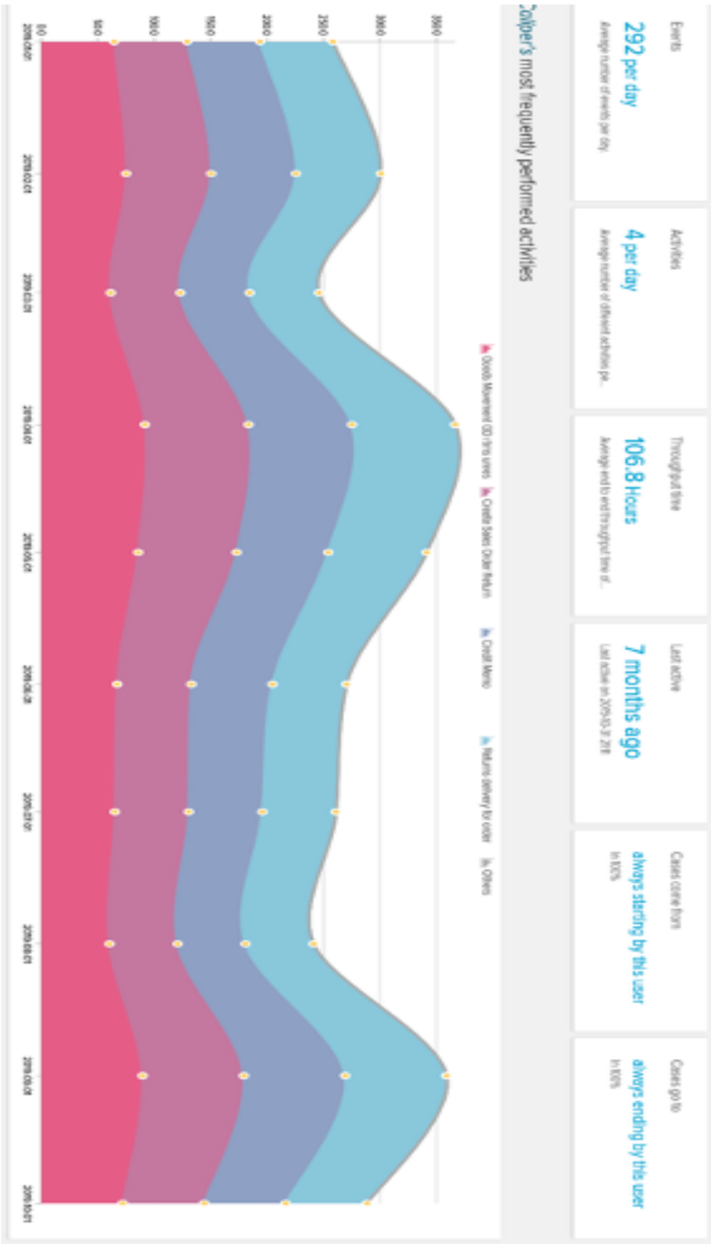
NAESST1's most frequently performed activities

Create Sales Order Return
Returns delivery for order
Goods Movement GD 4 to unres
0.0
50.0
100.0
150.0
200.0
250.0
300.0
350.0
400.0
450.0


NAESST1 - Create Sales Order Return
NAESST1 - Returns delivery for order
NAESST1 - Goods Movement GD rtns unres
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
0:00 - 2:00
2:00 - 4:00
4:00 - 6:00
6:00 - 8:00
8:00 - 10:00
10:00 - 12:00
12:00 - 14:00
14:00 - 16:00
16:00 - 18:00
18:00 - 20:00
20:00 - 22:00
22:00 - 24:00
Time of day

Appendix E

The figures illustrate detailed information about the most frequent product, *Caliper* that customers return to suppliers. Overall, the majority of Caliper seems to be delivered in April 2019 and in September 2019. Low season for delivering the product Caliper is in March 2019 and August 2019. It is reasonable to assume that vehicles need periodic vehicle control or service, and the demand for caliper increases in April 2019 for summer season as well as for the winter season in September 2019. In addition, it is reasonable to consider a working day to be between 08:00-16:00, as most Calipers are being returned during this period. Especially between 12:00- 14:00. Moreover, the system seems to create credit memo automatically between 18:00-24:00, as the amount of credit-memos are being produced at a high rate.



Events
292 per day
Average number of events per day

Activities
4 per day
Average number of different activities pe...

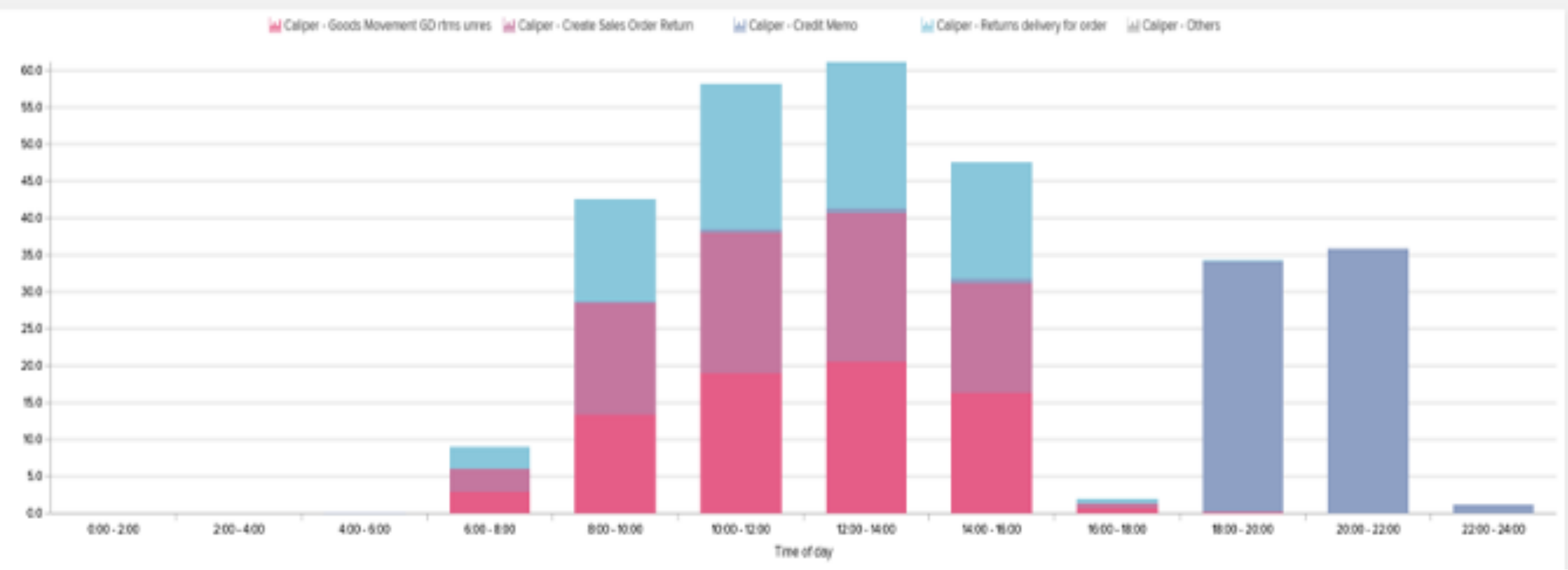
Throughput time
106.8 Hours
Average end to end throughput time of...

Last active
7 months ago
Last active on 2019-10-31 2:18

Cases come from
always starting by this user
in 100%

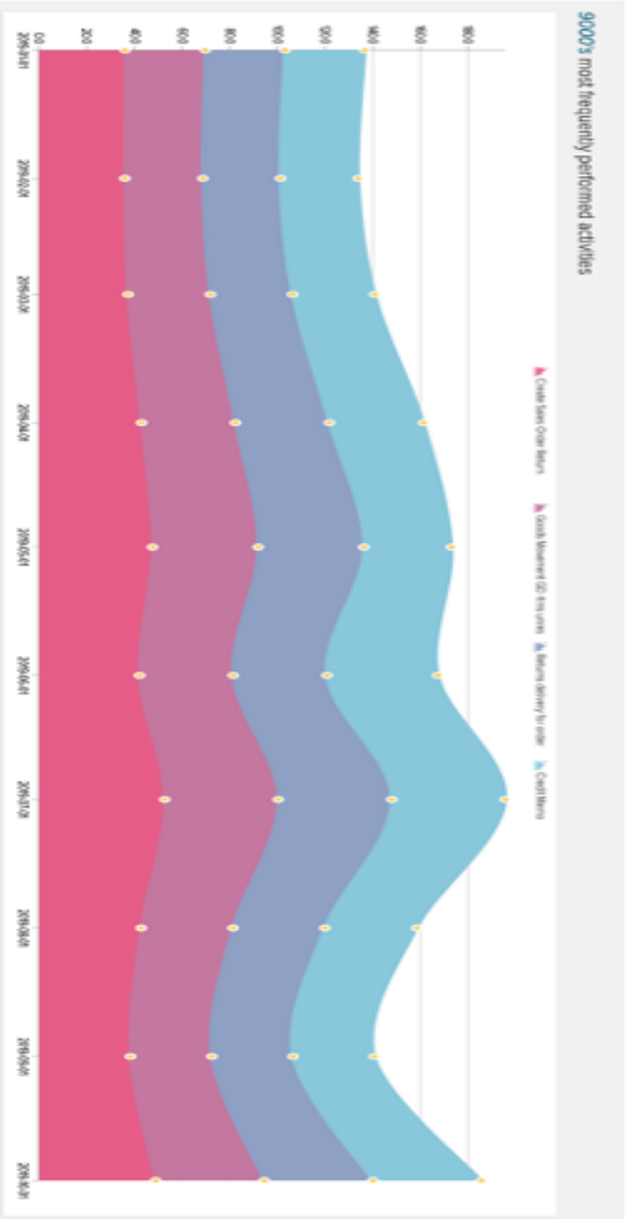
Cases go to
always ending by this user
in 100%

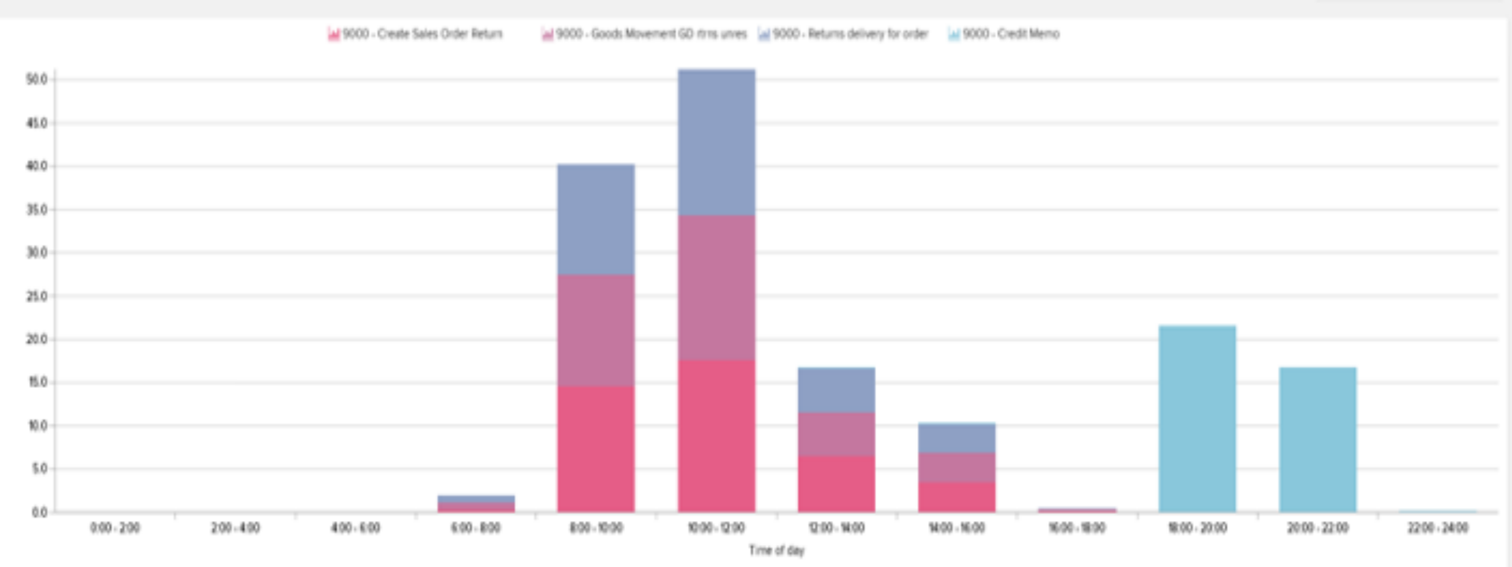
Caliper's most frequently performed activities


Caliper - Goods Movement GD rtns unres
Caliper - Create Sales Order Return
Caliper - Credit Memo
Caliper - Returns delivery for order
Caliper - Others
60.0
55.0
50.0
45.0
40.0
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
0:00 - 2:00
2:00 - 4:00
4:00 - 6:00
6:00 - 8:00
8:00 - 10:00
10:00 - 12:00
12:00 - 14:00
14:00 - 16:00
16:00 - 18:00
18:00 - 20:00
20:00 - 22:00
22:00 - 24:00
Time of day

Appendix F

The figures illustrate details of storage location 9000. Overall, storage location seems to have increasingly high number of returns. i.e., shipped to this location in July of 2019 and fewer returns in September 2019. The average number of events are 160 per day, whereas the majority of returned items seems to be handled between 10:00-12:00 during the day. It is reasonable to believe that the lower productivity to this particular storage location is caused by processing return-orders slower compared to other storage locations. Maybe they have lower capacity to handle return-orders, less staff or different procedures to handle return-orders.




9000 - Create Sales Order Return
9000 - Goods Movement GD rtns unres
9000 - Returns delivery for order
9000 - Credit Memo
50.0
45.0
40.0
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
0:00 - 2:00
2:00 - 4:00
4:00 - 6:00
6:00 - 8:00
8:00 - 10:00
10:00 - 12:00
12:00 - 14:00
14:00 - 16:00
16:00 - 18:00
18:00 - 20:00
20:00 - 22:00
22:00 - 24:00
Time of day

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway